Machine Learning I
80-629A

Apprentissage Automatique I
80-629

Recommender Systems
— Week #11

Introduction

- Recommendation task:
  - Suggest items of interest to users
    - Items: movies, books, articles, humans
    - Users: humans

Is it worth our attention?

- Recommendation is the next search
    - Search finds items (given a query)
    - Recommendation finds items of interest

Is it worth our attention?

- Recommendation is the ~~next~~ new search
  - Search finds items (given a query)
  - Recommendation finds items of interest

amazon.com

Frequently Bought Together

Price for all three: $230.47

Show availability and shipping details

☑ **This item:** Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop
Hardcover $64.01

☑ Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series) by Kevin P. Murphy
Hardcover $78.26

☑ Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning ... by Daphne Koller
Hardcover $88.20

Customers Who Bought This Item Also Bought

Page

Learning From Data
› Yaser S. Abu-Mostafa
(56)
Hardcover

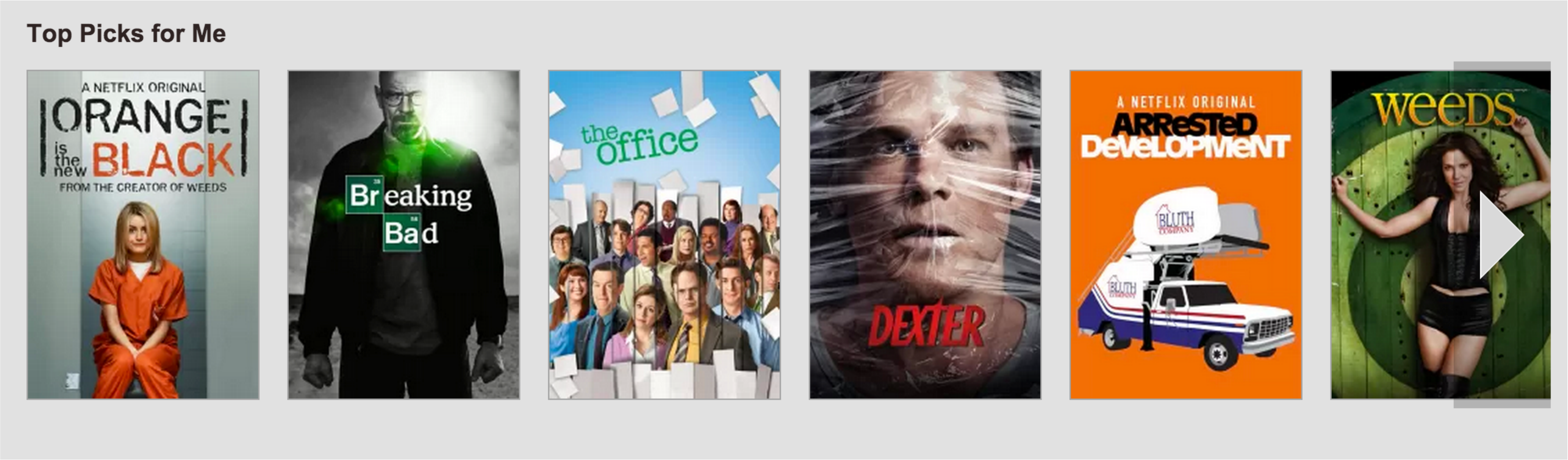
Machine Learning: A Probabilistic ...
› Kevin P. Murphy
(29)
Hardcover

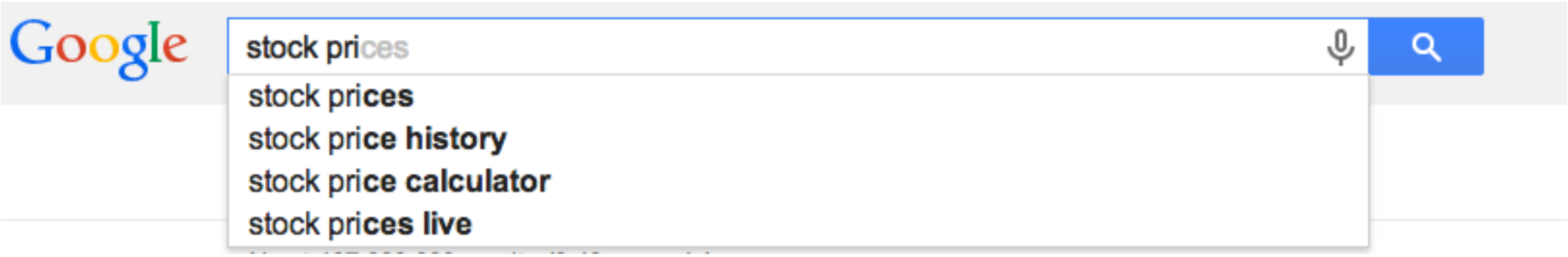
The Elements of Statistical Learning: ...
Trevor Hastie
(31)
Hardcover

Probabilistic Graphical Models: Principles ...
› Daphne Koller
(24)
Hardcover

Machine Learning
› Tom M. Mitchell
(47)
Hardcover
$195.71 ✓Prime

NETFLIX

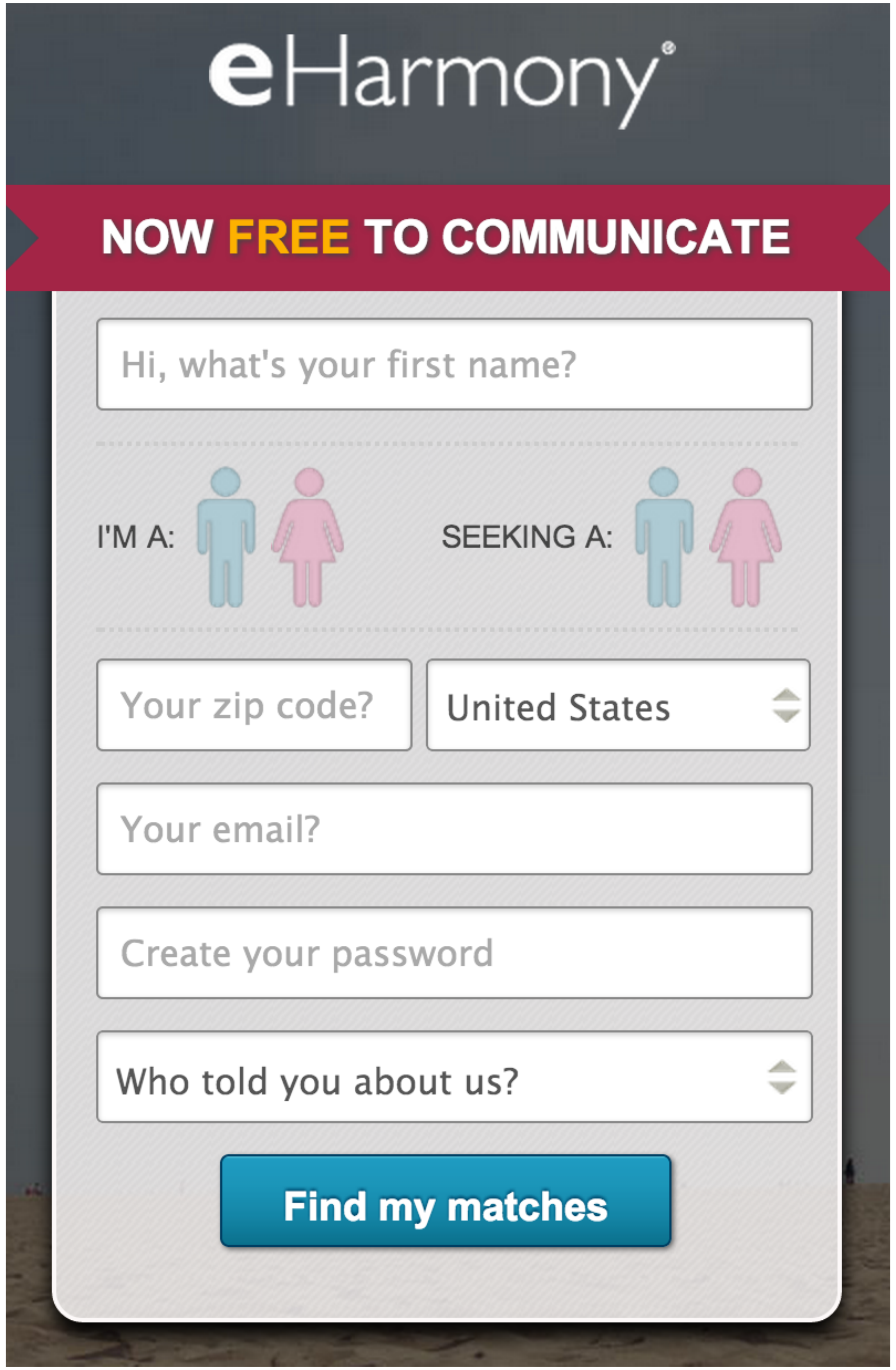

Top Picks for Me
A NETFLIX ORIGINAL
ORANGE is the new BLACK
FROM THE CREATOR OF WEEDS
Breaking Bad
the office
DEXTER
A NETFLIX ORIGINAL
ARRESTED DEVELOPMENT
BLUTH COMPANY
WEEDS



stock prices

- stock pri**ces**
- stock price **history**
- stock price **calculator**
- stock pri**ces live**

- They are responsible for 4% of US marriages (from 2005 to 2012)
  - And lower divorce rates



[Cacioppo'13]

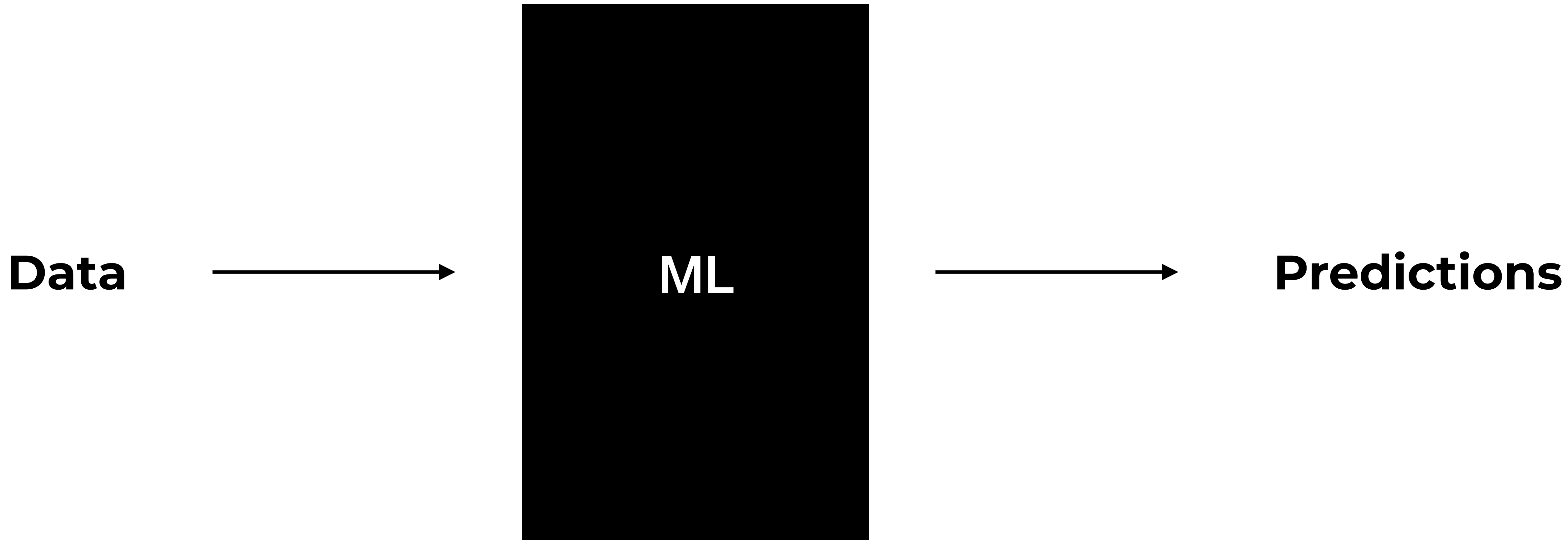
Machine Learning for Recommender Systems

- Task: Suggest items of interest to users
- From data how do you determine what denotes interest?
  - Item-specific signal (supervised learning)

1. Score: rating, bid
2. Consumption: click, buy, watch, bookmark

- Imagine
  - The data are user ratings
  - Task: Recommend items the user will like

- How do we set it up as a machine learning problem?



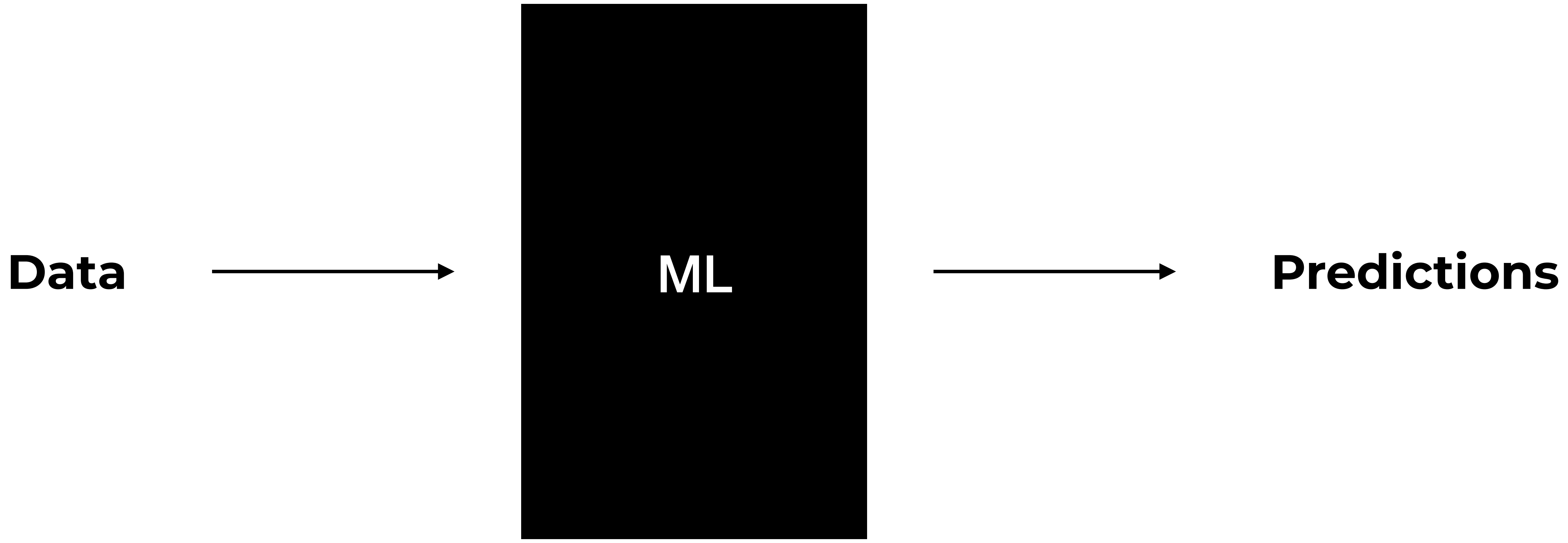
Data

ML

Predictions

- Imagine
    - The data are user ratings
    - Task: Recommend items the user will like

- How do we set it up as a machine learning problem?



Data → ML → **Predictions**

- **Task:** What do we learn? What do we predict? What is the model?
- **Performance measure:** How do we evaluate the results?
- **Experience:** How does our model interact with data?

Framework for recommendation problems

Data

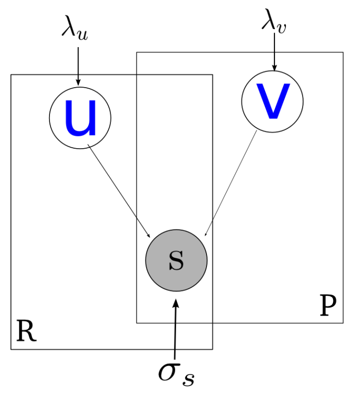
Representation of user preferences

Recommendations

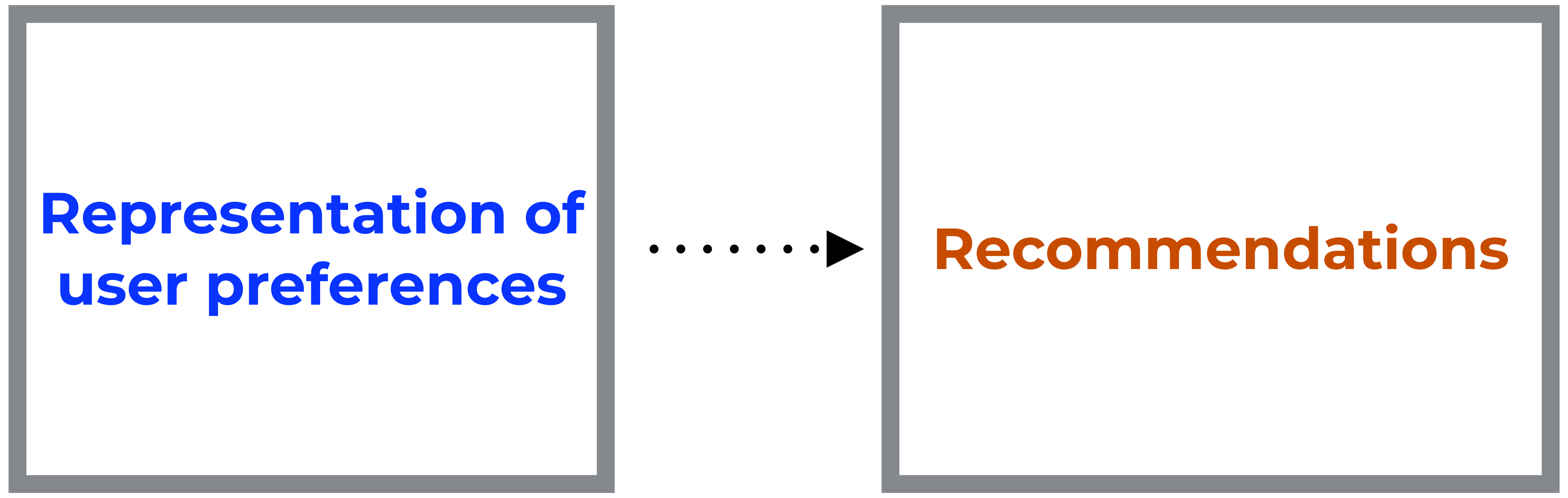
- User preferences

- User/Item features

- Model



- E.g. Top-N recommendations



- **Task:**
  - How we we set it up?
    1. Regression (Classification)
    2. Ranking

Recommendations

Ranking vs. Regression

A. Ranking models

- Computationally more expensive
- E.g., Have to consider a group of items (listwise)

$$f : \underbrace{(u, i_1, i_2, \ldots, i_m)}_{\text{user u's unseen items}} \rightarrow \underbrace{(r_1, r_2, \ldots, r_m)}_{\text{rank of each item}}$$

B. Score models

- For each user:
  1. Predict scores of all unseen items $f : (u, i) \rightarrow \mathbb{R}$
  2. Rank items (show top-K)

Framework for recommendation problems

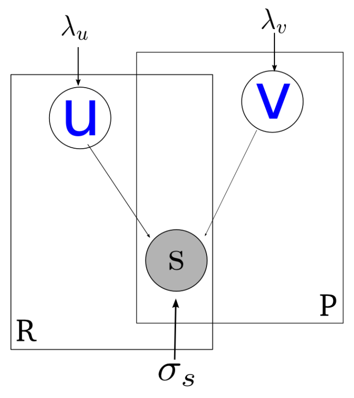
Data → **Representation of user preferences** → **Predict Missing Ratings**

- **User preferences**

  ★★★★☆

- **User/Item features**

- **Model**



Score Prediction as regression

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}_{\text{users}\times\text{items}} \xrightarrow{f} \begin{bmatrix} 3 & 2 & \cdots & 0 \\ 1 & 0 & \cdots & 3 \\ \vdots & & \ddots & \cdots \\ 0 & 2 & \cdots & 2 \end{bmatrix}$$

Train: Black S^o
Test : Red S^u

Modelling?

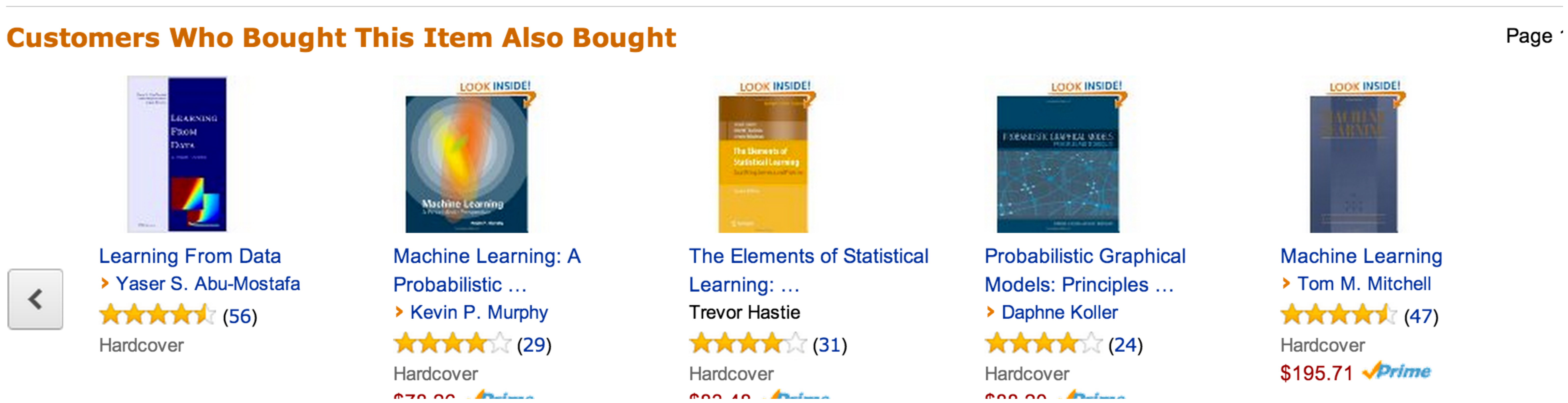
$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}_{\text{users} \times \text{items}} \xrightarrow{f} \begin{bmatrix} 3 & 2 & \cdots & 0 \\ 1 & 0 & \cdots & 3 \\ \vdots & & \ddots & \cdots \\ 0 & 2 & \cdots & 2 \end{bmatrix}$$

- How do we set this up as a learning problem?

$$S^u = f(S^o)$$

Collaborative Filtering (CF)

- Assumption:
  - Users with past similar preferences will have similar future preferences
- Work horse used in many recommender systems



Customers Who Bought This Item Also Bought

Page

Learning From Data
Yaser S. Abu-Mostafa
(56)
Hardcover

Machine Learning: A Probabilistic …
Kevin P. Murphy
(29)
Hardcover

The Elements of Statistical Learning: …
Trevor Hastie
(31)
Hardcover

Probabilistic Graphical Models: Principles …
Daphne Koller
(24)
Hardcover

Machine Learning
Tom M. Mitchell
(47)
Hardcover
$195.71 Prime

CF- Neighbourhood approaches

1. **For each user, find other users with similar past preferences**
2. **Predict that user's missing preferences as the weighted combination of its neighbours' preferences**

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}_{\text{users} \times \text{items}}$$

CF- Neighbourhood

1. Find similarity between every pair of users (or items)

$$\text{Sim(u,u')} = \frac{(S^0_u)^\top S^o_{u'}}{\|S^o_u\|\|S^o_{u'}\|}$$

$$\begin{array}{c} \\ S^o_{u'} \\ \\ S^o_u \end{array}\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

2. Predict missing scores using a user's neighbours

$$\hat{S}_{uj} = \sum_{u'} \text{Sim}(u, u') S^o_{u'j},$$

$$\forall u' \text{that have rated j}$$

CF- Neighbourhood approaches

- Non-parametric approach
  - A user is represented by a weighted combination of its neighbours
  - New users can change one's recommendations
- Different distance functions to capture different effects
  - Ratings vs. clicks
  - Could consider additional information

- Works well empirically
- Building similarity matrix can be slow (offline)
- Not probabilistic

CF - Matrix factorization

$$\text{users}\begin{bmatrix}3 & - & \cdots & 0\\ - & 0 & \cdots & -\\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & -\end{bmatrix} \approx \begin{array}{c}\text{k}\\ \text{users}\begin{bmatrix} \\ \theta \\ \\ \end{bmatrix}\end{array} \quad \begin{array}{cc} & \text{items}\\ \text{k} & \begin{bmatrix} & \beta & \end{bmatrix}\end{array}$$

Model. $\mathsf{S}_{ui} := \theta_u^\top \beta_i$

Parameters. $\theta_u \; \forall u, \beta_i \; \forall i$

Objective. $\sum_u \sum_i (\mathsf{S}_{ui} - \hat{\mathsf{S}}_{ui})^2$

- **Assumption: the observation matrix is low-rank**
- **Estimates user and item representations**
- **k is a hyperparameter**

  $k << \min(|\text{Users}|, |\text{Items}|)$

[Salakhutdinov, Mnih, '08]

CF - Matrix factorization. Alternative view.

Model. $\mathsf{S}_{\mathsf{ui}} := \theta_{\mathsf{u}}^{\top} \beta_{\mathsf{i}}$

Imagine that θ_u's are features of users

The model is then a linear regression for each item:

$$\begin{aligned} \mathsf{S}_{\mathsf{ui}} &= \theta_{\mathsf{u}}^{\top} \beta_{\mathsf{i}} \\ &= \sum_{\mathsf{k}} \theta_{\mathsf{uk}} \beta_{\mathsf{ik}} \\ &= \sum_{\mathsf{k}} \theta_{\mathsf{u1}} \beta_{\mathsf{i1}} + \theta_{\mathsf{u2}} \beta_{\mathsf{i2}} + \ldots + \theta_{\mathsf{up}} \beta_{\mathsf{ip}} \end{aligned}$$

Since the model is symmetric in θ and β,
β_i's can be seen as features of items

Model fitting

Objective. $\sum_u \sum_i (S_{ui} - \hat{S}_{ui})^2$

Model fitting

Objective. $\sum_u \sum_i (S_{ui} - \hat{S}_{ui})^2$

- **Joint parameter optimization**
  - **Gradient descent:** $(\nabla\theta, \nabla\beta)$

Model fitting

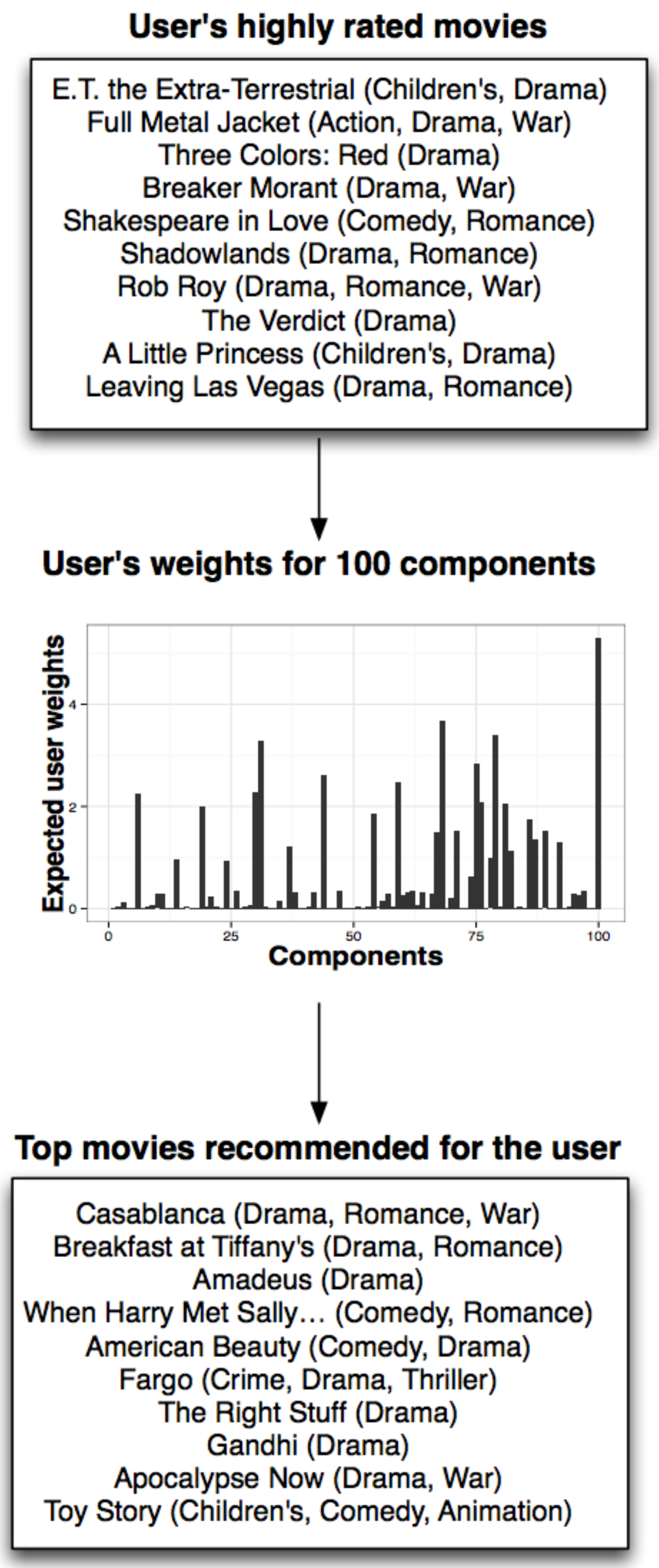
Objective. $\sum_u \sum_i (S_{ui} - \hat{S}_{ui})^2$

- Joint parameter optimization
  - Gradient descent: $(\nabla\theta, \nabla\beta)$
- Alternate optimization
  1. Fix θ, update β
  2. Fix β, update θ
  - Each step is a (regularized) least-squares problem
  - This procedure is known as alternating least squares (ALS)

S^o

Matrix Factorization

S^u



User's highly rated movies

E.T. the Extra-Terrestrial (Children's, Drama)
Full Metal Jacket (Action, Drama, War)
Three Colors: Red (Drama)
Breaker Morant (Drama, War)
Shakespeare in Love (Comedy, Romance)
Shadowlands (Drama, Romance)
Rob Roy (Drama, Romance, War)
The Verdict (Drama)
A Little Princess (Children's, Drama)
Leaving Las Vegas (Drama, Romance)

User's weights for 100 components

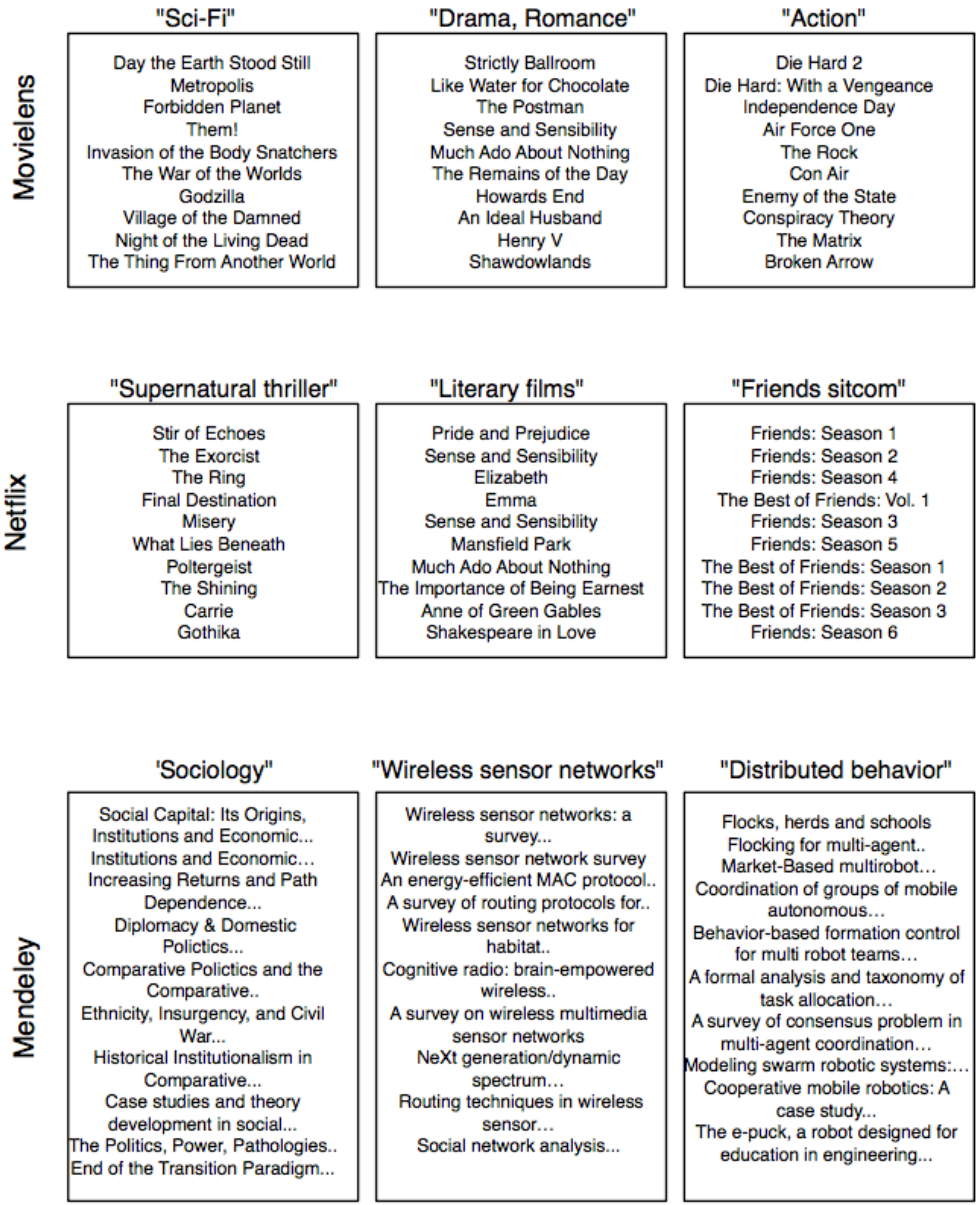
Top movies recommended for the user

Casablanca (Drama, Romance, War)
Breakfast at Tiffany's (Drama, Romance)
Amadeus (Drama)
When Harry Met Sally… (Comedy, Romance)
American Beauty (Comedy, Drama)
Fargo (Crime, Drama, Thriller)
The Right Stuff (Drama)
Gandhi (Drama)
Apocalypse Now (Drama, War)
Toy Story (Children's, Comedy, Animation)

[Gopalan et al.'15]

Model Exporation

$$\textbf{argsort}_i\ \beta_{ik}$$



Movielens

"Sci-Fi"
Day the Earth Stood Still
Metropolis
Forbidden Planet
Them!
Invasion of the Body Snatchers
The War of the Worlds
Godzilla
Village of the Damned
Night of the Living Dead
The Thing From Another World

"Drama, Romance"
Strictly Ballroom
Like Water for Chocolate
The Postman
Sense and Sensibility
Much Ado About Nothing
The Remains of the Day
Howards End
An Ideal Husband
Henry V
Shawdowlands

"Action"
Die Hard 2
Die Hard: With a Vengeance
Independence Day
Air Force One
The Rock
Con Air
Enemy of the State
Conspiracy Theory
The Matrix
Broken Arrow

Netflix

"Supernatural thriller"
Stir of Echoes
The Exorcist
The Ring
Final Destination
Misery
What Lies Beneath
Poltergeist
The Shining
Carrie
Gothika

"Literary films"
Pride and Prejudice
Sense and Sensibility
Elizabeth
Emma
Sense and Sensibility
Mansfield Park
Much Ado About Nothing
The Importance of Being Earnest
Anne of Green Gables
Shakespeare in Love

"Friends sitcom"
Friends: Season 1
Friends: Season 2
Friends: Season 4
The Best of Friends: Vol. 1
Friends: Season 3
Friends: Season 5
The Best of Friends: Season 1
The Best of Friends: Season 2
The Best of Friends: Season 3
Friends: Season 6

Mendeley

'Sociology"
Social Capital: Its Origins, Institutions and Economic...
Institutions and Economic…
Increasing Returns and Path Dependence...
Diplomacy & Domestic Polictics...
Comparative Polictics and the Comparative..
Ethnicity, Insurgency, and Civil War...
Historical Institutionalism in Comparative...
Case studies and theory development in social...
The Politics, Power, Pathologies..
End of the Transition Paradigm...

"Wireless sensor networks"
Wireless sensor networks: a survey...
Wireless sensor network survey
An energy-efficient MAC protocol..
A survey of routing protocols for..
Wireless sensor networks for habitat..
Cognitive radio: brain-empowered wireless..
A survey on wireless multimedia sensor networks
NeXt generation/dynamic spectrum…
Routing techniques in wireless sensor…
Social network analysis...

"Distributed behavior"
Flocks, herds and schools
Flocking for multi-agent..
Market-Based multirobot…
Coordination of groups of mobile autonomous…
Behavior-based formation control for multi robot teams…
A formal analysis and taxonomy of task allocation…
A survey of consensus problem in multi-agent coordination…
Modeling swarm robotic systems:…
Cooperative mobile robotics: A case study...
The e-puck, a robot designed for education in engineering...

6K Users
4K Movies
1M Ratings

480K Users
17.7K Movies
100M Ratings

80K Users
260K Sci. articles
100M Ratings

[Gopalan et al.'15]

Probabilistic matrix factorization

Matrix Factorization

$$\|\beta_i\|_2$$

$$\|\theta_u\|_2$$

$$(S_{ui} - \hat{S}_{ui})^2$$

Minimizing mean squared error is equivalent to maximizing likelihood under Gaussian noise

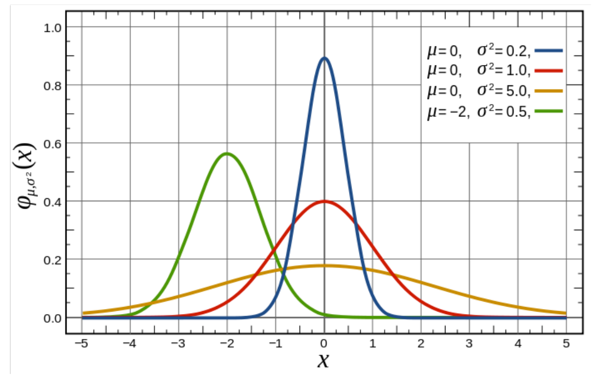
Gaussian Matrix Factorization

[Salakhutdinov et al. ’08]

$$\theta_u \sim \mathcal{N}(a, b)$$

$$\beta_i \sim \mathcal{N}(c, d)$$

$$S_{ui} \sim \mathcal{N}(\theta_u^\top \beta_i, \sigma)$$



[wikipedia]

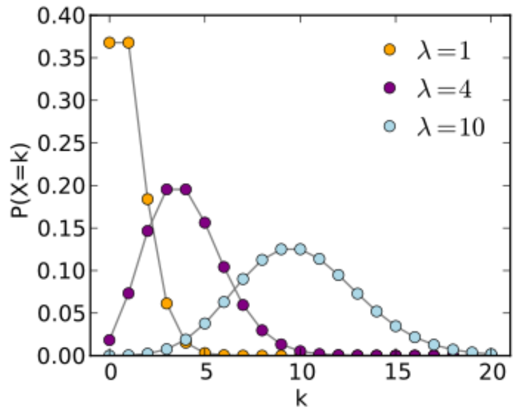
Poisson Matrix Factorization

[Gopalan et al. ’15]

$$\theta_u \sim \text{Gamma}(a, b)$$

$$\beta_i \sim \text{Gamma}(c, d)$$

$$S_{ui} \sim \text{Poisson}(\theta_u^\top \beta_i)$$



[wikipedia]

- Poisson factorization is correct
- Gaussian factorization is incorrect

- In practice MF typically gives better performance than PF

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Encode each categorical variable using a series of indicator variables ⟶

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Encode each categorical variable using a series of indicator variables

$$\longrightarrow$$

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

$$\underbrace{(1\ 0 \ldots 0, 1\ 0 \ldots 0}_{x}, \underbrace{3)}_{y}$$

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Encode each categorical variable using a series of indicator variables →

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

$$(\underbrace{\underbrace{1\ 0 \ldots 0}_{x1},\ \underbrace{1\ 0 \ldots 0}_{x2}}_{x},\ \underbrace{3}_{y})$$

Towards CF with deep learning

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Encode each categorical variable using a series of indicator variables →

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

$$\underbrace{(\underbrace{1\ 0\ \ldots\ 0}_{x1},\ \underbrace{1\ 0\ \ldots\ 0}_{x2}}_{x},\ \underbrace{3)}_{y} \longrightarrow \boldsymbol{y} = (\boldsymbol{x}_1^\top \theta)^\top (\boldsymbol{x}_2^\top \beta)$$

Towards CF with deep learning

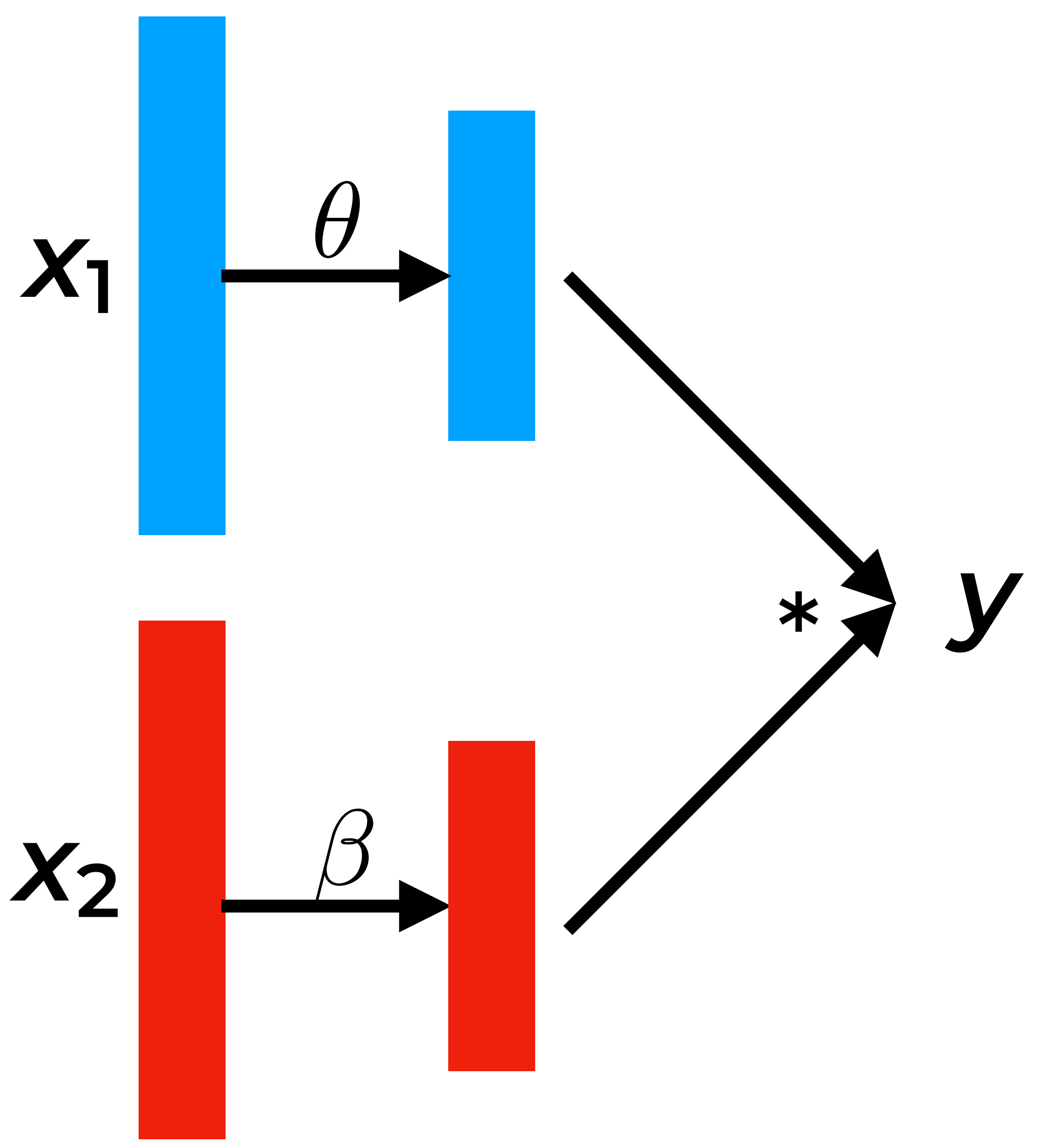
$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

Encode each categorical variable using a series of indicator variables ⟶

User Item
(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

$$(\underbrace{\underbrace{1\ 0 \ldots 0}_{x1}, \underbrace{1\ 0 \ldots 0}_{x2}}_{x}, \underbrace{3}_{y}) \longrightarrow \boldsymbol{y} = (\boldsymbol{x}_1^\top \theta)^\top (\boldsymbol{x}_2^\top \beta)$$



Towards CF with deep learning

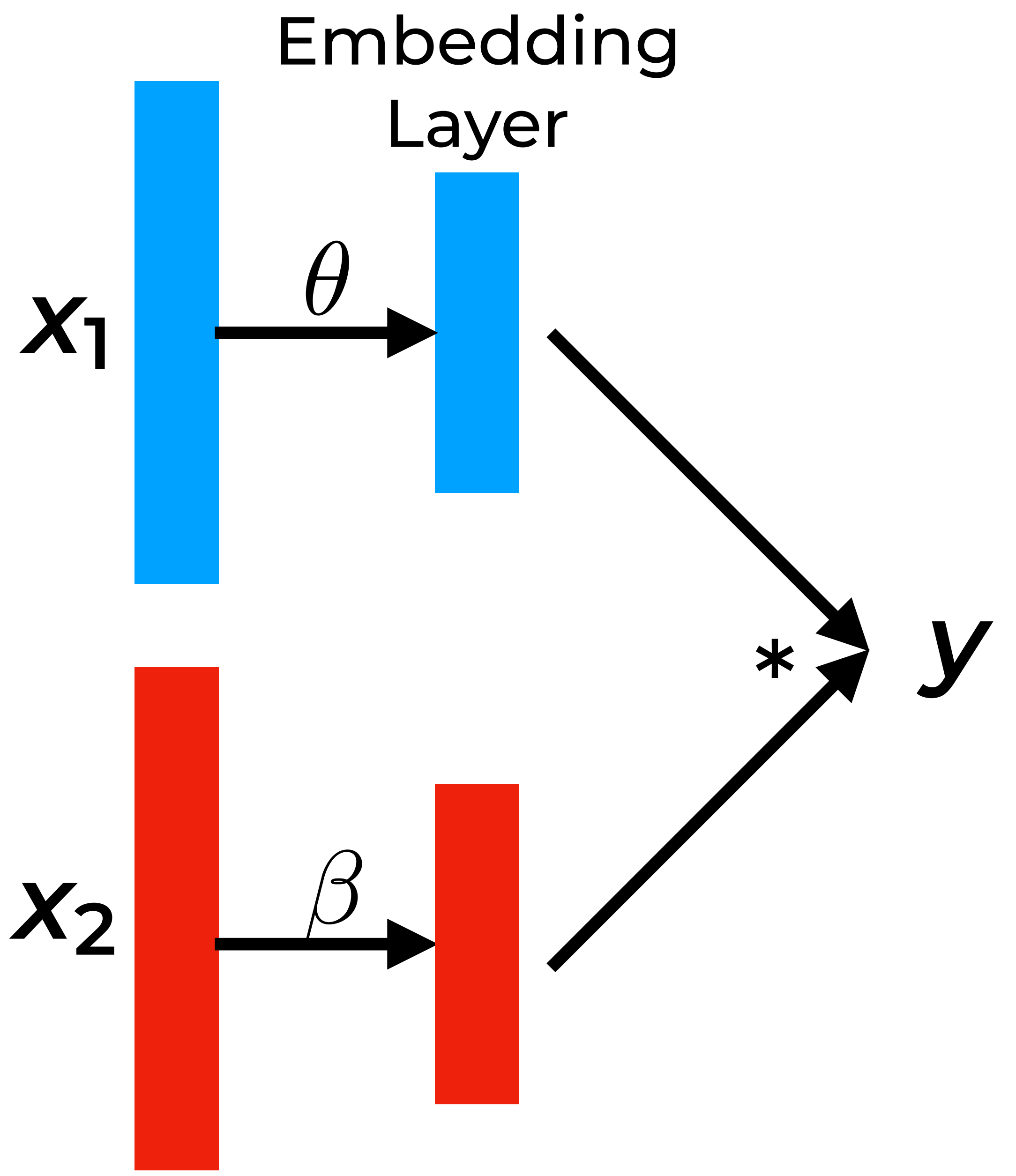
$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix} \longrightarrow$$

(User 1, Item 1, 3)
(User 1, Item 2, 0)
(User 2 Item 2, 0)
(User n, Item 1, 2)

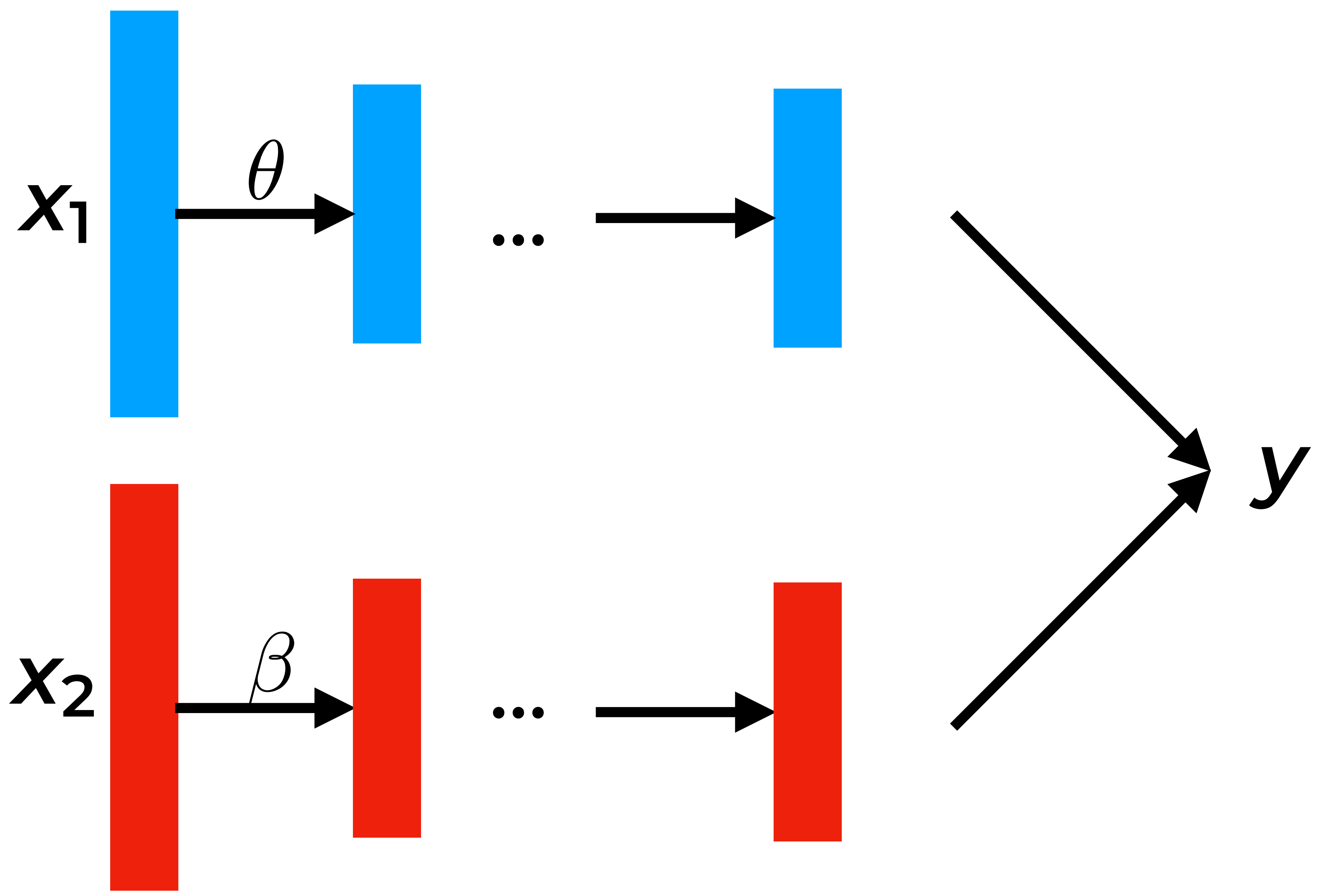
Encode each categorical variable using a series of indicator variables ⟶

User Item
(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

$$\underbrace{\underbrace{(1\ 0 \dots 0}_{x1},\ \underbrace{1\ 0 \dots 0}_{x2}}_{x},\ \underbrace{3)}_{y} \longrightarrow \boldsymbol{y} = (\boldsymbol{x}_1^\top \theta)^\top (\boldsymbol{x}_2^\top \beta)$$



A version of Deep Matrix Factorization



- Can do more complicated user and item combinations (beyond dot product)

[Xue et al.'17]

Autoencoders

- Popular neural-network model often used in unsupervised learning (e.g., dim reduction)

Autoencoders

- Popular neural-network model often used in unsupervised learning (e.g., dim reduction)
  - Non-linear PCA

Autoencoders

- Popular neural-network model often used in unsupervised learning (e.g., dim reduction)
  - Non-linear PCA
  - Intuition: let’s learn to copy the data

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x})$$

Autoencoders

- Popular neural-network model often used in unsupervised learning (e.g., dim reduction)
  - Non-linear PCA
  - Intuition: let's learn to copy the data

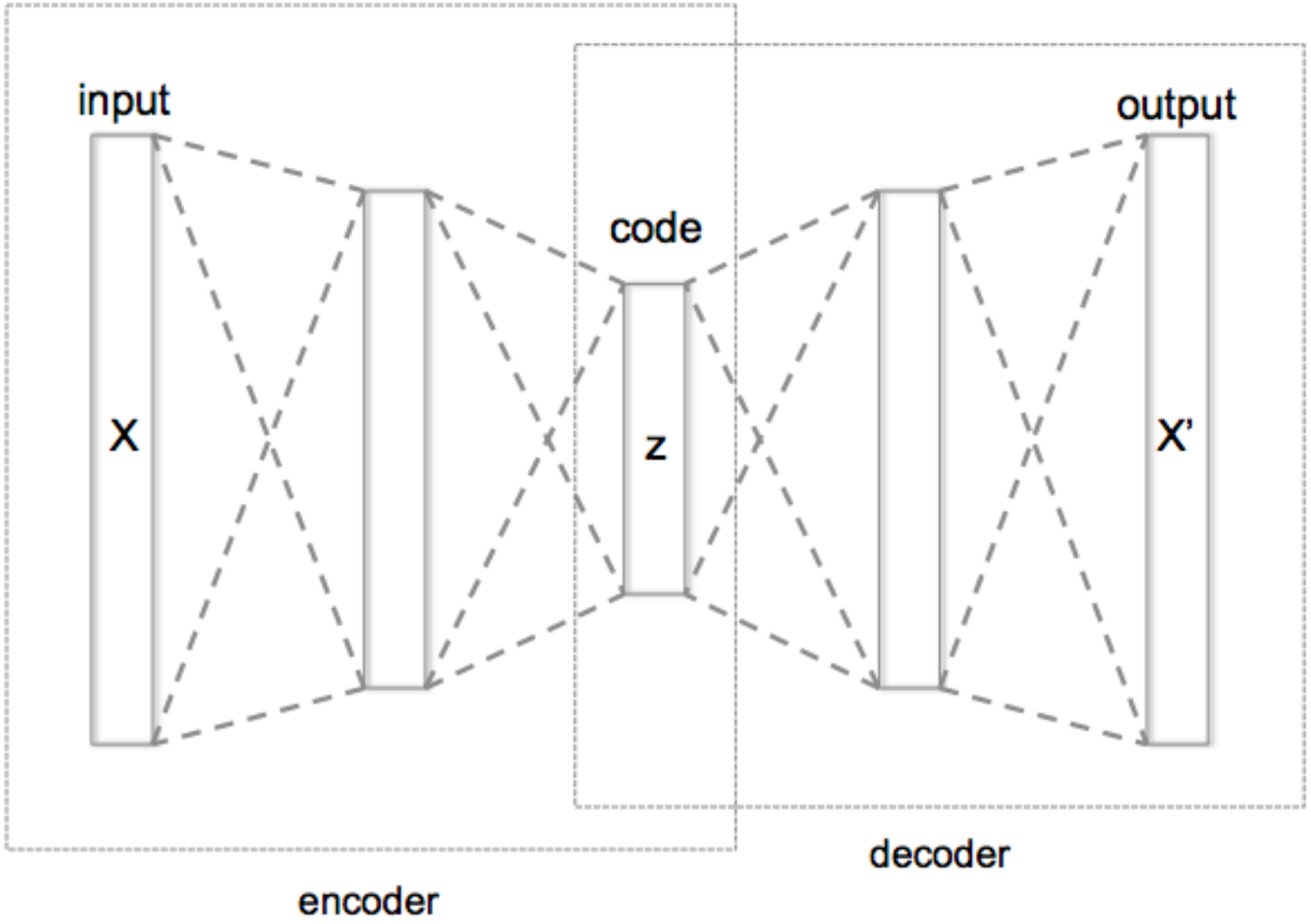
$$x' = f(x)$$

    - We force a "bottleneck"

$$z = f_1(x)$$
$$x' = f_2(z)$$
$$dim(z) < dim(x)$$

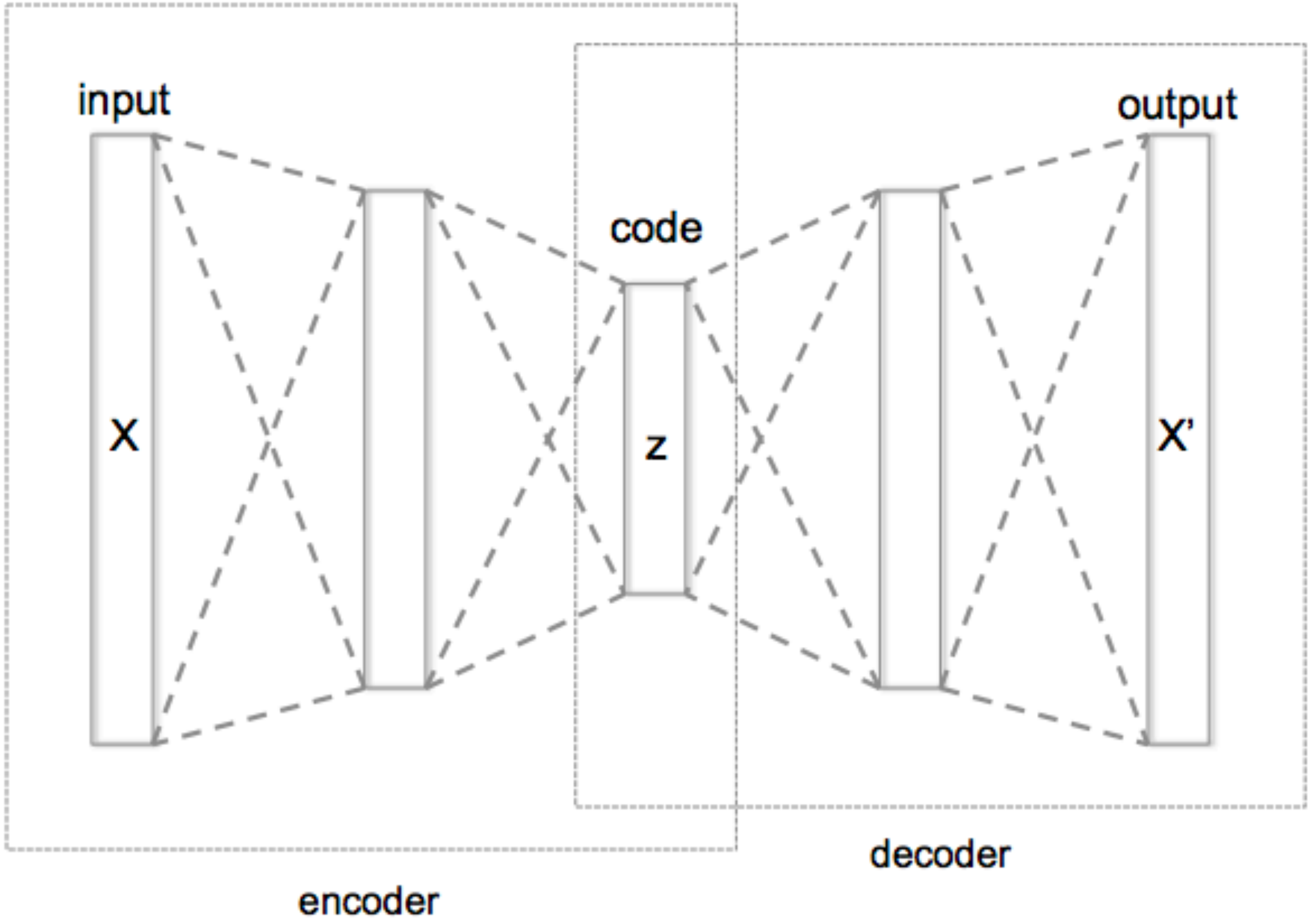
Autoencoders

$$z = f_1(x)$$
$$x' = f_2(z)$$
$$dim(z) < dim(x)$$



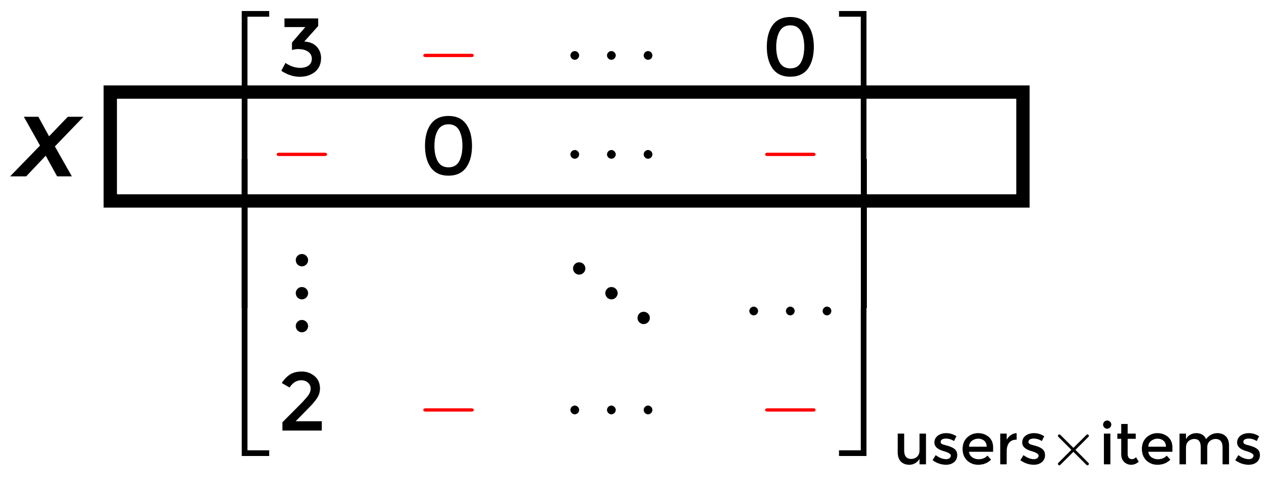
[From wikipedia]

Autoencoders for CF



[From wikipedia]

- X: Either a row or a column of the ratings matrix



- Missing entries:
  - Set to 0 in the input
  - Not considered in the output
- Many versions using denoising-autoencoders (DAEs), VAEs, different likelihoods (etc.)

MF vs. AE for CF

- AE are more “naturally” non-linear
- AE are asymmetric
  - must choose whether to model users or items
- Versions of AEs are close to the state-of-the-art today

How to choose the right model?

- Search for papers that compare different models
  - Keep a healthy does of scepticism
    - I.e., results in papers is not necessarily ground truth
- Try it out on your data
  - Compare performance on held-out data
  - Can it handle: your data size, service speed, updating schedule, other desiderata (fairness, uncertainty estimates) …

Are deep models better?

- 2014: No
- 2018: Yes
- 2019: Maybe not … *
  - “Embarrassingly Shallow Autoencoders for Sparse Data”, Steck’19
  - “On the Difficulty of Evaluating Baselines: A Study on Recommender Systems”, Rendle et al.’19:
    - “With a careful setup of a **vanilla matrix factorization** baseline, we are not only able to improve upon the reported results for this baseline but even outperform the reported results of any newly proposed method.”

(* This is not considering possibly available covariates)

Explicit vs. Implicit data

Explicit vs. Implicit data

- Up to now we assumed ratings
- Ratings are **explicit data:**
  - "Users explicitly provide their preferences"
  - A high rating means the user liked the item
  - A low rating means the user disliked the item

Explicit vs. Implicit data

- Up to now we assumed ratings
- Ratings are **explicit data:**
  - "Users explicitly provide their preferences"
  - A high rating means the user liked the item
  - A low rating means the user disliked the item
- In practice implicit data is much more common:
  - click, buy, watch, listen

Challenge with Implicit Data

- Consuming an item usually implies a positive preference
- Not Consuming an item may either indicate:
    A. A negative preference
    B. Something else: e.g., lack of exposure or time

Challenge with Implicit Data

- The preference matrix is “full” (as opposed to sparse)
  - ‘1’ indicates a consumed item
  - ‘0’ indicates an unconsumed item

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

  - You must take both 0s and 1s into account
  - In practice many models can be adapted to implicit case

Common strategy for implicit data

- Model the 0s as being "less certain" than the 1s

Objective MF: $\frac{1}{|\text{users}|} \sum_u \sum_i (S_{ui} - \hat{S}_{ui})^2$

Objective WMF: $\frac{1}{|\text{users}|} \sum_u \sum_i c_{ui}(S_{ui})(S_{ui} - \hat{S}_{ui})^2$

$$c_{\text{ui}}(0) < c_{\text{ui}}(1)$$

  - Weighted Matrix Factorization [Hu et al. '08]
- Learn the weight of each zero
  - Exposure Matrix Factorization [Liang et al.'15]

User/Item features (I)

- Often additional information exists

User/Item features (I)

- Often additional information exists
  - Users: demographic information, social networks

User/Item features (I)

- Often additional information exists
  - Users: demographic information, social networks
  - Items: content (e.g., movie genre/trailer, book text)
  - Users & items:

User/Item features (I)

- Often additional information exists
  - Users: demographic information, social networks
  - Items: content (e.g., movie genre/trailer, book text)
  - Users & items:
    - timestamps, session information

User/item features (II)

- Allow for **content-based** recommendations

User/item features (II)

- Allow for **content-based** recommendations
  - Good to combat the **cold-start problem**

User/item features (II)

- Allow for **content-based** recommendations
  - Good to combat the **cold-start problem**
- **Assume that features are predictive of preferences**

User/item features (II)

- Allow for **content-based** recommendations
  - Good to combat the **cold-start problem**
- Assume that features are predictive of preferences
  - More difficult in some domains than others (e.g., movies)

User/item features (II)

- Allow for **content-based** recommendations
  - Good to combat the **cold-start problem**
- Assume that features are predictive of preferences
  - More difficult in some domains than others (e.g., movies)
- A practical approach is to bootstrap with content-based to gather preference data and then switch to CF

User/item features (II)

- Allow for **content-based** recommendations
  - Good to combat the **cold-start problem**
- Assume that features are predictive of preferences
  - More difficult in some domains than others (e.g., movies)
- A practical approach is to bootstrap with content-based to gather preference data and then switch to CF
- In the next slides we explore **hybrid models** for these data

Modelling Strategy

1. Generic models
   - Easily extend to many different use cases
2. Tailored modelling for specific features
   - This is where neural nets shine (images, text, networks)

Factorization Machines (FM) [generic model]

- Model "all" additional information

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

[Rendle'10]

Factorization Machines (FM) [generic model]

- **Model "all" additional information**

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

Encode each categorical variable using a series of indicator variables →

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

[Rendle'10]

Factorization Machines (FM) [generic model]

- Model "all" additional information

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

Encode each categorical variable using a series of indicator variables →

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

Add features as columns →

User Item Age

(1 0 … 0, 1 0 … 0, 25, 3)
(1 0 … 0, 0 0 … 1, 22, 0)
(0 1 … 0, 0 1 … 1, 55, 0)
(0 0 … 1, 1 0 … 0, 60, 2)

[Rendle'10]

Factorization Machines (FM) [generic model]

- Model “all” additional information

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

Encode each categorical variable using a series of indicator variables →

User Item

(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

Add features as columns →

User Item Age

(1 0 … 0, 1 0 … 0, 25, 3)
(1 0 … 0, 0 0 … 1, 22, 0)
(0 1 … 0, 0 1 … 1, 55, 0)
(0 0 … 1, 1 0 … 0, 60, 2)

Model. $S_{ui} := w_0 + \underbrace{\sum_{i}^{p} w_i x_i}_{\text{per-feature regression}} + + \underbrace{\sum_{j=0}^{p} \sum_{j'=j+1}^{p} \theta_j^\top \theta x_j x_{j'}}_{\text{per-pair regression}}$

[Rendle’10]

Factorization Machines (FM) [generic model]

- Model “all” additional information

$$\begin{bmatrix} 3 & - & \cdots & 0 \\ - & 0 & \cdots & - \\ \vdots & & \ddots & \cdots \\ 2 & - & \cdots & - \end{bmatrix}$$

Encode each categorical variable using a series of indicator variables →

User Item
(1 0 … 0, 1 0 … 0, 3)
(1 0 … 0, 0 0 … 1, 0)
(0 1 … 0, 0 1 … 1, 0)
(0 0 … 1, 1 0 … 0, 2)

Add features as columns →

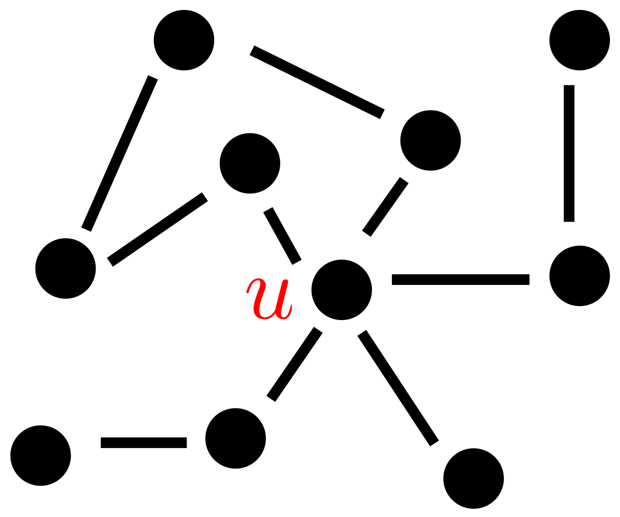
User Item Age
(1 0 … 0, 1 0 … 0, 25, 3)
(1 0 … 0, 0 0 … 1, 22, 0)
(0 1 … 0, 0 1 … 1, 55, 0)
(0 0 … 1, 1 0 … 0, 60, 2)

Model. $S_{ui} := w_0 + \underbrace{\sum_{i}^{p} w_i x_i}_{\text{per-feature regression}} + + \underbrace{\sum_{j=0}^{p} \sum_{j'=j+1}^{p} \theta_j^\top \theta x_j x_{j'}}_{\text{per-pair regression}}$

- Features added to the data (extra columns) are “automatically” used in the model
- Modelling extra information implies adding the feature

[Rendle’10]

A. Social network

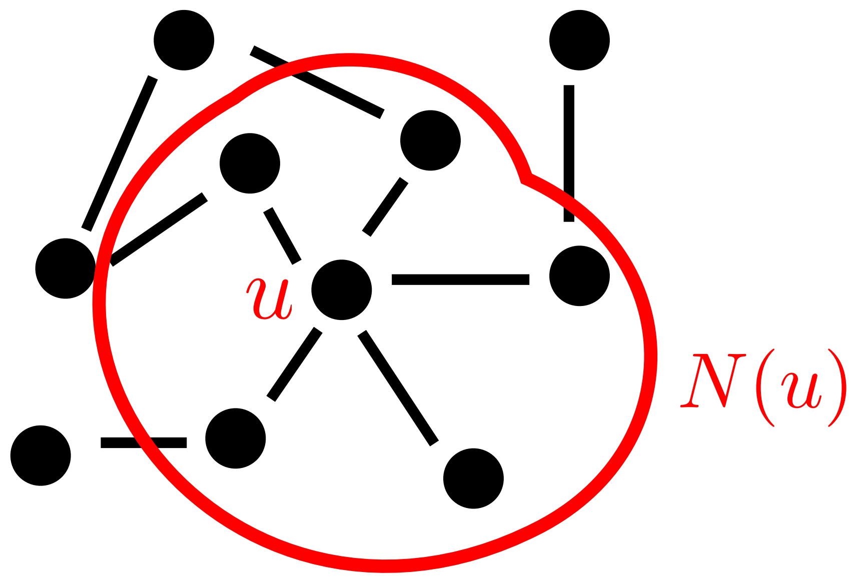


- **Data: user ratings and users' friends**
- **Assume:**
    1. **Friends influence your preferences**
    2. **Different levels of trusts for different friends**

Model. $S_{ui} := \theta_u^\top \beta_i + \sum_{u' \in N(u)} \tau_{un} s_{u'i}$

[Chaney et al. '15]

A. Social network



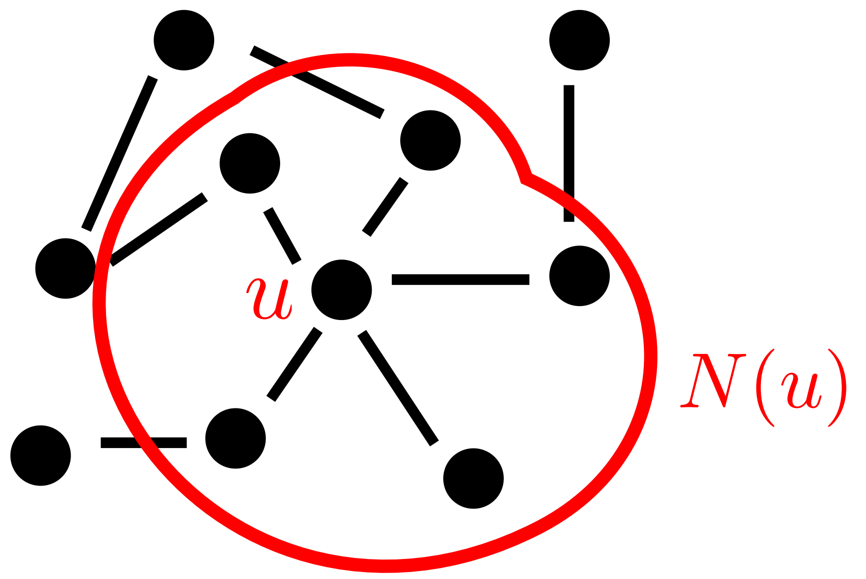
- Data: user ratings and users' friends
- Assume:

1. Friends influence your preferences
2. Different levels of trusts for different friends

Model. $\mathsf{s}_{ui} := \theta_u^\top \beta_i + \sum_{u' \in N(u)} \tau_{un} \mathsf{s}_{u'i}$

[Chaney et al. '15]

A. Social network



- Data: user ratings and users' friends
- Assume:
  1. Friends influence your preferences
  2. Different levels of trusts for different friends

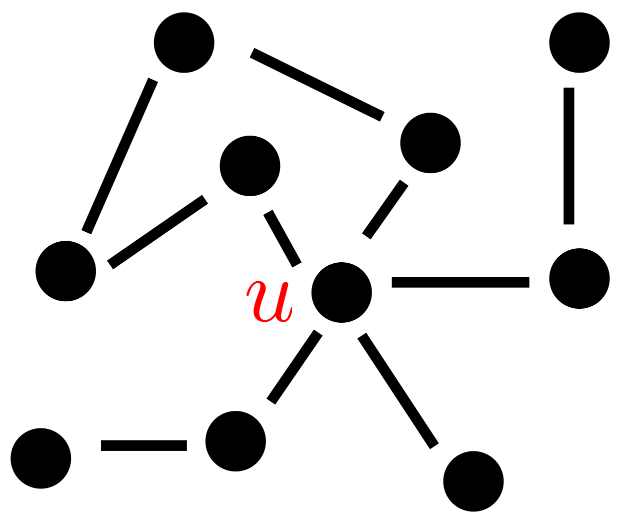
Model. $\mathsf{s}_{\mathsf{ui}} := \theta_{\mathsf{u}}^{\top} \beta_{\mathsf{i}} + \sum_{\mathsf{u}' \in \mathsf{N(u)}} \tau_{\mathsf{un}} \mathsf{s}_{\mathsf{u}'\mathsf{i}}$

The rating of u' on item i

How much u "trusts" u'

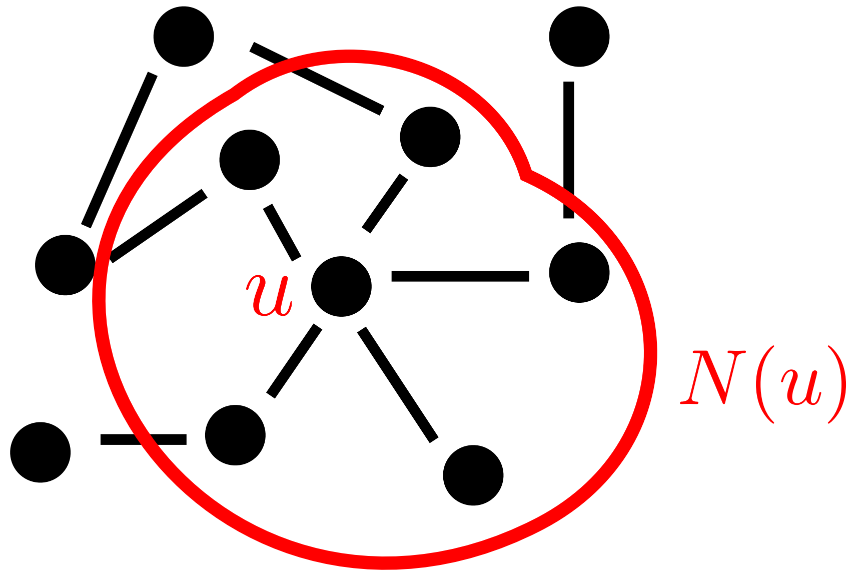
[Chaney et al. '15]

A. Social network



- Recent models use Graph Convolutional Networks (GCNs)
  - Powerful model for graph data

A. Social network



- Recent models use Graph Convolutional Networks (GCNs)
  - Powerful model for graph data

B. Item content

- **Data: user ratings and item text/image/...**

Model. $S_{ui} := \theta_u^\top (\beta_i + \gamma_i)$

[Wang et al. '14]

B. Item content

- **Data: user ratings and item text/image/...**

Model. $S_{ui} := \theta_u^\top(\beta_i + \underset{\substack{\uparrow \\ \boxed{\text{Content features}}}}{\gamma_i})$

[Wang et al. '14]

Questions?

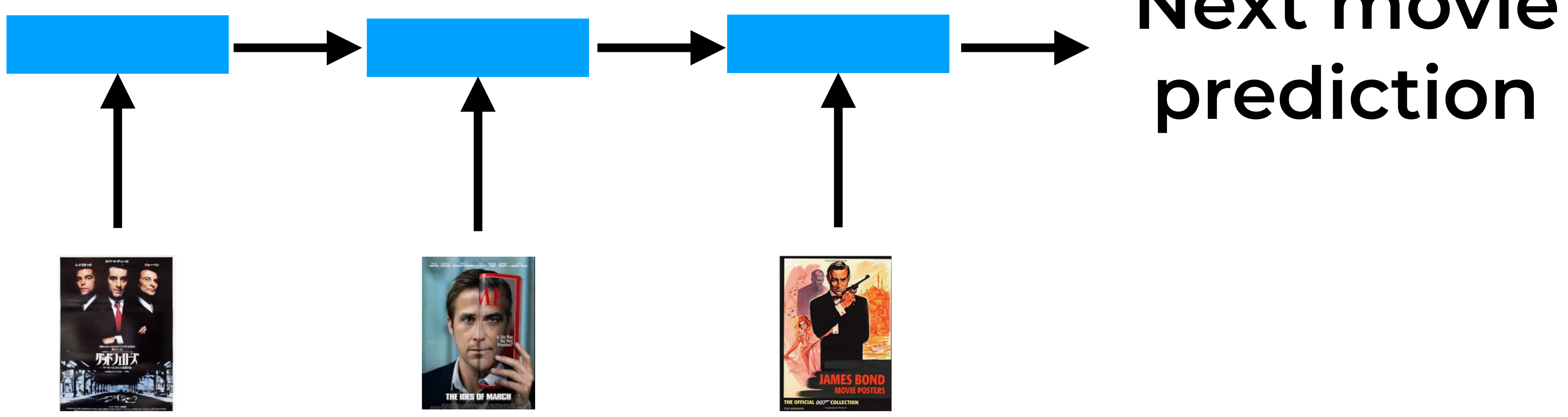
C. Dynamic Modelling

- Data: user ratings with timestamps
- Assume:
  - User tastes change over time
  - Item popularity change over time
- Model. $S_{ui}^{t} := \theta_{u}^{t\top} \beta_{i}^{t}$

  $\theta^{t} = \theta^{t-1} + \epsilon$

C.1 Session-Based Modelling

- Data: user ratings with timestamps
- Assume: Users consume related items over short periods of time
    - Domains: Music playlist, exercises, short videos
- Model. Sequential models like RNNs.



Session-based + Social Networks

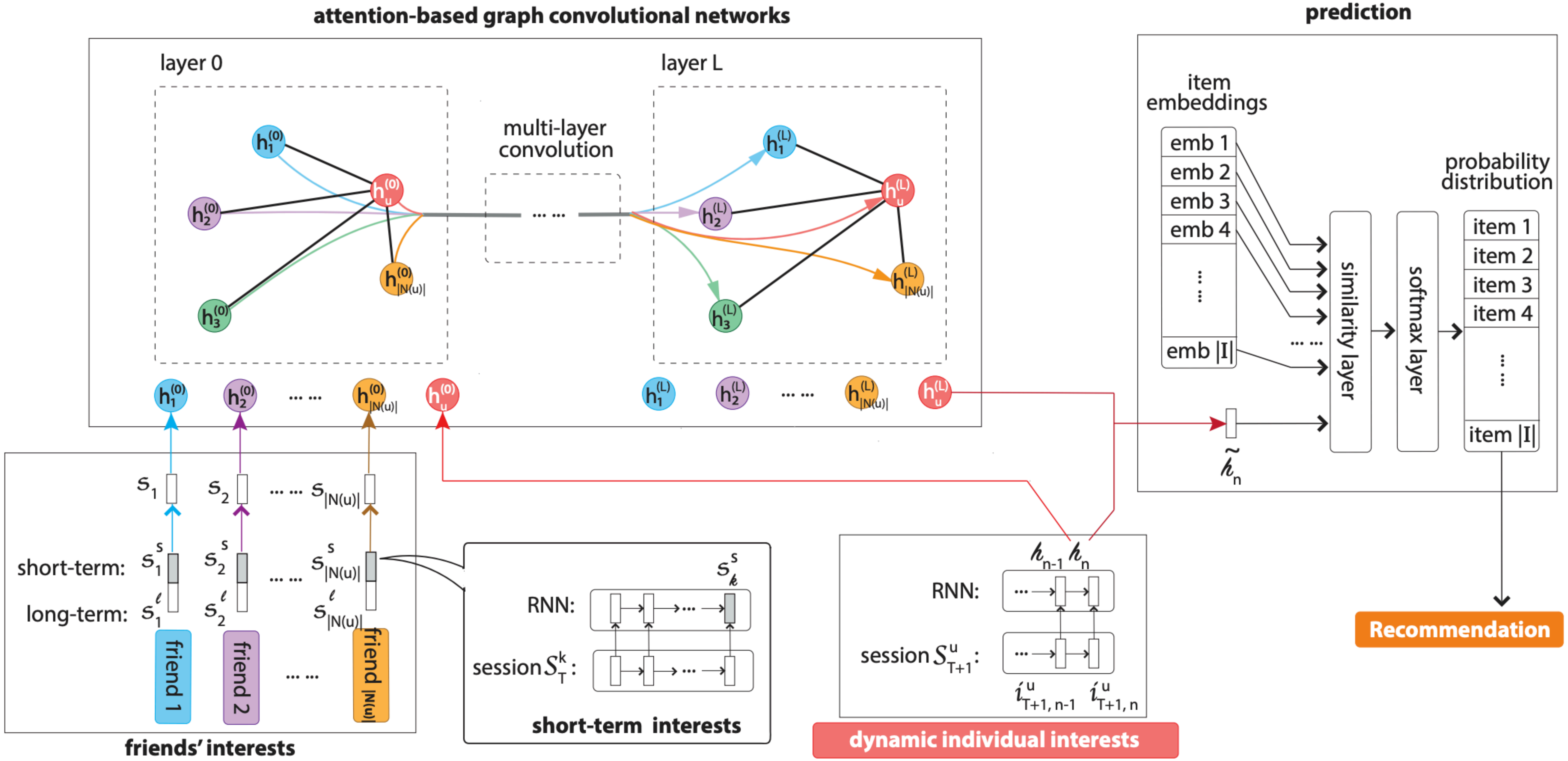
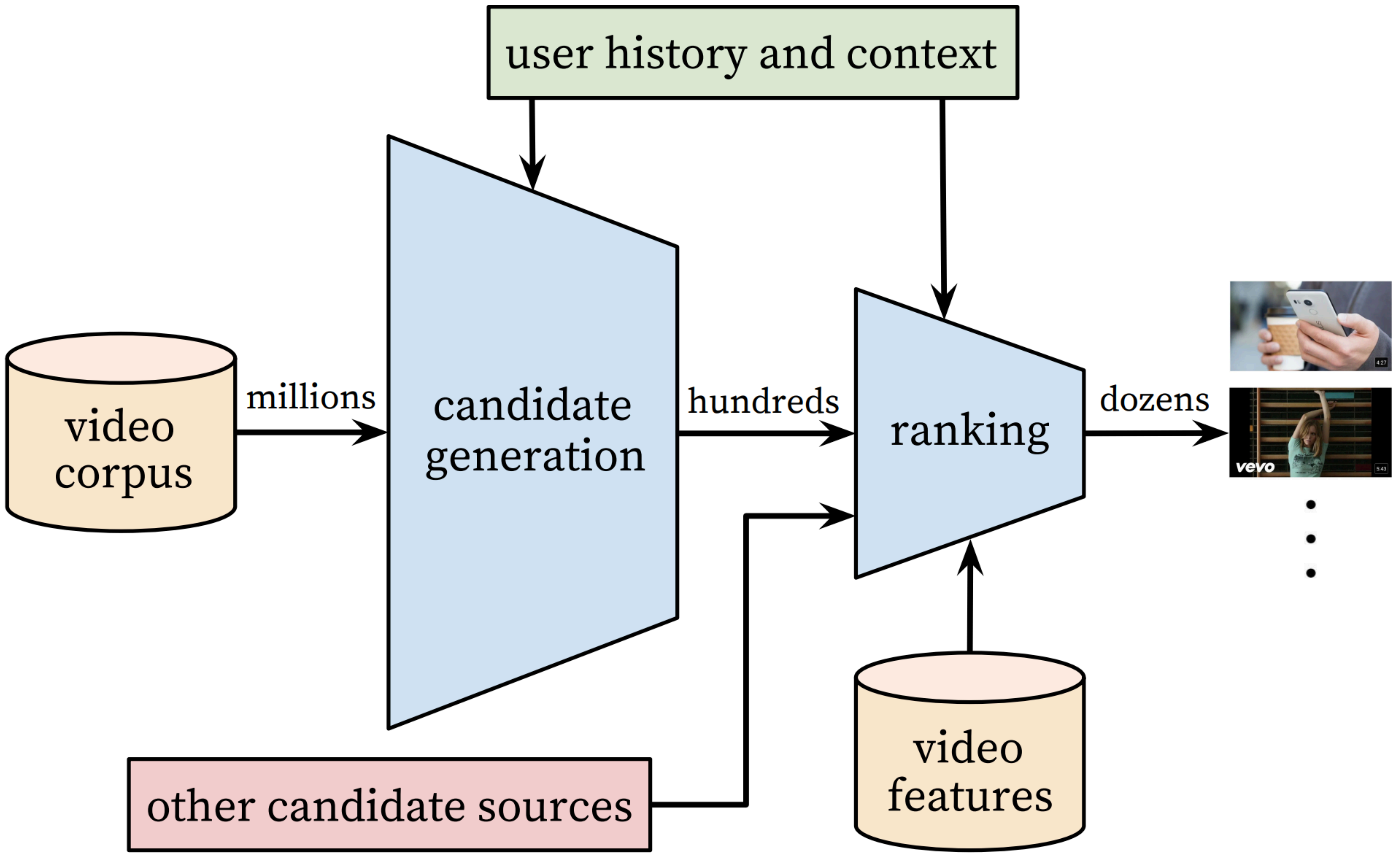


Figure 2: A schematic view of our proposed model for dynamic social recommendation.

[Song et al. 2019]

An example from Youtube



[Covington et al., ‘16]

Evaluation

- Evaluate performance on held-out data (standard)
- Splitting data into train/validation/test:
    - Split by user to give equal “weight” to each user
    - Ensure that each user has enough data (no cold-start)

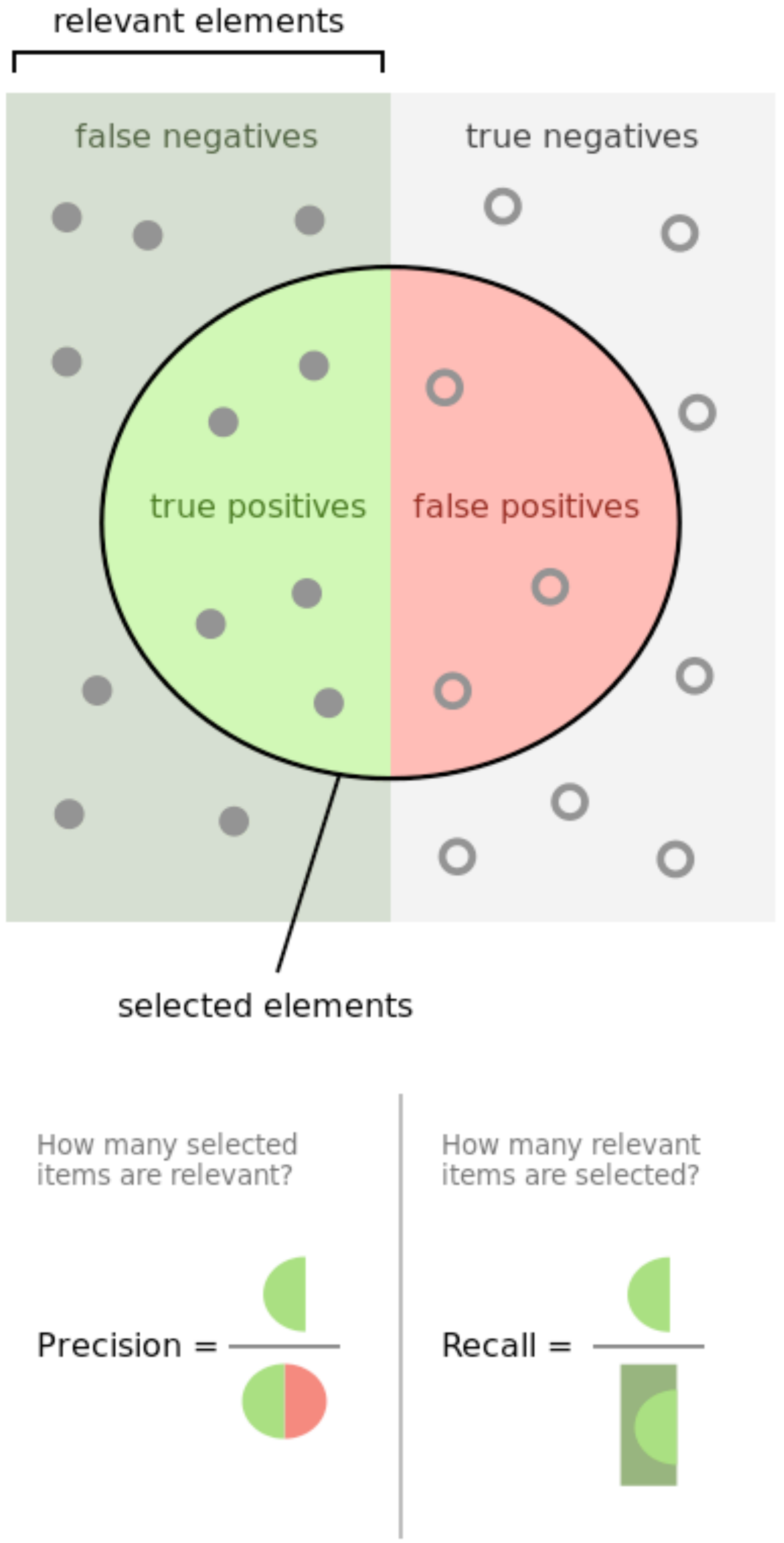
Evaluation metrics

1. **Score prediction (explicit data only)**
    - **Mean squared error:** $\frac{1}{|\mathbf{users}|}\sum_u \sum_i (\mathbf{S}_{ui} - \hat{\mathbf{S}}_{ui})^2$

Evaluation metrics

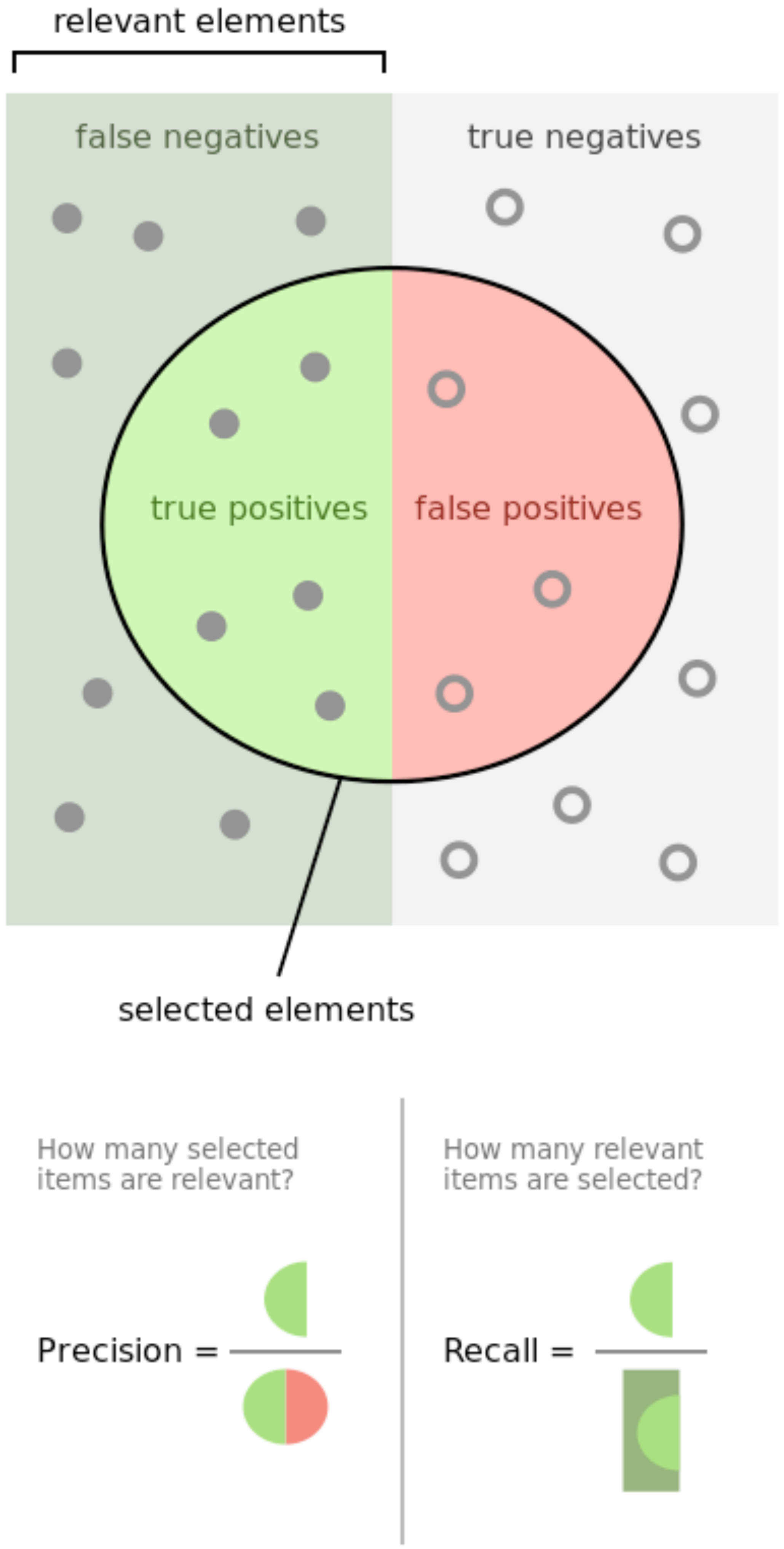
1. Score prediction (explicit data only)
    - Mean squared error: $\frac{1}{|\textbf{users}|}\sum_u \sum_i (\textbf{S}_{ui} - \hat{\textbf{S}}_{ui})^2$
2. Information retrieval
    - Precision, Recall
    - Average rank, Mean average precision
    - Normalized Discounted Cumulative Gain (NDCG)
        - Compares the ranking of your system with the optimal ranking
        - (Exponentially) Discounts items lower ranked items

Precision/Recall



[From wikipedia]

Precision/Recall

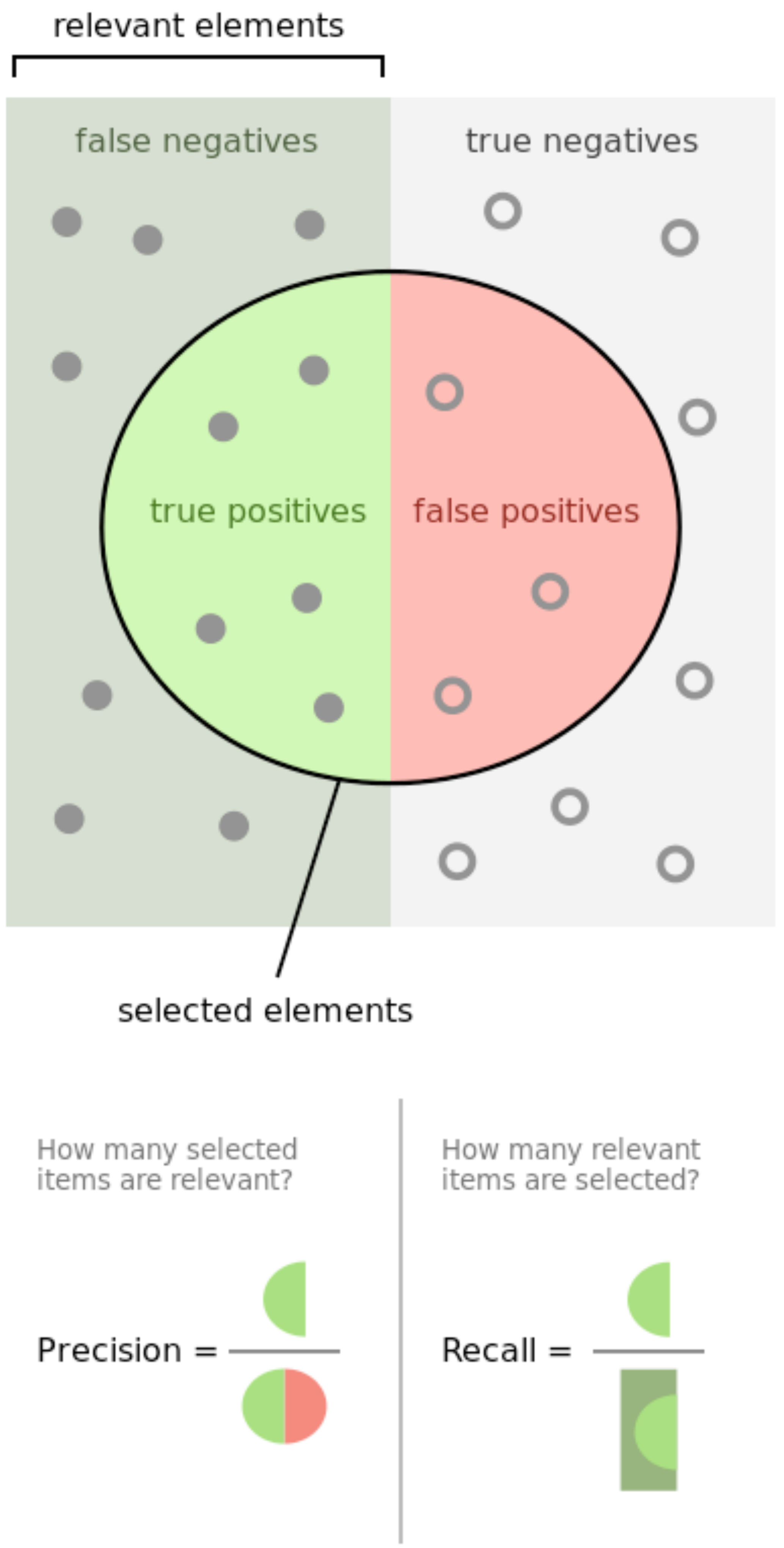


[From wikipedia]

- For implicit data recall is more appropriate

$$\text{Recall} := \frac{TP}{TP+FN}$$

Precision/Recall



[From wikipedia]

- For implicit data recall is more appropriate

$$\text{Recall} := \frac{TP}{TP+FN}$$

- Consider only the top items (Recall@K)

Other topics

- Lots of other possible signals
  - Search queries, engagement (time spent on page)
- Structured recommendations
  - E.g., Recommend a trip, a curriculum of courses

Concluding Remarks (I)

- Type of models we have discussed are useful for:
  - Domains with large number of items (and users for CF)
  - Subjective preferences over attributes (features)
    - E.g., movies and not plane tickets
  - Items can be consumed relatively fast
    - E.g., restaurants/movies and not cars/houses

Concluding Remarks (II)

- CF models “work well” especially in large-data regimes
  - Commercial systems are reasonably good
  - There is evidence that companies derive value from them
- Much progress remains to be done
  - Modelling preferences is a very active research topic
  - Good preference models gave rise to other questions

References

- Marital satisfaction and break-ups differ across on-line and off-line meeting venues, J. Cacioppo et al., PNAS'13
- A Probabilistic Model for Using Social Networks in Personalized Item Recommendation, A. Chaney et al., Recsys'15
- Deep Neural Networks for YouTube Recommendations, P. Covington et al., Recsys'16
- Content-based recommendations with Poisson factorization, P. Gopalan et al., NIPS'14
- Scalable Recommendation with Hierarchical Poisson Factorization., P. Gopalan et al., UAI'15
- Collaborative Filtering for Implicit Feedback Datasets, Hu et al., ICDM'08
- Exposure Matrix Factorization, D. Liang et al., WWW'16
- Factorization Machines, S. Rendle, ICDM '10
- Probabilistic Matrix Factorization, R. Salakhutdinov and A. Mnih, NIPS'08
- Session-based Social Recommendation via Dynamic Graph Attention Networks, W. Song et al., WSDM'19
- Collaborative Topic Modeling for Recommending Scientific Articles, C. Wang and D. Blei, KDD'11
- Deep matrix factorization models for recommender systems, HJ Xue et al., IJCAI'17