Hidden factors and hidden topics: understanding rating dimensions with review text

Julian McAuley & Jure Leskovec,

Stanford University

Overview

Reviews help to explain users' ratings, but how can they be used for recommendations?

- Reviews help us to discover the dimensions or aspects of people's opinions
- Reviews are useful at modeling new users: one review tells us much more than one rating

By combining rating & review models, we can

- Better predict ratings (in terms of the MSE)
- Automatically identify product categories
- Identify reviews that the community considers "useful"

Goal

Given a set of users and items, we want to predict each user's rating of each item:

$$rec : users \times items \rightarrow ratings$$

Common approach

products/items

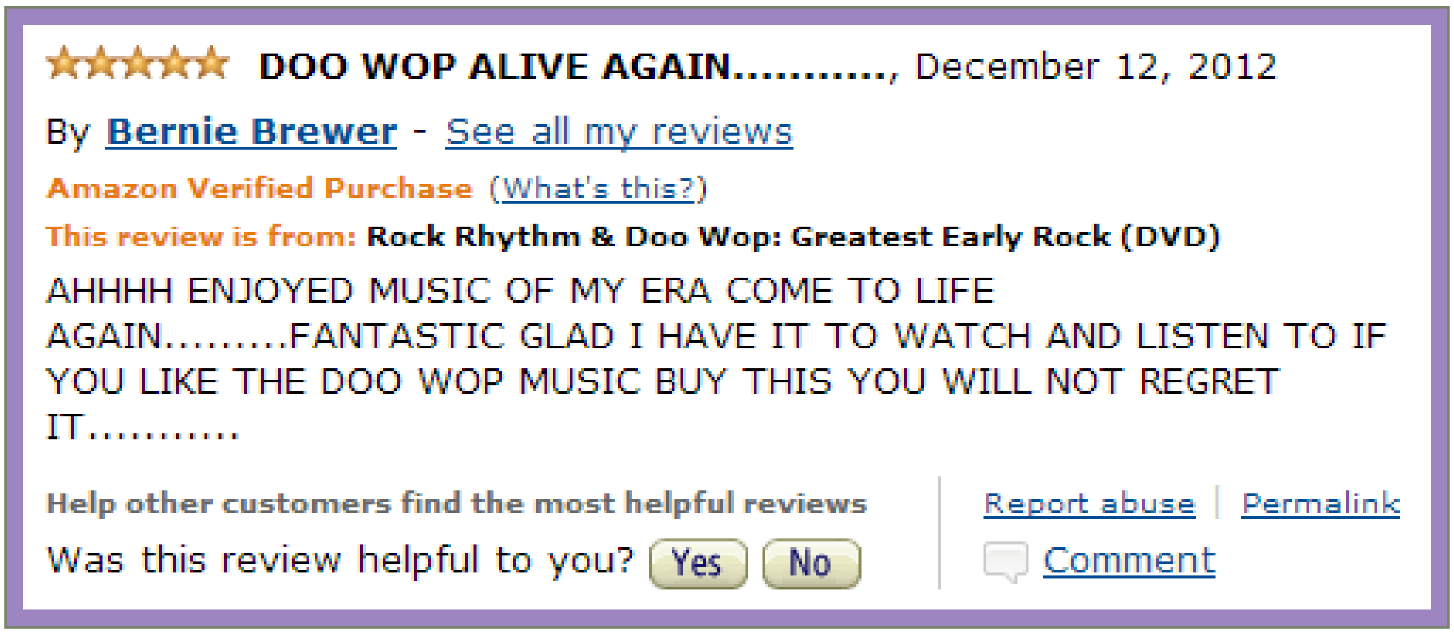
ratings

$$
\text{users} \quad
\begin{bmatrix}
\cdot & 4 & \cdot & \cdot & 3 & \cdot \\
\cdot & \cdot & 2 & 3 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & 5 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & 4 & 3 \\
\cdot & 2 & \cdot & 5 & 1 & \cdot
\end{bmatrix}
$$

This can be cast as matrix completion on a partially observed matrix of users' ratings

Product reviews

This approach ignores the text of users' reviews:



Can we make use of this rich source of data?

Are reviews actually useful?

Reviews should be useful because they tell us why a user rated a product the way they did

Are reviews actually useful?

Reviews should be useful because they tell us why a user rated a product the way they did

Reviews are hard to use because they're not available at test time

Training time: User, Item, Review → Rating

Test time: User, Item → Rating

Are reviews actually useful?

Reviews can help us model new users & items: a single review tells us more than a single rating

Reviews can help us to explain or justify users' ratings

Part 1

Low-rank models of ratings and reviews

Low-rank models of ratings

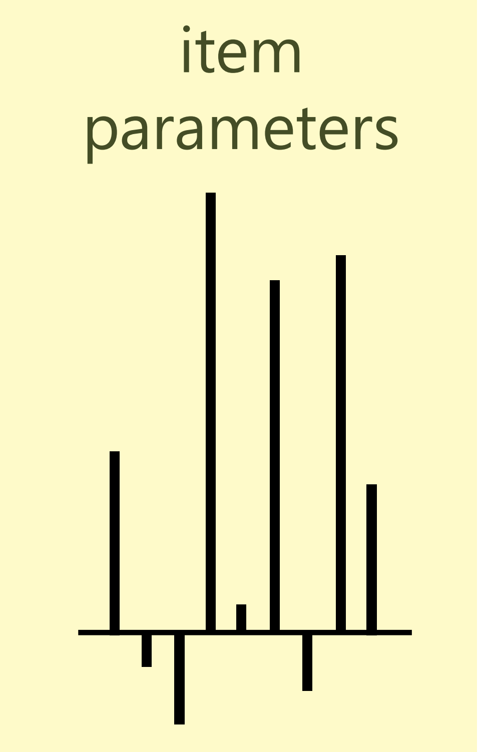
Latent factor recommender systems find low-dimensional structure of users and items:

$$rec(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

$$rec(\text{julian}, \text{Harry Potter}) \simeq [0.1, 0.3, 0.8] \cdot [0.2, 0.5, 0.7]$$

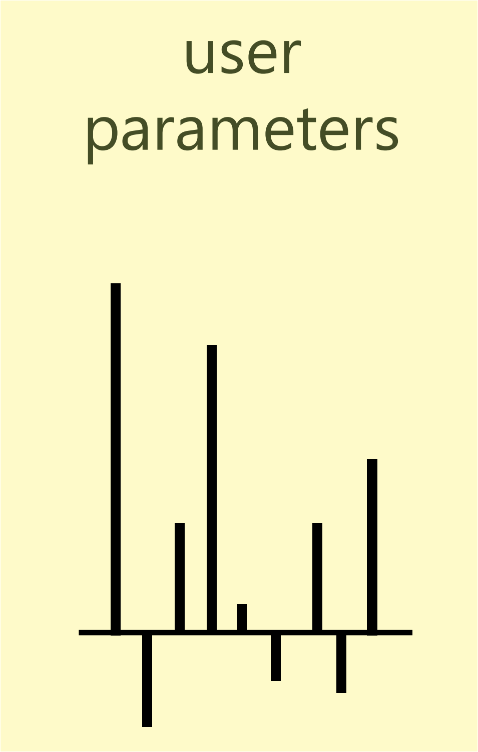
Low-rank models of ratings

Item parameters ostensibly represent **the extent to which items exhibit certain properties**



$$\gamma_i \in \mathbb{R}^K$$

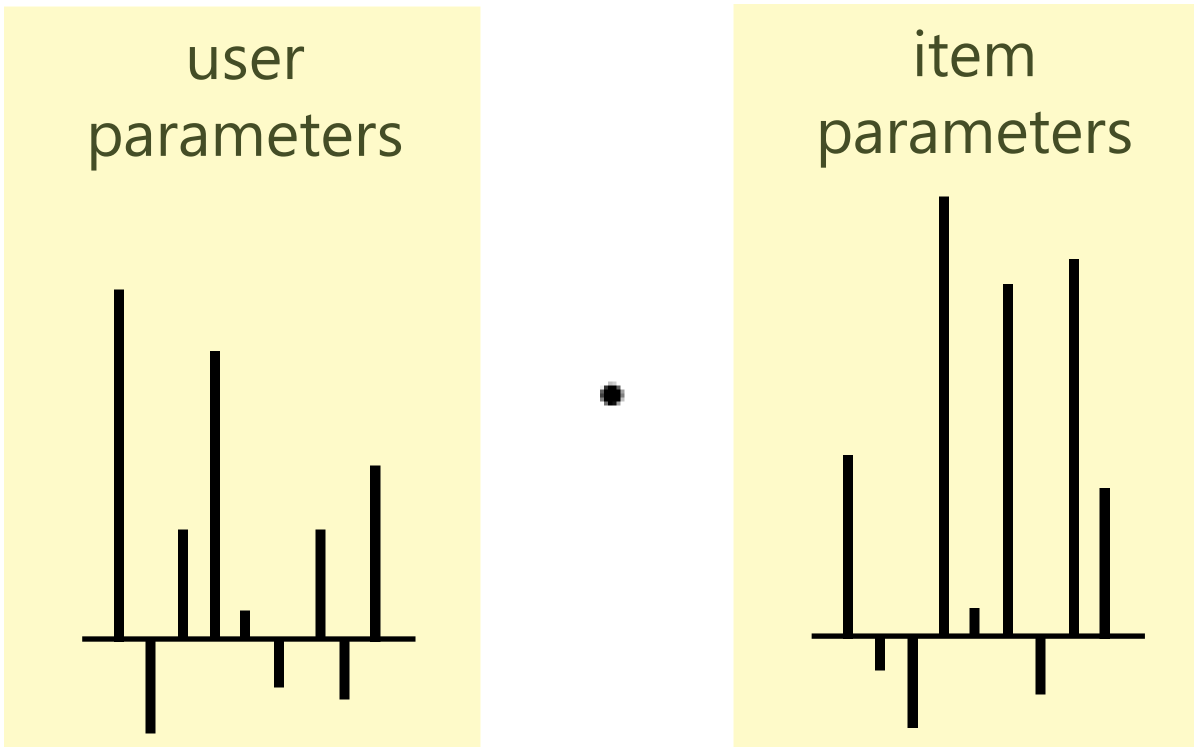
Low-rank models of ratings



$\gamma_u \in \mathbb{R}^K$

User parameters ostensibly represent **the extent to which users are attracted to those properties**

Low-rank models of ratings



$\gamma_u \in \mathbb{R}^K$ $\gamma_i \in \mathbb{R}^K$

The inner product then encodes the compatibility between the two

Low-rank models of reviews

Latent Dirichlet Allocation finds low-dimensional structure in documents

- Documents have a distribution of topics, θ
- Topics have distribution of words, ϕ

Low-rank models of reviews

Topic distributions (e.g. in LDA) represent **the extent to which certain sets of words are used in a document**

item topic distribution

$$\theta_i \in \Delta^K \text{ (i.e., } \sum_k \theta_{i,k} = 1)$$

Part 2

Combining rating and review models

Combining ratings & reviews

The parameters of a "standard" recommender system

$$rec(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

are fit so as to minimize the mean-squared error

$$\arg\min_{\alpha,\beta,\gamma} \frac{1}{|\mathcal{T}|} \sum_{r_{u,i} \in \mathcal{T}} (rec(u,i) - r_{u,i})^2 + \lambda \|\gamma\|_2^2$$

where $r_{u,i} \in \mathcal{T}$ is a training corpus of ratings

Combining ratings & reviews

We replace this objective with one that uses the review text as a regularizer:

$$\frac{1}{|\mathcal{T}|} \sum_{r_{u,i} \in \mathcal{T}} \underbrace{(rec(u,i) - r_{u,i})^2}_{\text{rating error}} - \lambda \underbrace{l(\mathcal{T}|\Theta, \phi, z)}_{\text{corpus likelihood}}$$

rating parameters

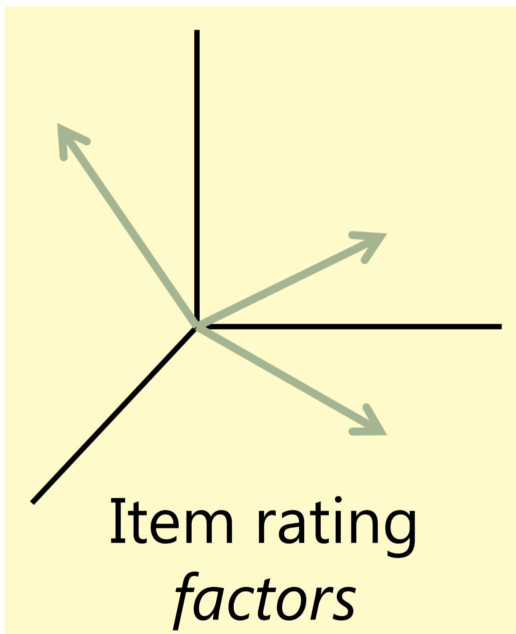
$\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i$

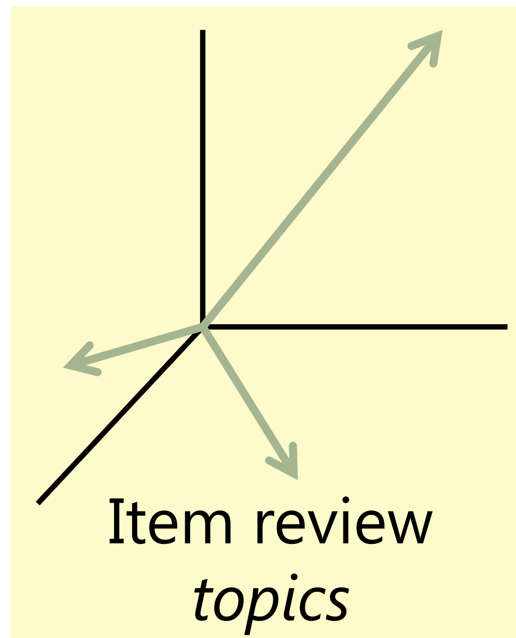
LDA parameters

Θ, ϕ, z

Combining ratings & reviews

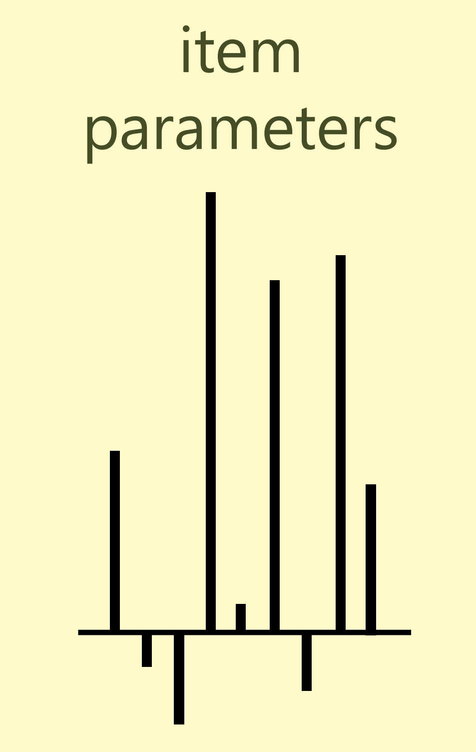
Matrix factorization and LDA project users and items into low-dimensional space

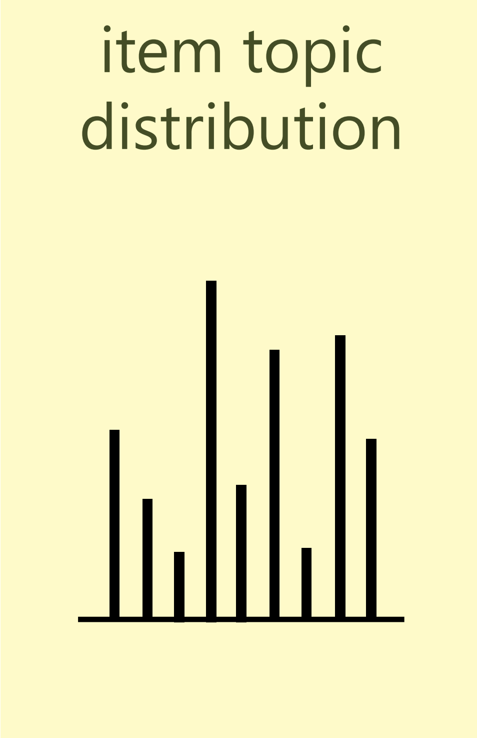




How can we align the two?

Combining ratings & reviews





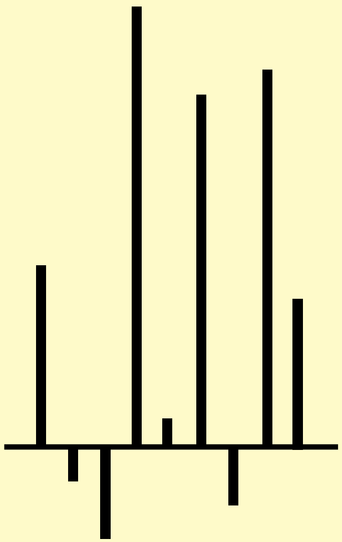
$\gamma_i \in \mathbb{R}^K$

$\theta_i \in \Delta^K$ (i.e., $\sum_k \theta_{i,k} = 1$)

We need to identify a transform between item parameters (real vectors) and topics (stochastic vectors)

Combining ratings & reviews

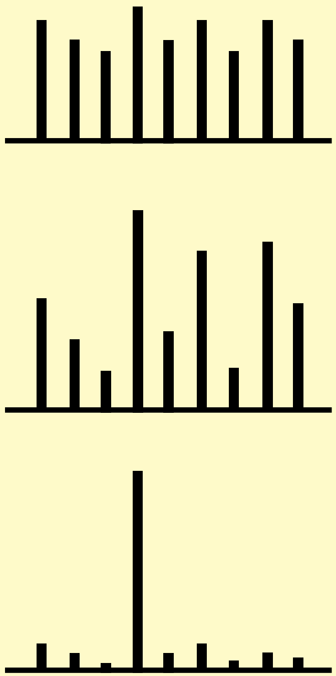
item parameters



$\gamma_i \in \mathbb{R}^K$

transform

$$\theta_{i,k} = \frac{\exp(\kappa\gamma_{i,k})}{\sum_{k'} \exp(\kappa\gamma_{i,k})}$$



$\kappa \to 0$

$\kappa \to \infty$

$$\theta_i \in \Delta^K \ (\text{i.e., } \sum_k \theta_{i,k} = 1)$$

Model fitting

Repeat steps (1) and (2) until convergence:

$$\arg\min_{\Theta} \frac{1}{|\mathcal{T}|} \sum_{r_{u,i} \in \mathcal{T}} \underbrace{(rec(u,i) - r_{u,i})^2}_{\text{rating error}} - \mu \underbrace{l(\mathcal{T}|\Theta, \phi, z)}_{\text{corpus likelihood}}$$

(solved via gradient ascent using L-BFGS)

Step 1:
minimize the MSE using gradient descent

$$\text{sample } z_{d,j} \text{ with probability } p(z_{d,j} = k) = \phi_{k,w_{d,j}}$$

(solved via Gibbs sampling)

Step 2:
sample topic assignments for each word

Part 3

Results:

recommendations

Datasets

Dataset	#Reviews	#Words
Citysearch	53K	3.94M
Yelp	230K	29.88M
Pubs (RateBeer + BeerAdvocate)	252K	31.74M
Wine (CellarTracker)	1.57M	60.02M
Beer (RateBeer + BeerAdvocate)	4.51M	349.32M
Amazon	35.28M	4.63B
Total	**41.89M**	**5.10B**

These datasets are available online at
snap.stanford.edu/data

Results (selection)

Dataset	offset	Latent factors	LDA	ours	Improvement
Amazon	1.774	1.423	1.410	1.325	**6.03%**
Beer	0.521	0.371	0.372	0.366	**1.61%**
Wine	0.043	0.029	0.029	0.027	**4.03%**
Citysearch	2.022	1.873	1.875	1.731	**7.66%**
Yelp	1.488	1.272	1.282	1.224	**4.53%**

(improvements over latent factor models are similar)

(link to complete results)

Topics - beer

pale ales	lambics	dark beers	spices	wheat beers
ipa	funk	chocolate	pumpkin	wheat
pine	brett	coffee	nutmeg	yellow
grapefruit	saison	black	corn	straw
citrus	vinegar	dark	cinnamon	pilsner
ipas	raspberry	roasted	pie	summer
piney	lambic	stout	cheap	pale
citrusy	barnyard	bourbon	bud	lager
floral	funky	tan	water	banana
hoppy	tart	porter	macro	coriander
dipa	raspberries	vanilla	adjunct	pils

Topics – amazon categories

Musical instruments:

drums	strings	wind	mics	software
cartridge	guitar	reeds	mic	software
sticks	violin	harmonica	microphone	interface
strings	strap	cream	stand	midi
snare	neck	reed	mics	windows
stylus	capo	harp	wireless	drivers
cymbals	tune	fog	microphones	inputs
mute	guitars	mouthpiece	condenser	usb
heads	picks	bruce	battery	computer
these	bridge	harmonicas	filter	mp3
daddario	tuner	harps	stands	program

Video games:

Fantasy	nintendo	windows	ea/sports	accessories
fantasy	mario	sims	drm	cable
rpg	ds	flight	ea	controller
battle	nintendo	windows	spore	cables
tomb	psp	xp	creature	ps3
raider	wii	install	nba	batteries
final	gamecube	expansion	football	sonic
battles	memory	program	nhl	headset
starcraft	wrestling	software	basketball	wireless
characters	metroid	mac	madden	controllers
ff	smackdown	sim	hockey	component

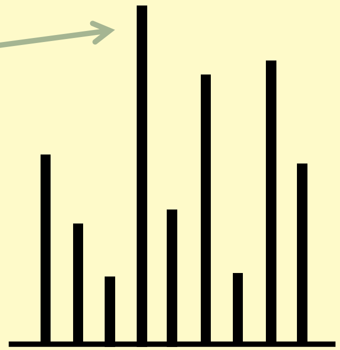
Part 4

Results: other applications

Product category discovery

1) Let each product's 'category' be $c_i = \arg\max \gamma_{i,k}$

item topic distribution



category = 4

2) Compute the best alignment between predicted and ground-truth categories (from Yelp)

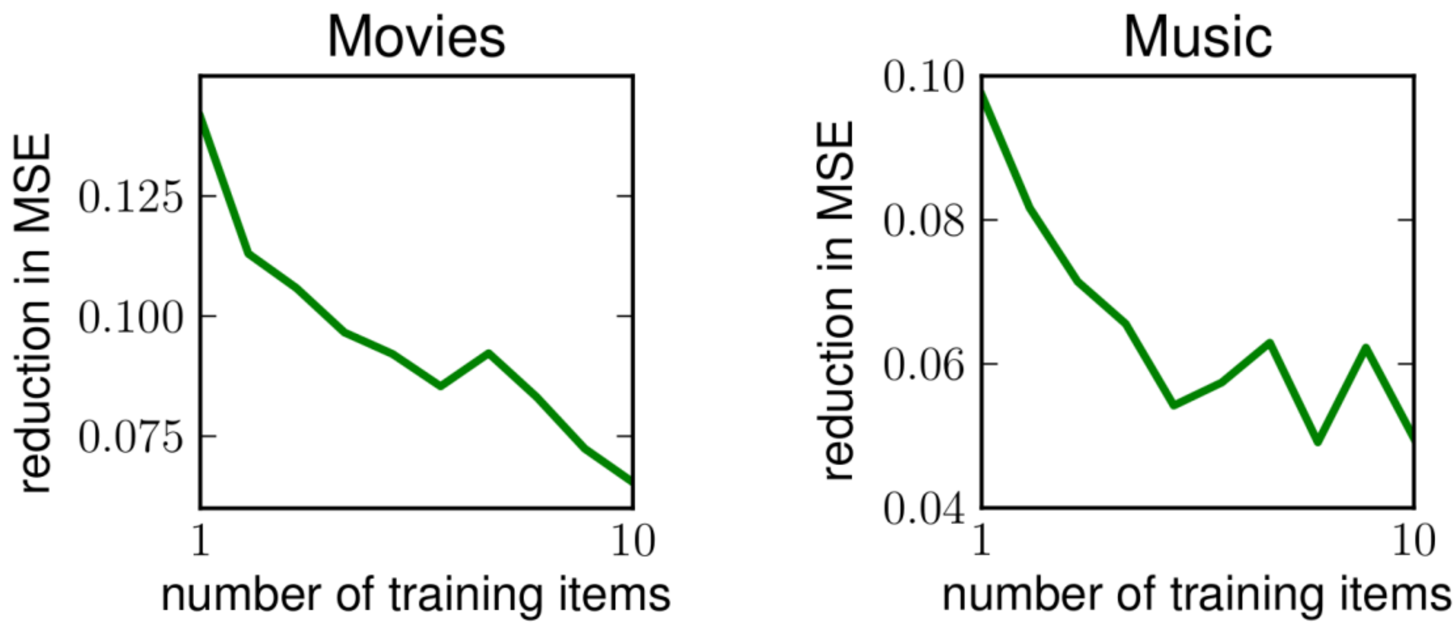
Product category discovery

We report the F1 score between the predicted categories and the ground-truth

#topics	lat. factor model	LDA	HFT (ours)	improv. vs lat. factors	improv. vs LDA
5	0.166	0.205	0.412	148%	100%
10	0.097	0.169	0.256	163%	51%
20	0.066	0.091	0.165	151%	81%
50	0.042	0.047	0.199	369%	317%

(yelp businesses)

New reviewers/items



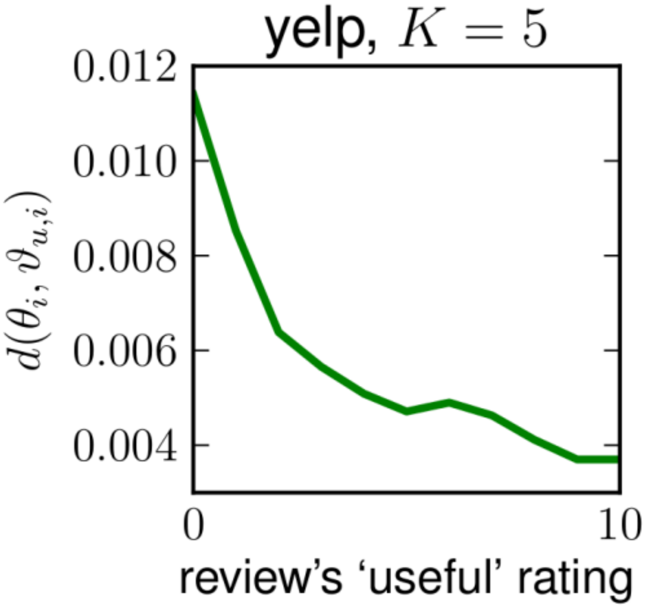
We obtain the largest improvements for users/items with few reviews

Identifying good reviewers

topics the community considers important for an item

$$d(\theta_i, \vartheta_{u,i})$$

topics discussed in a particular user's review of the item



'Useful' reviews are those that discuss each topic in proportion to its importance

Related work

- Latent factor models & LDA: Blei & McAuliffe (2007), Lin & He (2009), Koren and Bell (2011)
- Aspects: Blair-Goldensohn et al. (2008), Ganu et al. (2009), Titov & McDonald (2008), Lerman et al. (2009), Wang et al. (2010)
- Automatic aspect discovery: Zhao et al. (2010), Moghaddam & Ester (2011), Popescu & Etzioni (2005)

Conclusion

1. We discovered "topics" that simultaneously explain variation in ratings and reviews
2. A small number of reviews tells us more about a user/item than a small number of ratings
3. Our model outperforms alternatives on a variety of large-scale recommendation datasets
4. Our model allows us to automatically discover product categories, and to identify useful reviews

Code and data is available online!
Code: http://i.stanford.edu/~julian/
Data: http://snap.stanford.edu/data/web-Amazon-links.html

Conclusion

1. We discovered "topics" that simultaneously explain variation in ratings and reviews
2. A small nu … about a user/item … s
3. Our mode … a variety of large-scal …
4. Our mode … scover product c … ul reviews

Thanks!

Code and data is available online!
Code: http://i.stanford.edu/~julian/
Data: http://snap.stanford.edu/data/web-Amazon-links.html