Does Content Matter?

November 15, 2016

Malcolm Slaney

Not a Google project!

Does Content Matter?

Recommending Movies
- Yehuda Koren (Netflix Competition)

Music Similarity
- Malcolm Slaney (ISMIR 2007)

Tagging Images
- Dhruv Mahajan (ACM Multimedia 2010)

Early vs. Late Fusion
- Conclusions

Netflix Competition

Create new recommendation algorithm
- 10% better than Netflix algorithm

Data
- 100M ratings
- 480k users, 17k movies

Winner
- Gradient Boosted Decision Trees
- Hundreds of features

NO content features!!!!

Movie rating data

Training data

User	Movie	Score
1	21	1
1	213	5
2	345	4
2	123	4
2	768	3
3	76	5
4	45	4
5	568	1
5	342	2
5	234	2
6	76	5
6	56	4

Test data

User	Movie	Score
1	62	?
1	96	?
2	7	?
2	3	?
3	47	?
3	15	?
4	41	?
4	28	?
5	93	?
5	74	?
6	69	?
6	83	?

Baseline Predictors

Minimize

- Error +
- Coefficient sizes

Average for all items

Average for user u

Average for item i

$$\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_3\left(\sum_u b_u^2 + \sum_i b_i^2\right)$$

True rating for item i by user u

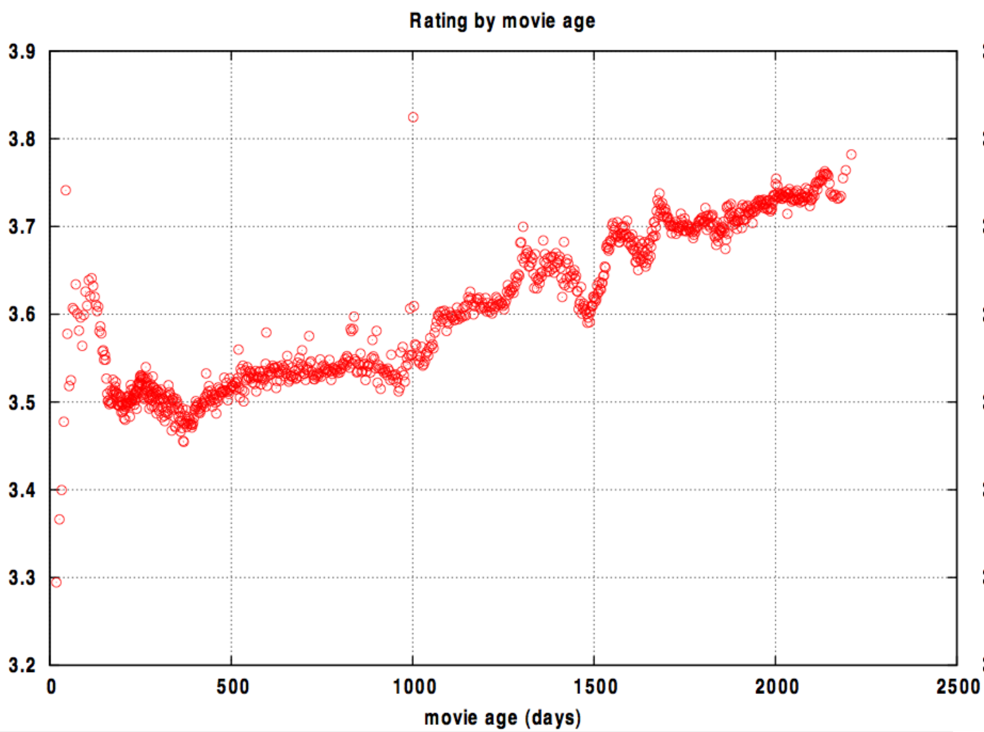
Regularization Parameter

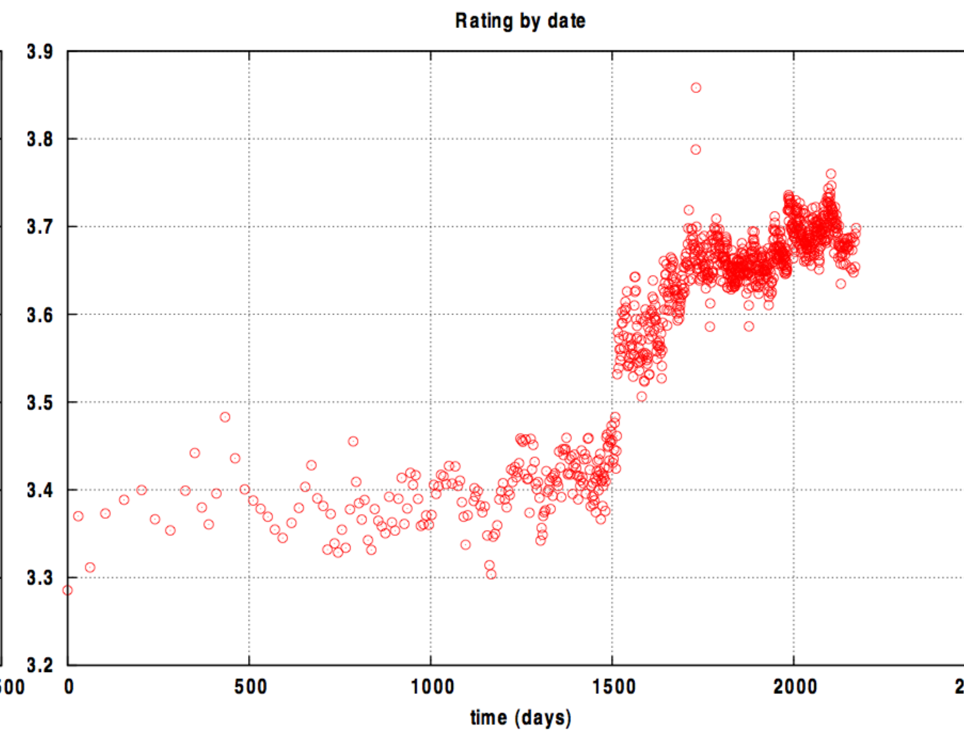
Netflix Temporal Effects

Ratings change with

- Age (older are rated higher)
- Time (big average change in early 2004)

No content features!



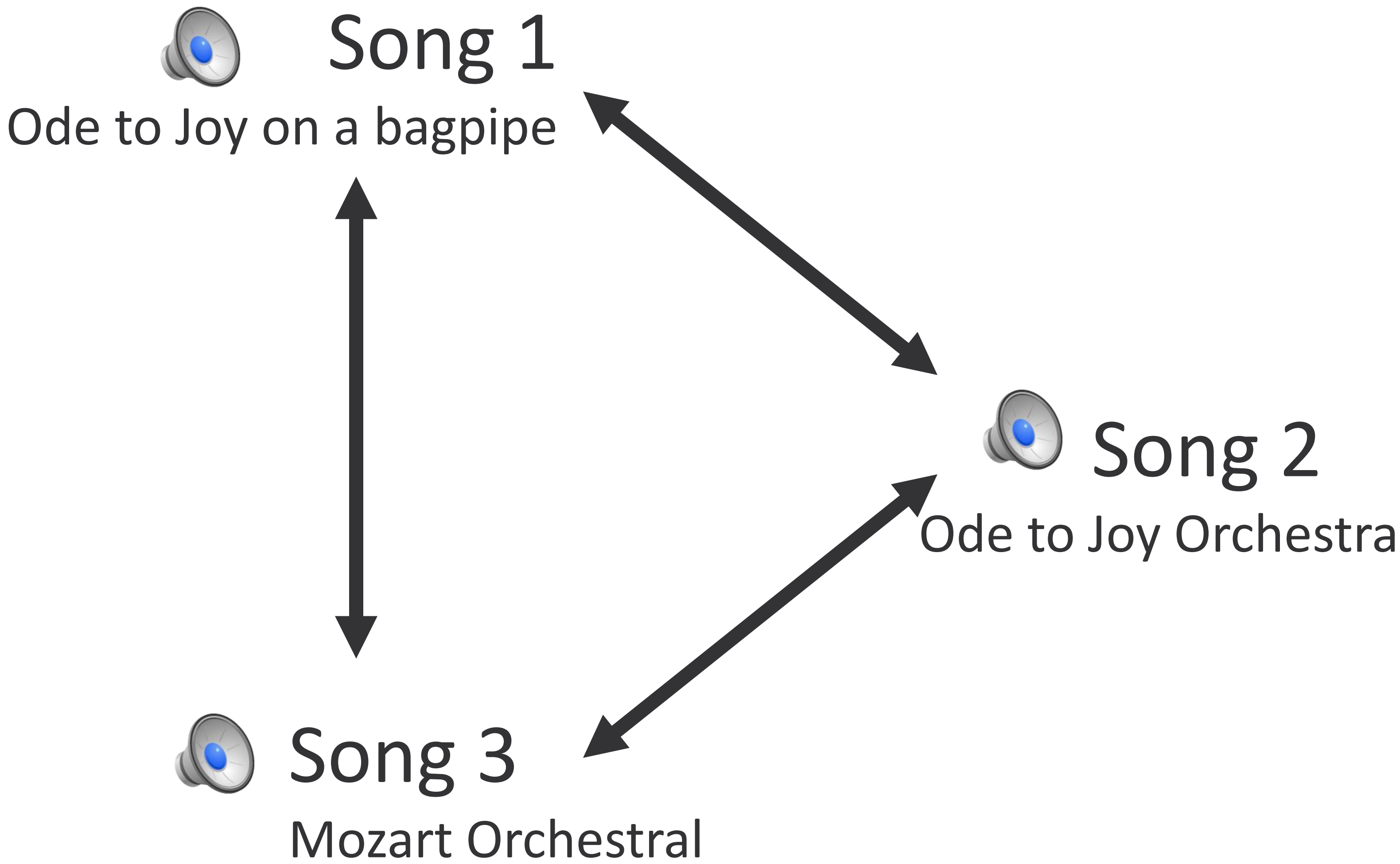


Music Similarity

Are these two songs similar?

Took me 1.5 hours

Most Similar?



Context

Song

Rater

Song Similarity Example

	Song 1	Song 2	Song 3
Jazz Lover	5	0	5
Rock Lover	5	0	5
Classical Lover	0	5	0

Similar Songs

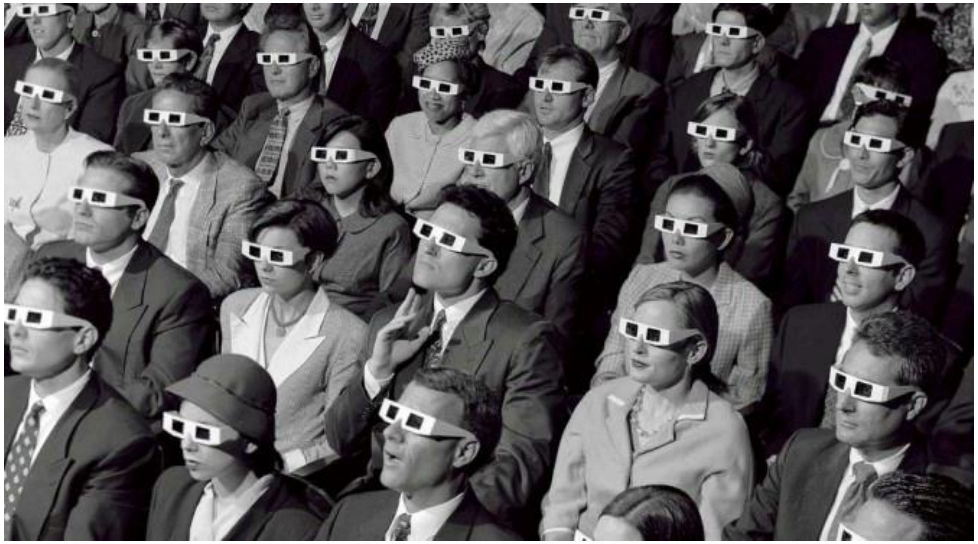
Like an anchor model (from speaker ID) or beacon model (from CS)

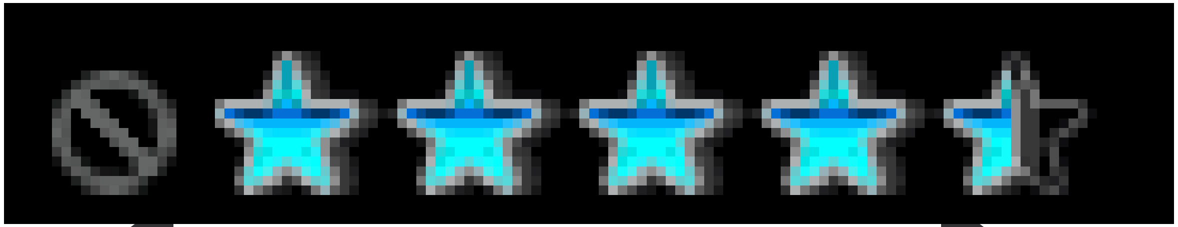
Our Experiment

380,911 Subjects

1000 Jazz Songs

1,449,335 Ratings





Never Play this Again

Love It!

Similarity User Tests

Which playlist is most similar?

Approach	Most Similar Votes	Least Similar Votes
Random	1	13
Content Based	1	4
Rating Based	**16**	**1**

Tagging Images

Labeling is hard!

- ESP Game: Perhaps >10 guesses

Small differences matter!

www.catrescue.com

www.doglovers.com

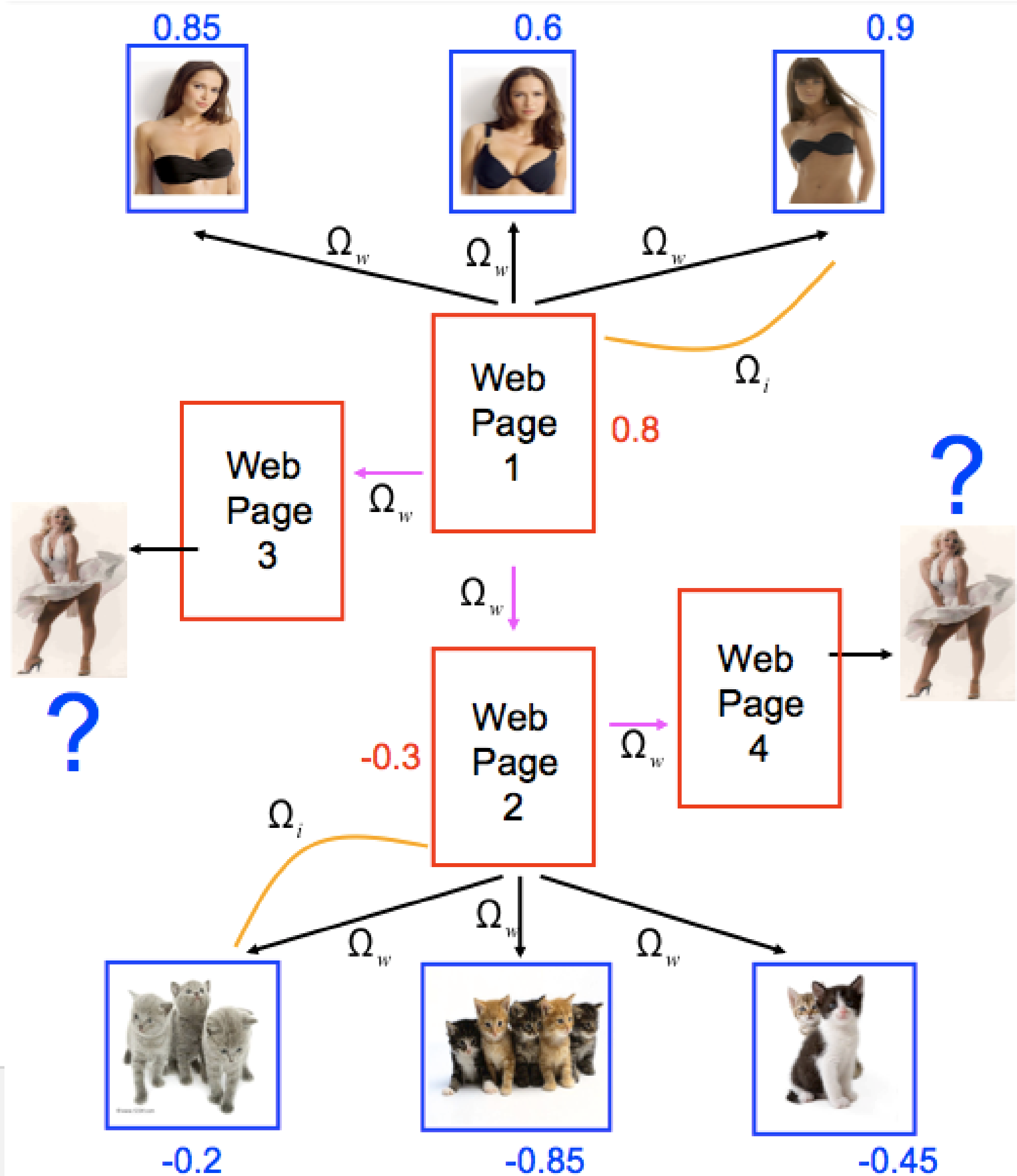
Web-Graph

Context matters

Web neighbors matter

Images drive pages

Semi-supervised learning



Graph Method

Optimize loss (and regularize)

$$\Omega(w,z) = \Omega_s(w,z) + \Omega_w(w,z) + \Omega_i(w,z)$$

Enforce continuity across directed web graph edges

Propagate image score to webpage

Experiment

Connected subgraph of entire web

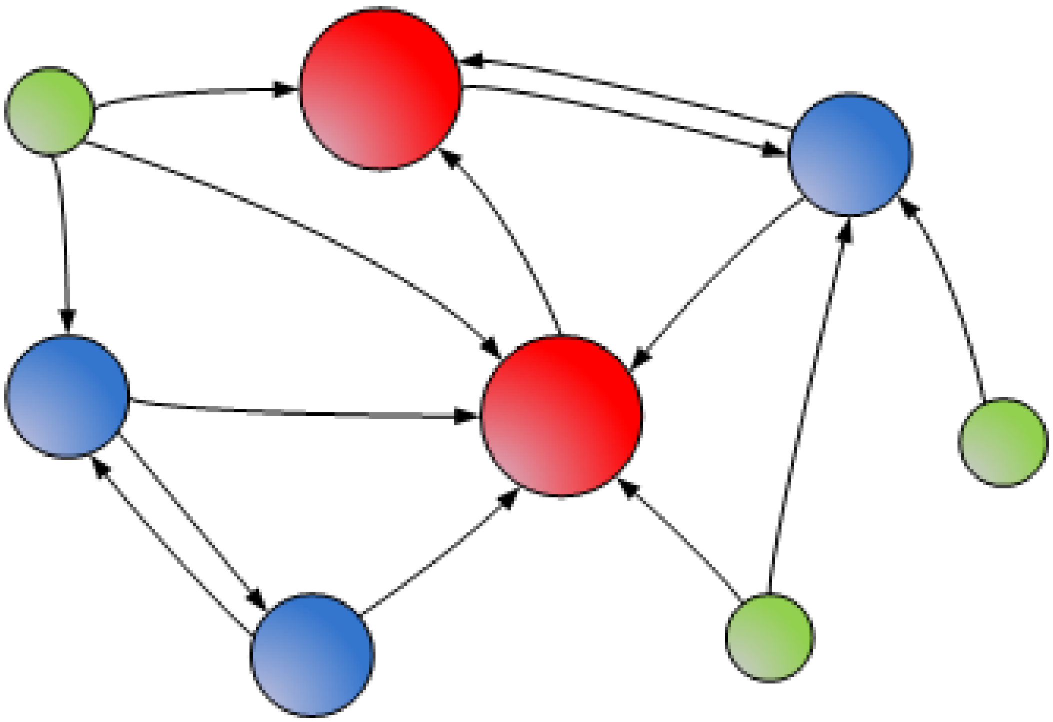
- 82k web pages
- 211k attached images

Labeled Data

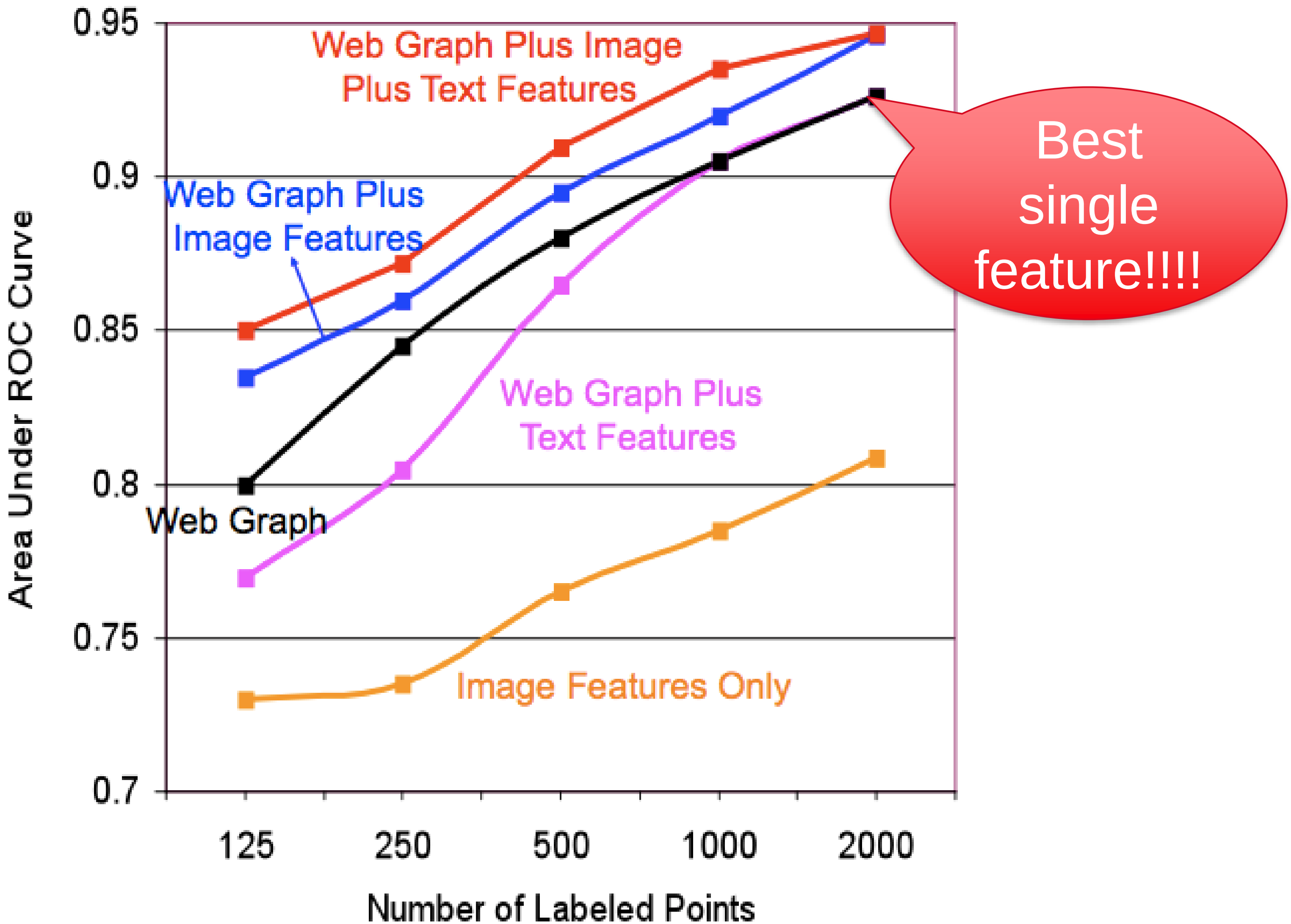
- 1291 Positive
- 1405 Negative

Image Features

- 500-d deep belief network (DBN)
- Small by today's standards



Tagging Performance



Does Content Matter?

Recommending Movies

- Yehuda Koren (Netflix Competition)

Music Similarity

- Malcolm Slaney (ISMIR 2007)

Tagging Images

- Dhruv Mahajan (ACM Multimedia 2010)

Early vs. Late Fusion

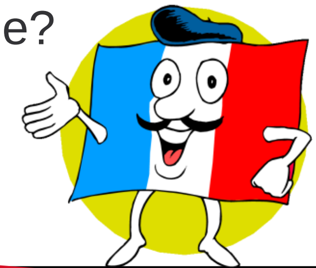
- Conclusions

Early vs. Late Fusion

Late Fusion
- Easier (less memory)

Early Fusion
- Better performance

Late Fusion Example—Is this person an oddball?

	Odd because likes smelly cheese? 	Odd because of home? 	Late Fusion (combine by multiplication)
Midwest Farm Boy 	40%	10%	4%
Parisian 	40%	10%	

Information lost with each decision!!!

Thank You

malcolm@ieee.org

Does Content Matter?

Recommending Movies

- Yehuda Koren (Netflix Competition)

Music Similarity

- Malcolm Slaney (ISMIR 2007)

Tagging Images

- Dhruv Mahajan (ACM Multimedia 2010)

Early vs. Late Fusion

- Conclusions

Movie rating data

- Training data
  - 100 million ratings
  - 480,000 users
  - 17,770 movies
  - 6 years of data: 2000-2005
- Test data
  - Last few ratings of each user (2.8 million)
- Dates of ratings are given

Training data

user	movie	score
1	21	1
1	213	5
2	345	4
2	123	4
2	768	3
3	76	5
4	45	4
5	568	1
5	342	2
5	234	2
6	76	5
6	56	4

Test data

user	movie	
1	62	?
1	96	?
2	7	?
2	3	?
3	47	?
3	15	?
4	41	?
4	28	?
5	93	?
5	74	?
6	69	?
6	83	?

Bottom Line

Gradient Boosted Decision Trees

- Find weightings and best features
- All features/predictors
  - 454+75+24
- Additive regression model

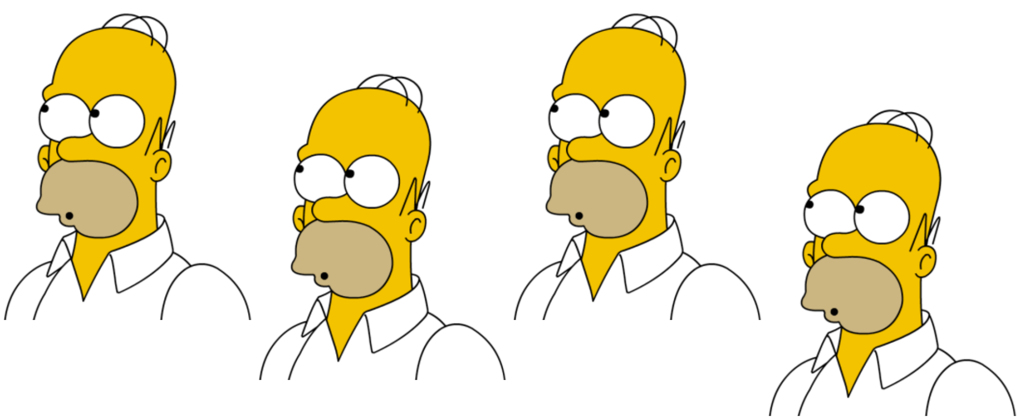
NO content features!!!!

Does Content Matter?

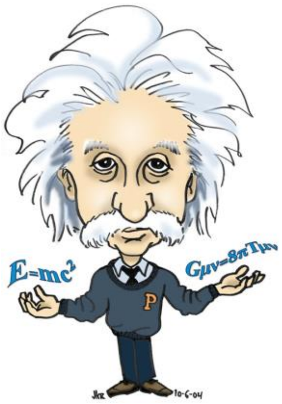
Yes, but how?

Leverage human signals

1B users are smarter than 1 Multimedia PhD

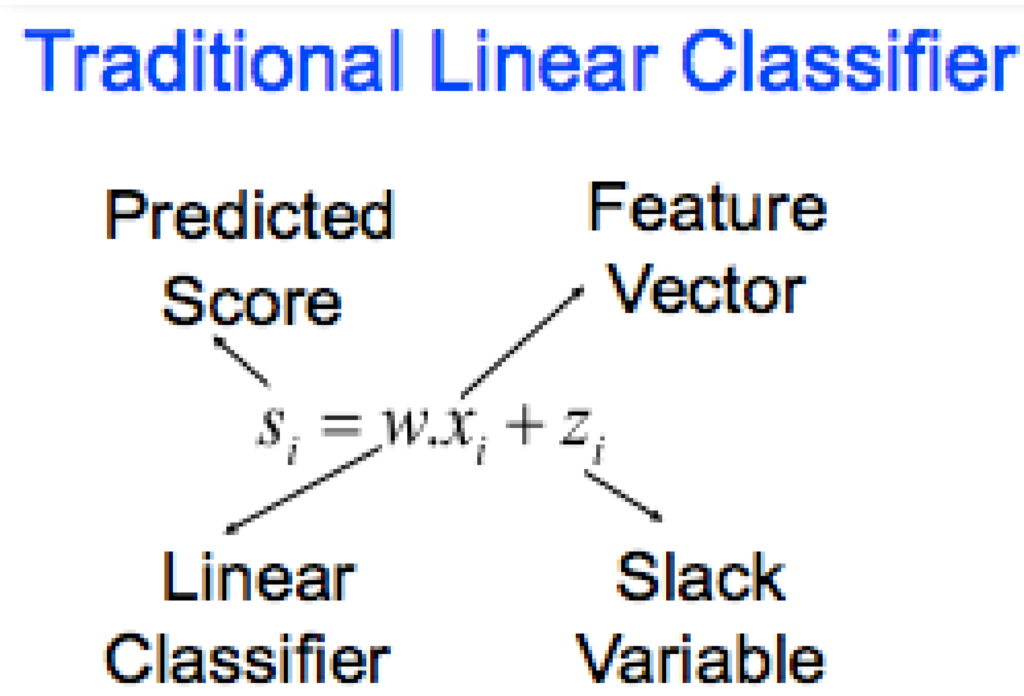


>



Linear Classifier

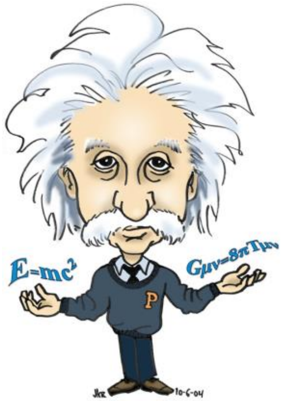
Simplest Classifier



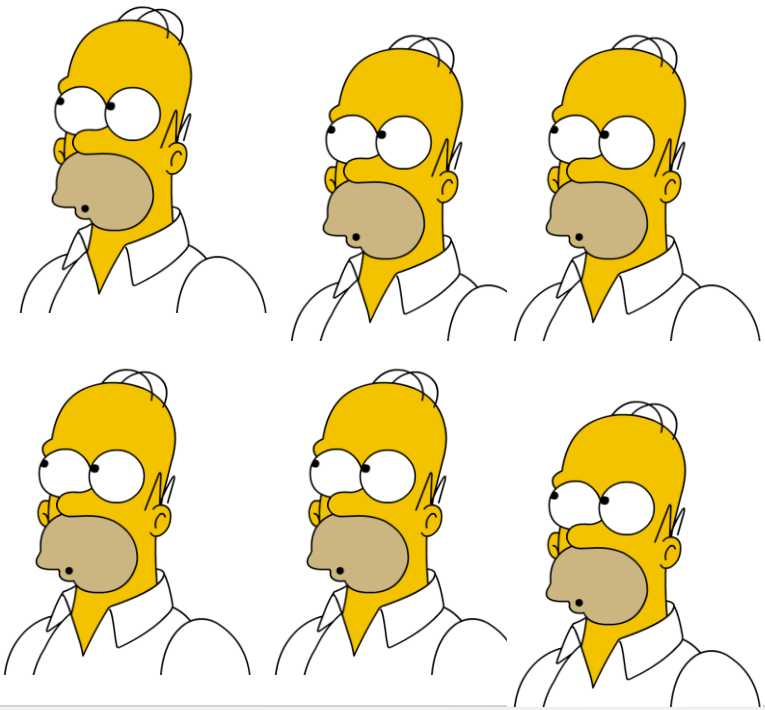
Today’s Theme

One not so bright

Be very smart



Or use lots of data and simple classifiers

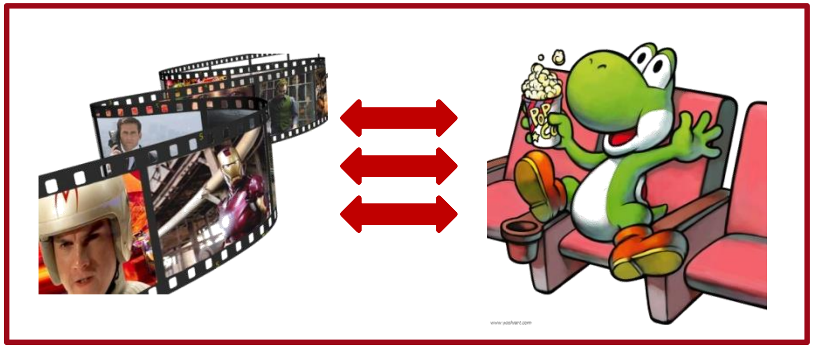


Components of a rating predictor

user bias

movie bias

user-movie interaction



Baseline predictor

- Separates users and movies
- Often overlooked
- Benefits from insights into users' behavior
- Among the main practical contributions of the competition

User-movie interaction

- **Characterizes the matching between users and movies**
- **Attracts most research in the field**
- **Benefits from algorithmic and mathematical innovations**

Factorization Model

Find hidden factors

Can use explicit (stars) or implicit data (viewed)

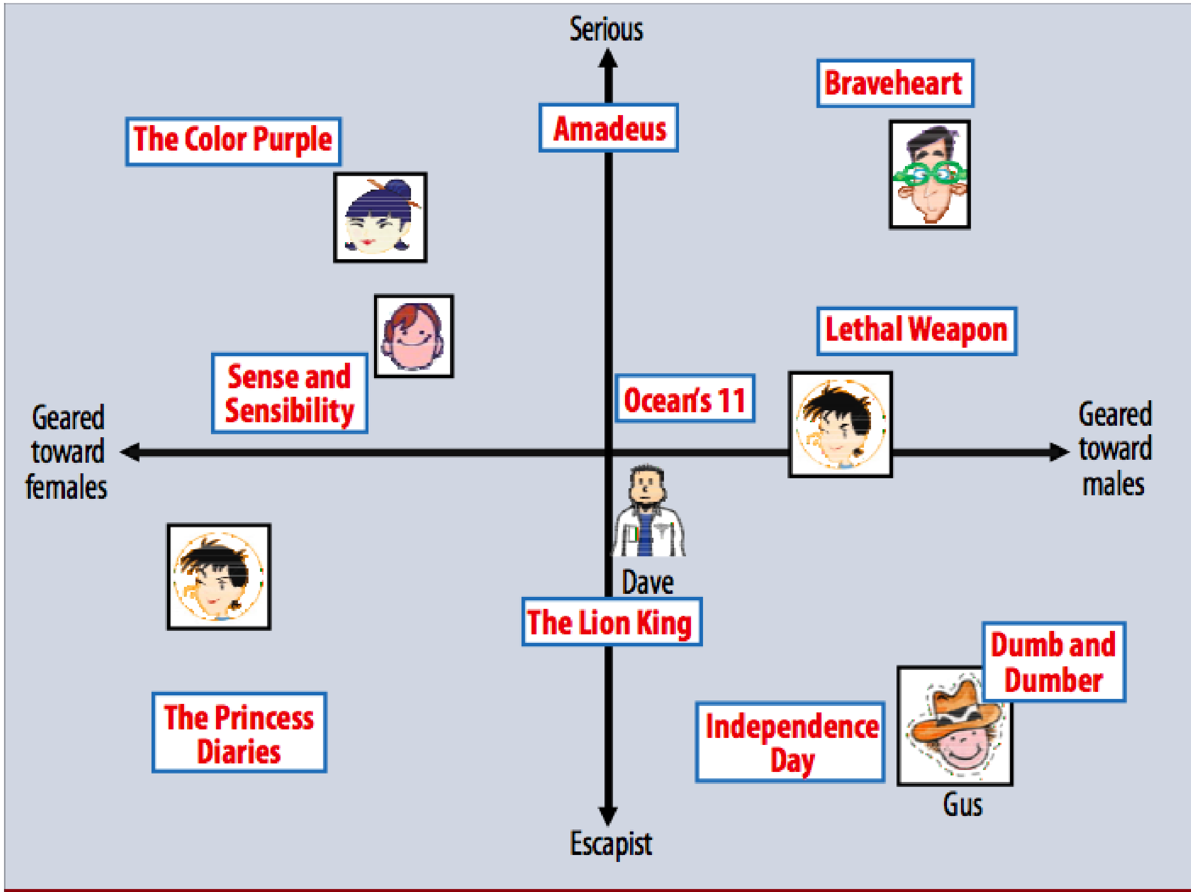


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

From: Yehuda Koren, Robert Bell, Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, August, 2009.

Estimate unknown ratings as inner-products of factors

users

items

1		3			5			5		4	
		5		2.4		4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

~=

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

●

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

A rank-3 SVD approximation

Courtesy of YehudaKoren

Neighborhood Models

Find similar users (or items)

Weighted average

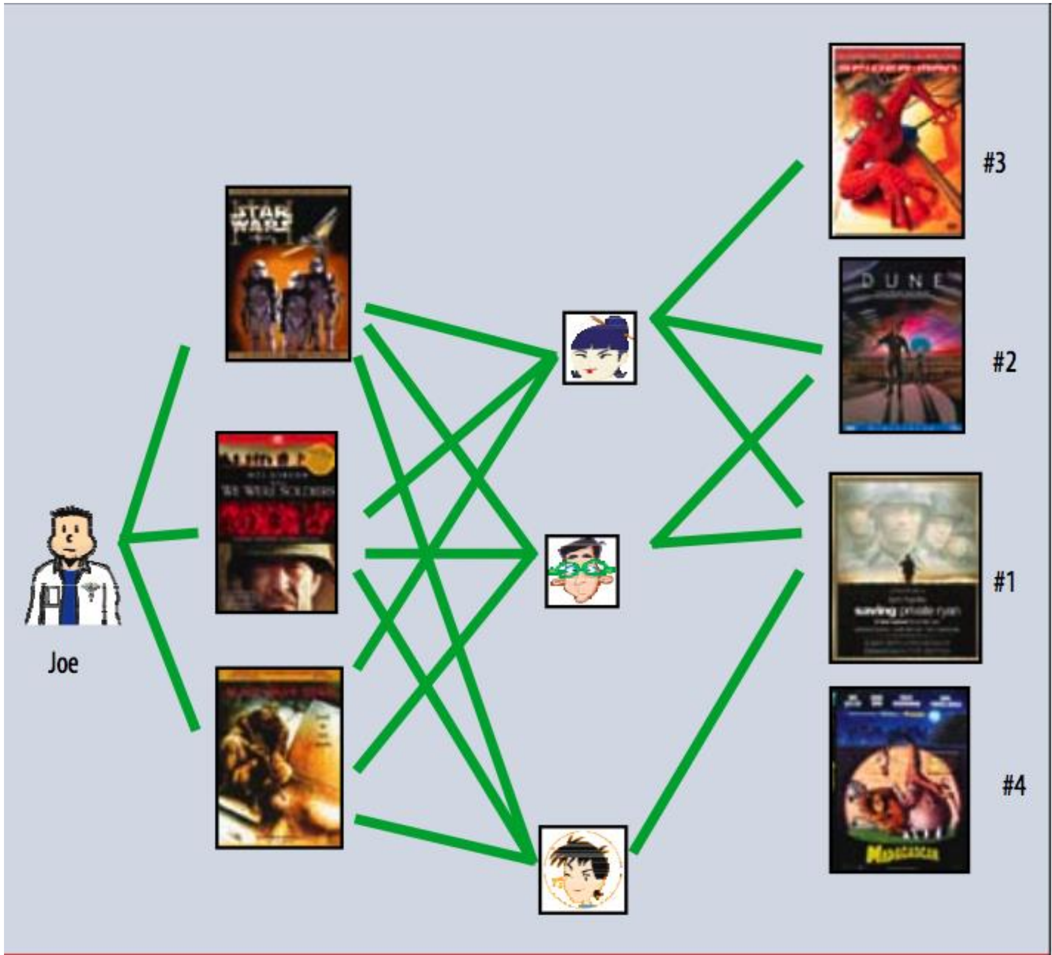


Figure 1. The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked *Saving Private Ryan*, so that is the first recommendation. Two of them liked *Dune*, so that is next, and so on.

From: YehudaKoren, Robert Bell, Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, August, 2009.

Neighborhood Math

1. Define a similarity measure between items: s_{ij}
2. Select neighbors – $N(i;u)$:
   K items most similar to i, that were rated by u
3. Estimate unknown rating, r_{ui}, as the weighted average:

$$\hat{r}_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} r_{uj}}{\sum_{j \in N(i;u)} s_{ij}}$$

- Results are improved when normalizing data

Courtesy of

Neighborhood modeling through global optimization

A basic model:

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in R(u)} \left(r_{uj} - b_{uj} \right) w_{ij}$$

Need to estimate rating of user u for item i

Baseline estimate

Set of items rated by u

Deviation from baseline estimate for item j

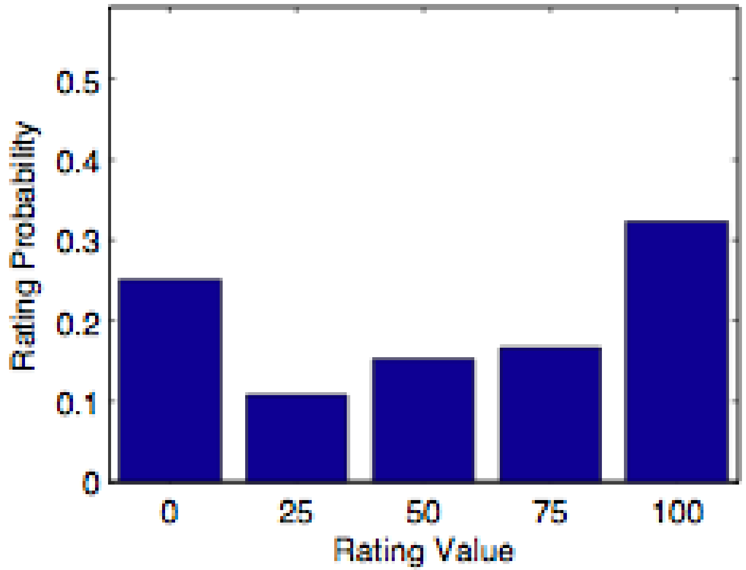
Offset from j to i

Constants

learned from the data through optimization

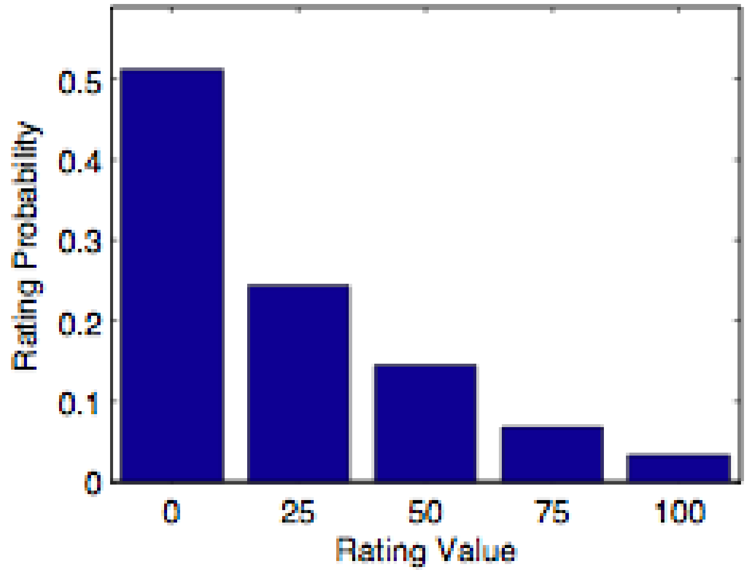
Users do not rate everything....

Self-Selected Rating Histogram



(1.5B ratings)

True Rating Histogram



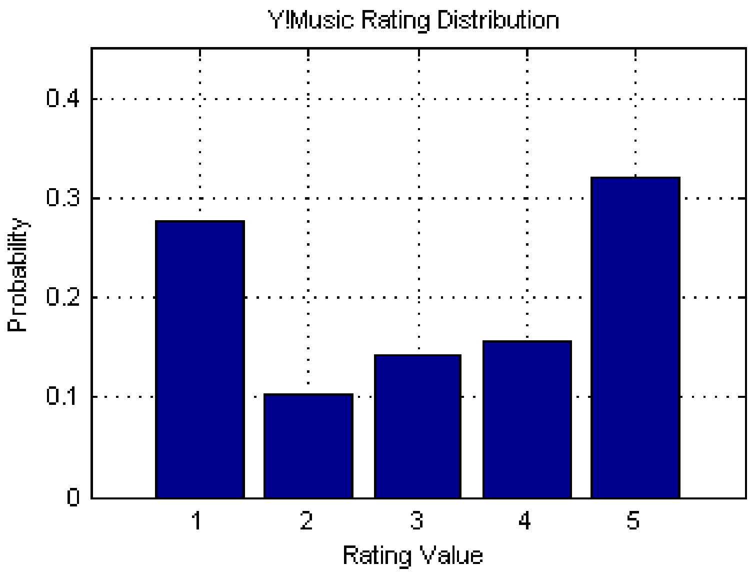
(350k ratings)

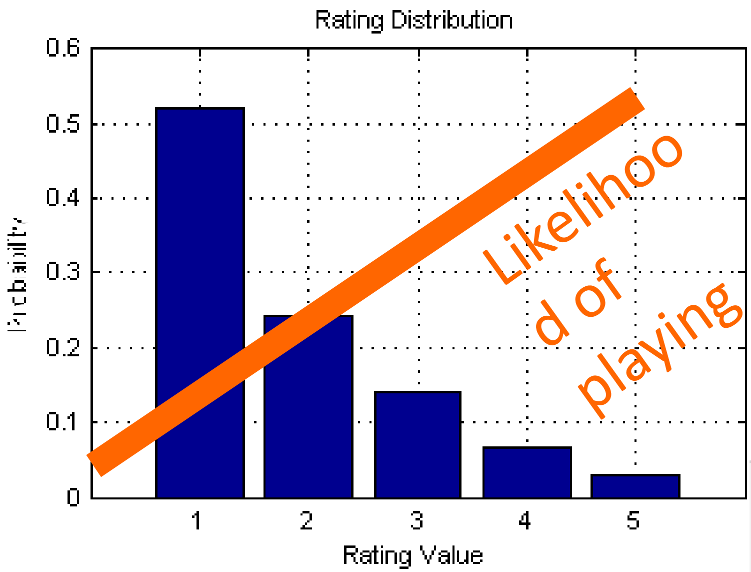
From: Marlin, Zemel, Roweis, Slaney. "Collaborative Filtering and the missing at random assumption." UAI 2007

About the Data

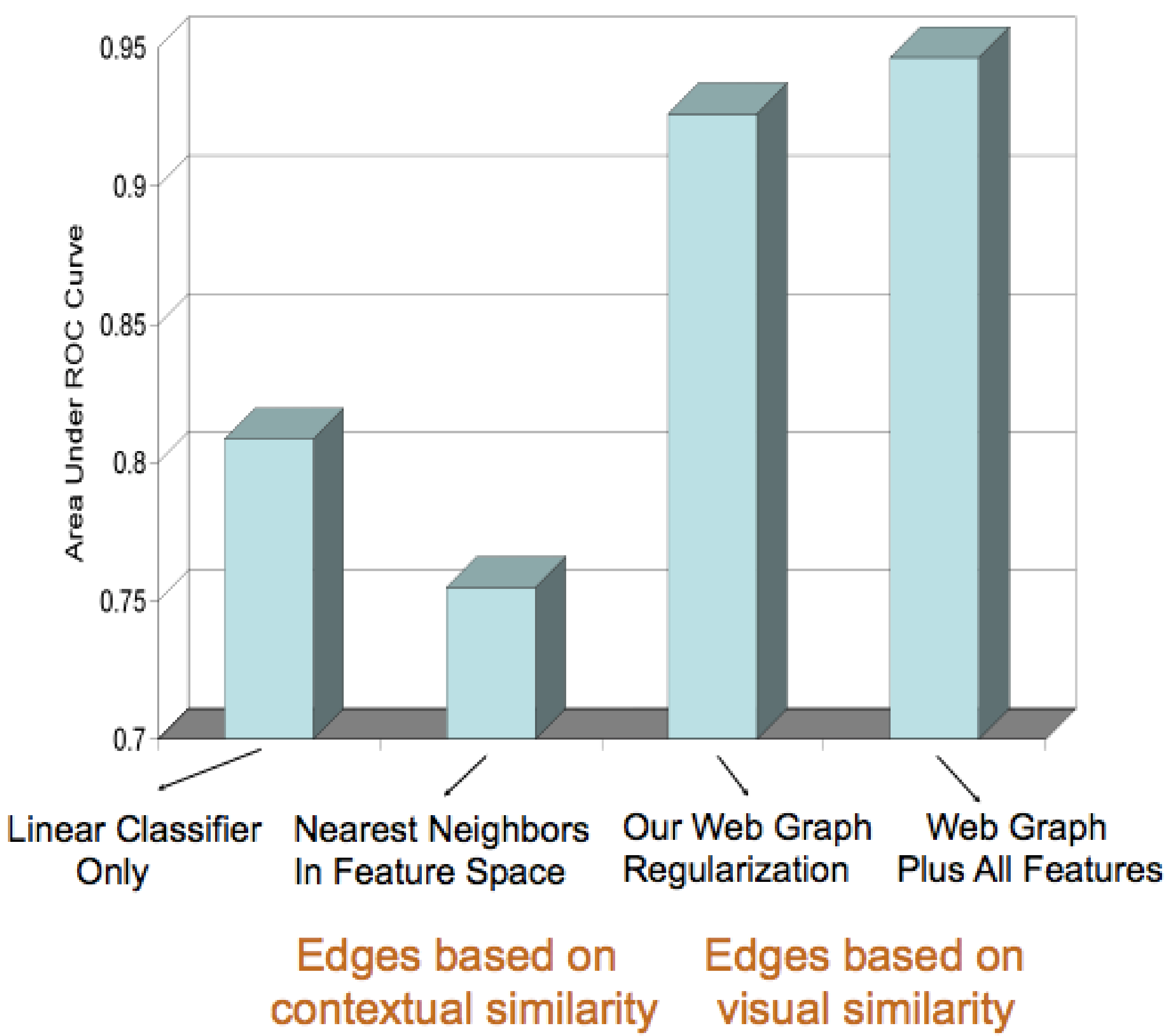
Real rating data
- Y! Music
- 700M ratings

Random ratings
- 35k subjects
- 350k ratings

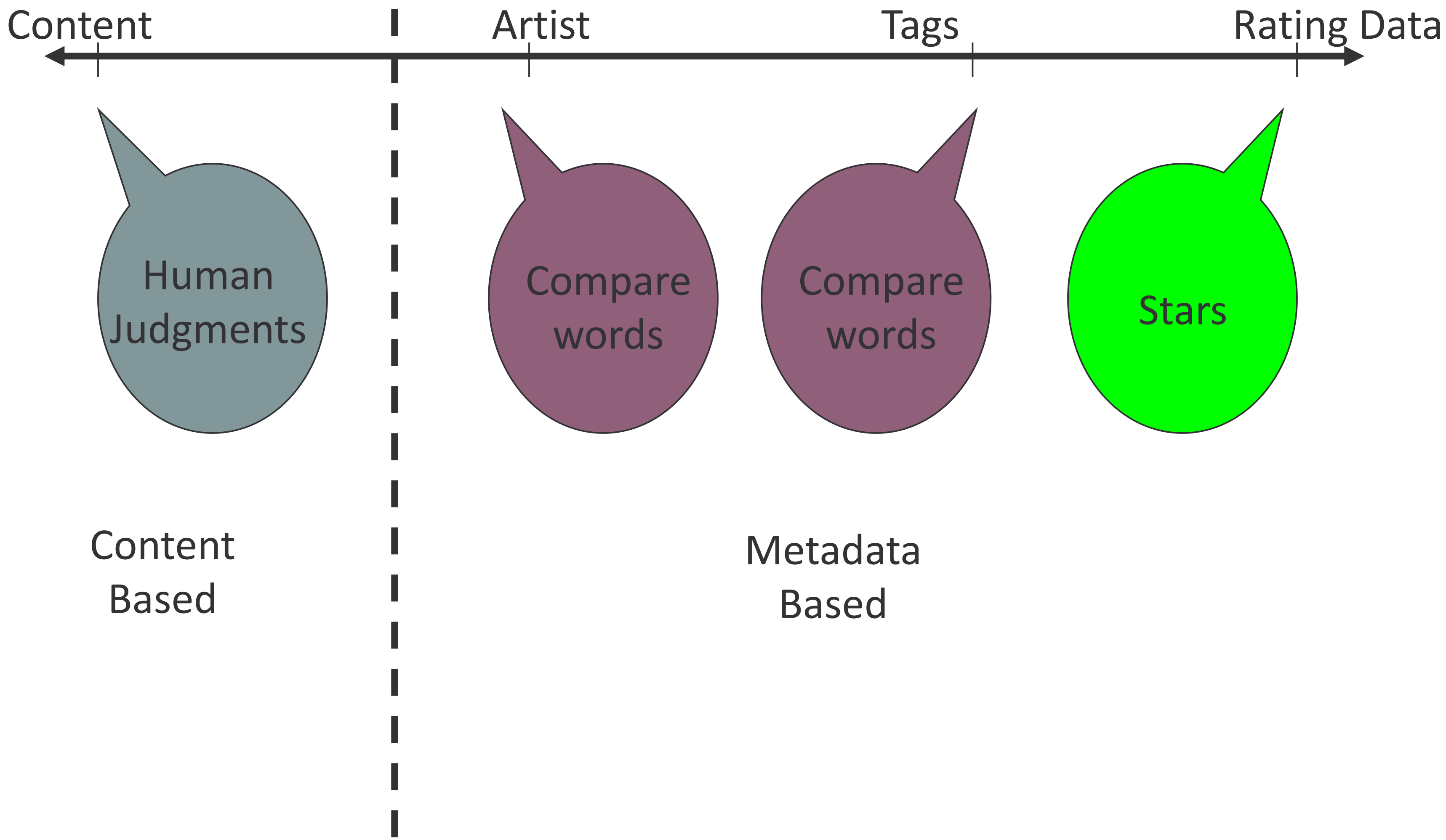




Semi-Supervised Learning



Metadata Spectrum



A Small Experiment

- **380,911 Subjects**
- **1000 Jazz Songs**
- **1,449,335 Ratings**