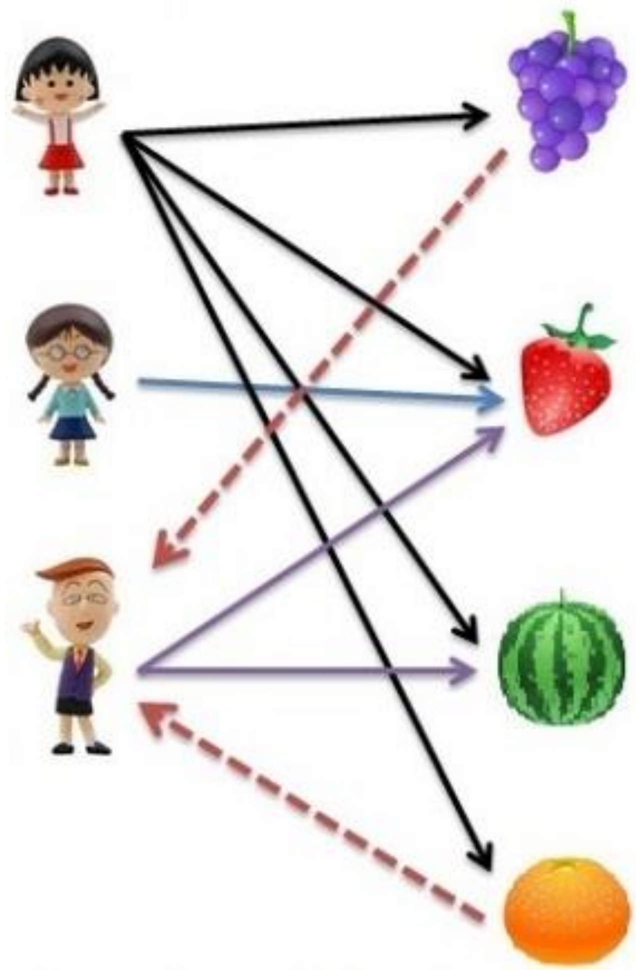
RECOMMENDATION

EXAMPLES

- Diapers and beer (legend?)
- Target pregnant teenager



diapers and beer. http://www.forbes.com/forbes/1998/0406/6107128a.html
pregnancy. teenager http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html

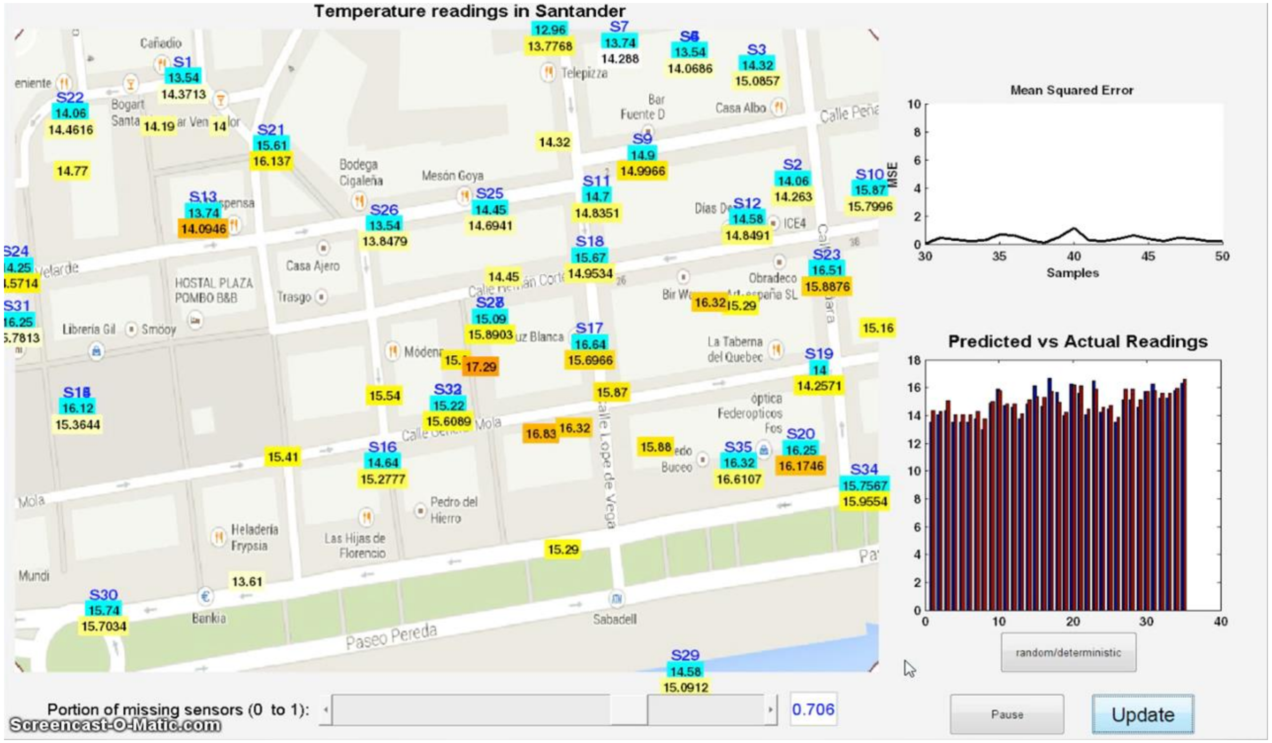
EXAMPLES

- Netflix Prize 2009

- Amazon Recommendation Engine

EXAMPLES

- Missing sensor data



COLLABORATIVE FILTERING

- 400,000 users
  17,000 movies
  But only a few ratings (1%)
- User a
  movie i
  Rating $Y_{ai} \in \mathbb{R}$ (e.g. 1-5)
- Goal is to predict unobserved ratings

m movies

n users

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

Y_{ai}

COLLABORATIVE FILTERING

- Collaborative: cross-users
  Filtering: prediction
- Matrix or tensor
  completion problems

A tensor is a multi-
dimensional array.
e.g. $n \times m$ 2-tensor
e.g. $p \times q \times r$ 3-tensor
(2-tensor = matrix)

m movies

n users

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

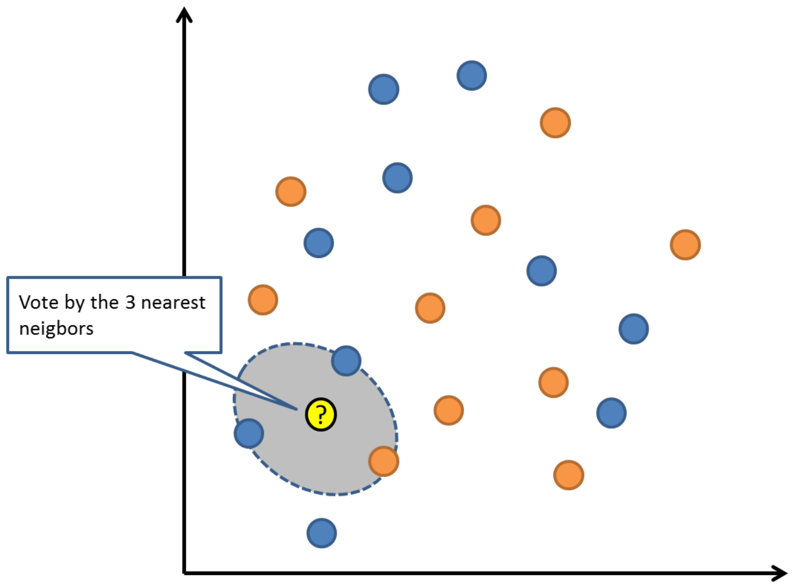
Y_{ai}

NEAREST NEIGHBOR

K-NEAREST NEIGHBORS

Basic Idea.

- Find a few users $b_1, \ldots, b_k$ (neighbors) that are similar to user a
- Use information from users $b_1, \ldots, b_k$ to predict ratings of user a



DISTANCE METRIC

Measure statistical correlation between users a and b

$$\text{sim}(a,b) = \text{corr}(a,b) = \frac{\sum_{j \in CR(a,b)} (Y_{aj} - \tilde{Y}_a)(Y_{bj} - \tilde{Y}_b)}{\sqrt{\sum_{j \in CR(a,b)} (Y_{aj} - \tilde{Y}_a)^2} \sqrt{\sum_{j \in CR(a,b)} (Y_{bj} - \tilde{Y}_b)^2}} \in [-1, 1]$$

cosine similarity!

where

1. $\text{CR}(a,b)$ is the set of movies rated by both a and b (common ratings)
2. $\tilde{Y}_a = \frac{1}{|\text{CR}(a,b)|} \sum_{j \in \text{CR}(a,b)} Y_{aj}$
   (average rating for user a among common ratings)

PREDICTION

$$\hat{Y}_{ai} = \bar{Y}_a + \frac{\sum_{b \in kNN(a,i)} \mathrm{sim}(a,b)(Y_{bi} - \bar{Y}_b)}{\sum_{b \in kNN(a,i)} |\mathrm{sim}(a,b)|}$$

where

1. $\bar{Y}_a$ is the mean rating for user a
2. $\mathrm{kNN}(a,i)$ are the k nearest neighbors of user a that also rated movie i

DISCUSSION

$$\hat{Y}_{ai} = \bar{Y}_a + \frac{\sum_{b \in kNN(a,i)} \text{sim}(a,b)(Y_{bi} - \bar{Y}_b)}{\sum_{b \in kNN(a,i)} |\text{sim}(a,b)|}$$

- Weighted sum (could be negative weights).
  Should we consider anti-correlated neighbors?
- Good to account for bias (mean) of each user,
  but sensitive to the spread (variance) of each user.
- How do we choose the optimal k?
  - Minimize validation error
  - Problem statement
  - Computational resources

MATRIX FACTORIZATION

LOW-RANK APPROXIMATIONS

Idea.

Assume fully observed $\hat{Y}$ lie in a small class $\mathcal{H}$ of matrices.

Find matrix in $\mathcal{H}$ that is closest to partially observed Y.

Low-rank matrices.

$$\hat{Y} = UV^\top, \quad \text{where } U \in \mathbb{R}^{n\times d},\ V \in \mathbb{R}^{m\times d},\ d \ll \min(m,n)$$

WHY LOW-RANK?

Suppose there are d pure types of users,
with rating preferences

$$V_{*1}, V_{*2}, \ldots, V_{*d} \in \mathbb{R}^m.$$

Assume every user's rating can be expressed
as a weighted sum of these pure ratings:

$$Y_{a*} = U_{a1} V_{*1} + U_{a2} V_{*2} + \ldots + U_{ad} V_{*d}.$$

Then, $Y = UV^\top$ where $V_{*1}, V_{*2}, \ldots, V_{*d}$ are the columns of V.

Question. What happens when $d = \min(m, n)$?

ALGORITHM

Coordinate Descent.

$$\mathcal{L}(U,V) = \sum_{(a,i)\in D} \frac{1}{2}(Y_{ai} - (UV^\top)_{ai})^2 + \frac{\lambda}{2}\|U\|^2 + \frac{\lambda}{2}\|V\|^2$$

$$= \sum_{(a,i)\in D} \frac{1}{2}\left(Y_{ai} - u^{(a)} \cdot v^{(i)}\right)^2 + \frac{\lambda}{2}\sum_a \left\|u^{(a)}\right\|^2 + \frac{\lambda}{2}\sum_i \left\|v^{(i)}\right\|^2$$

where $u^{(a)}$ is the a-th row of U and $v^{(i)}$ is the i-th row of V.

Repeat until convergence:

1. Fix V and minimize $\mathcal{L}(U,V)$ over U.
2. Fix U and minimize $\mathcal{L}(U,V)$ over V.

ALGORITHM

1. Randomly initialize $v^{(1)}, v^{(2)}, \ldots, v^{(m)}$.
2. Repeat until convergence:
   a. For each user a, find $u^{(a)}$ that minimizes
   $$\sum_{i:\,(a,i)\in D} \frac{1}{2}\left(Y_{ai} - u^{(a)} \cdot v^{(i)}\right)^2 + \frac{\lambda}{2}\left\|u^{(a)}\right\|^2$$
   b. For each movie i, find $v^{(i)}$ that minimizes
   $$\sum_{a:\,(a,i)\in D} \frac{1}{2}\left(Y_{ai} - u^{(a)} \cdot v^{(i)}\right)^2 + \frac{\lambda}{2}\left\|v^{(i)}\right\|^2$$

These are standard linear regression problems.

DISCUSSION

Optimization.

1. Like k-means, the algorithm converges to a local minimum.
2. Perform multiple initializations, and pick best result.

Generalization.

1. Use validation to pick right hyperparameters d and λ.
   a. Split data set into Training Data and Validation Data.
   b. For each d and λ, train a predictor using Training Data.
   c. Choose predictor that minimizes Validation Error.