Recommender Systems

Jia-Bin Huang
Virginia Tech

ECE-5424G / CS-5824

Spring 2019

Administrative

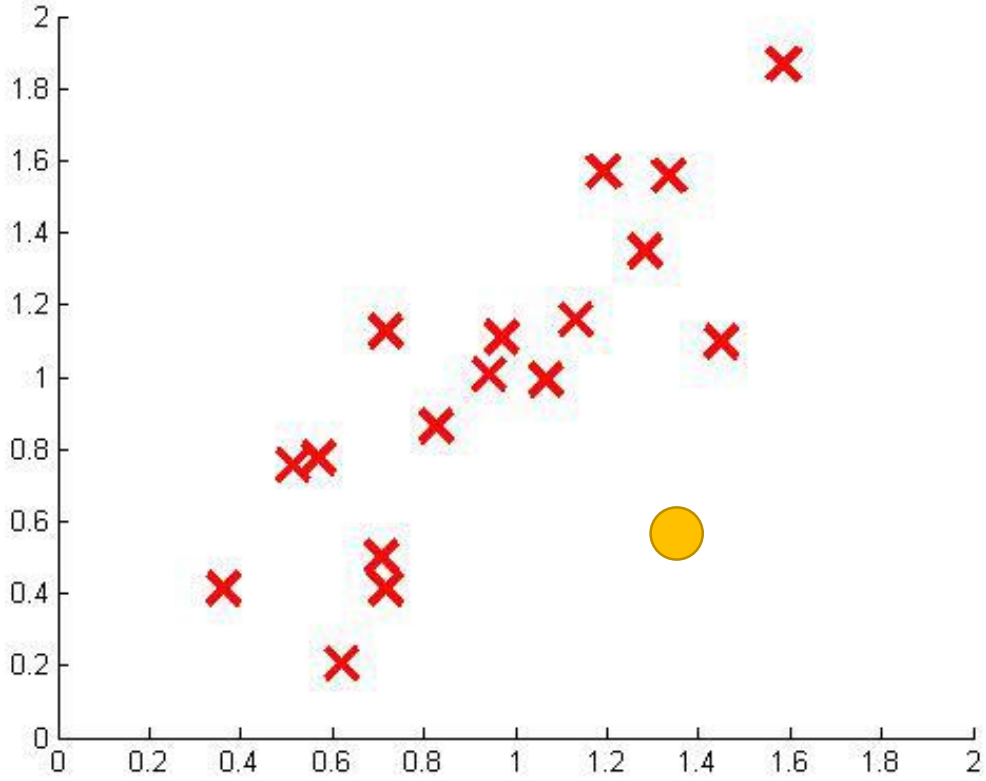
- HW 4 due April 10

Unsupervised Learning

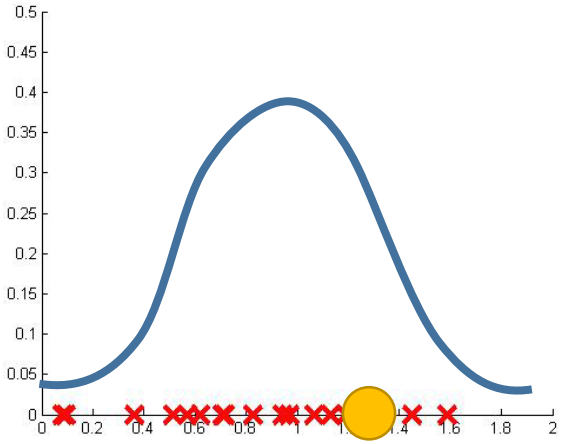
- Clustering, K-Mean
- Expectation maximization
- Dimensionality reduction
- Anomaly detection
- Recommendation system

Motivating example: Monitoring machines in a data center

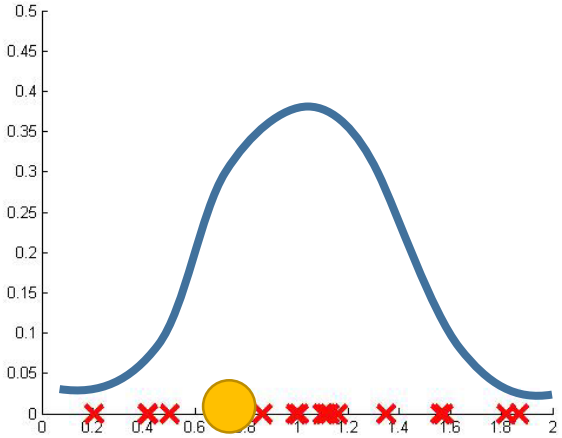
x_2 (Memory use)



x_1 (CPU load)



x_1 (CPU load)



x_2 (Memory use)

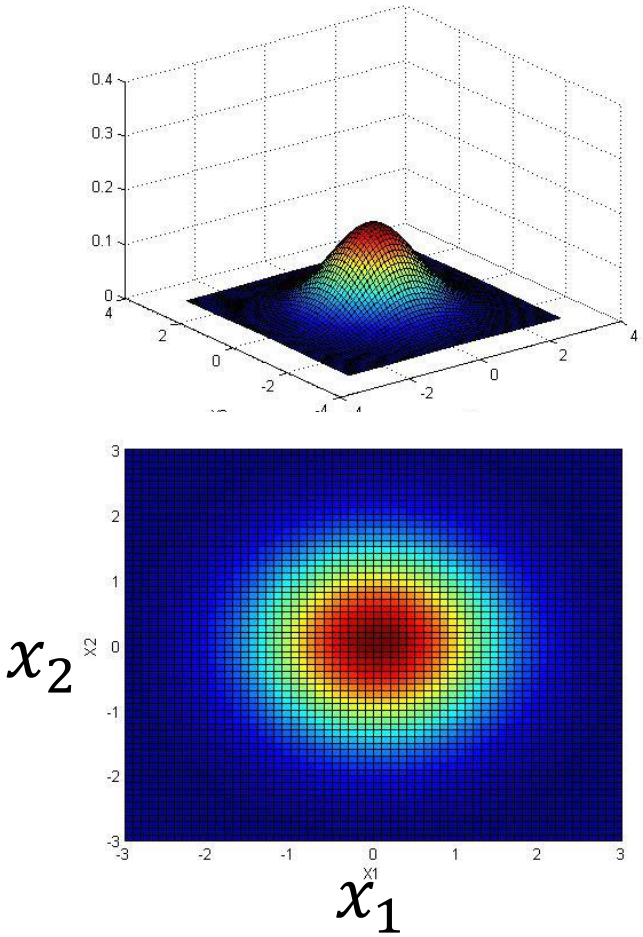
Multivariate Gaussian (normal) distribution

- $x \in R^n$. Don't model $p(x_1), p(x_2), \cdots$ separately
- Model $p(x)$ all in one go.
- Parameters: $\mu \in R^n, \Sigma \in R^{n \times n}$ (covariance matrix)

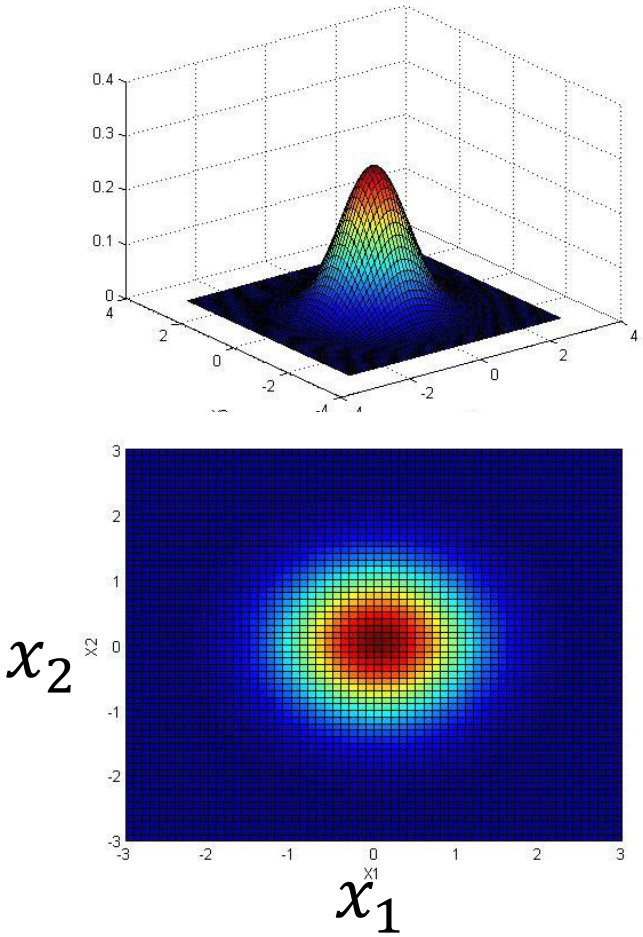
- $p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-(x-\mu)^\top \Sigma^{-1}(x-\mu))$

Multivariate Gaussian (normal) examples

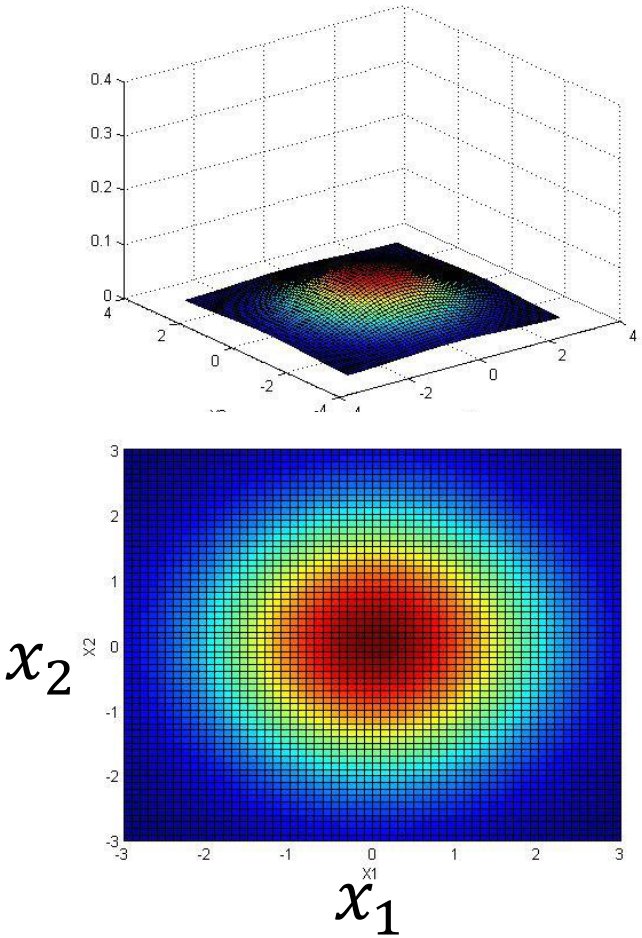
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$$

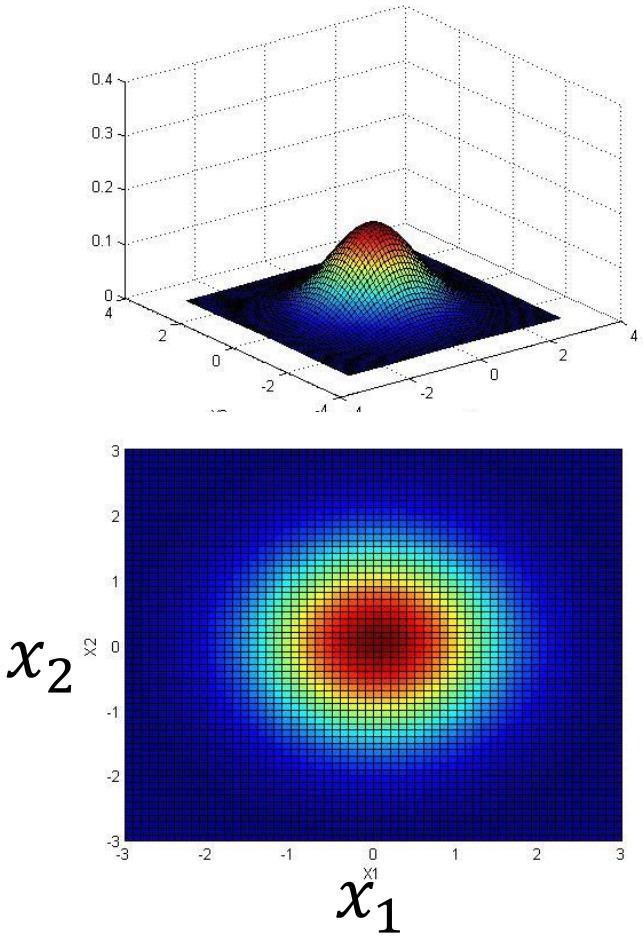


$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

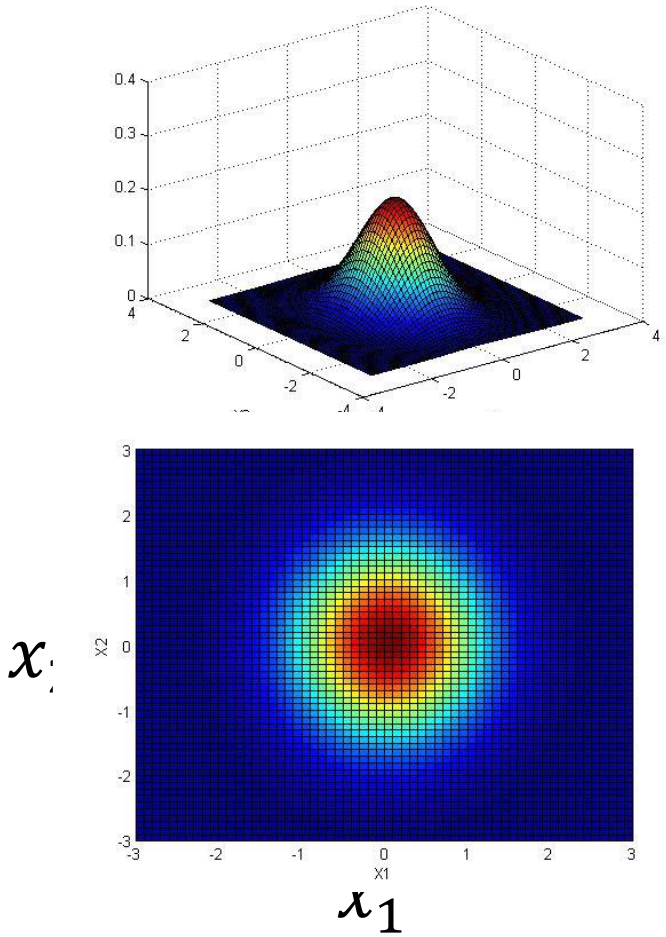


Multivariate Gaussian (normal) examples

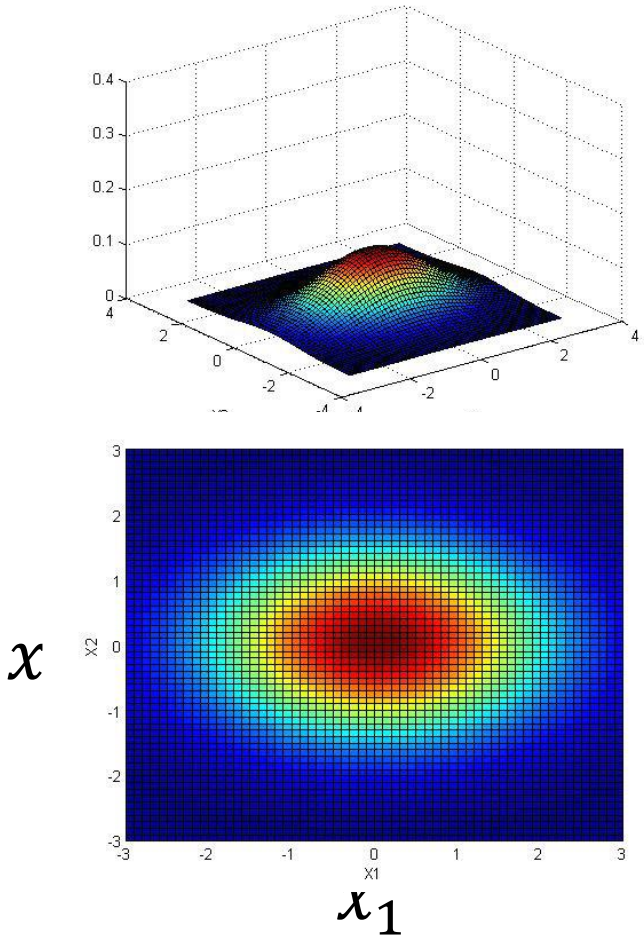
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$$

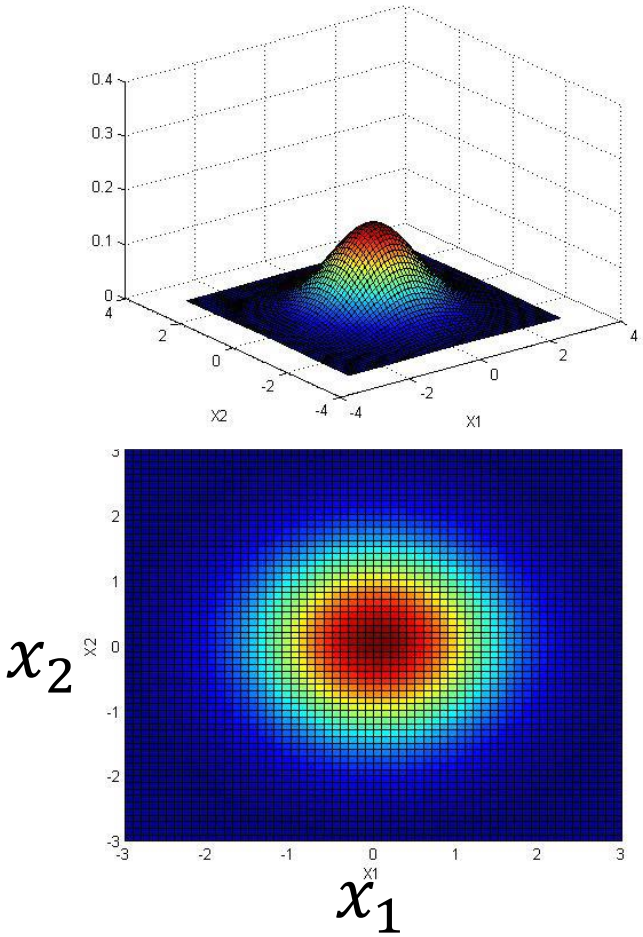


$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

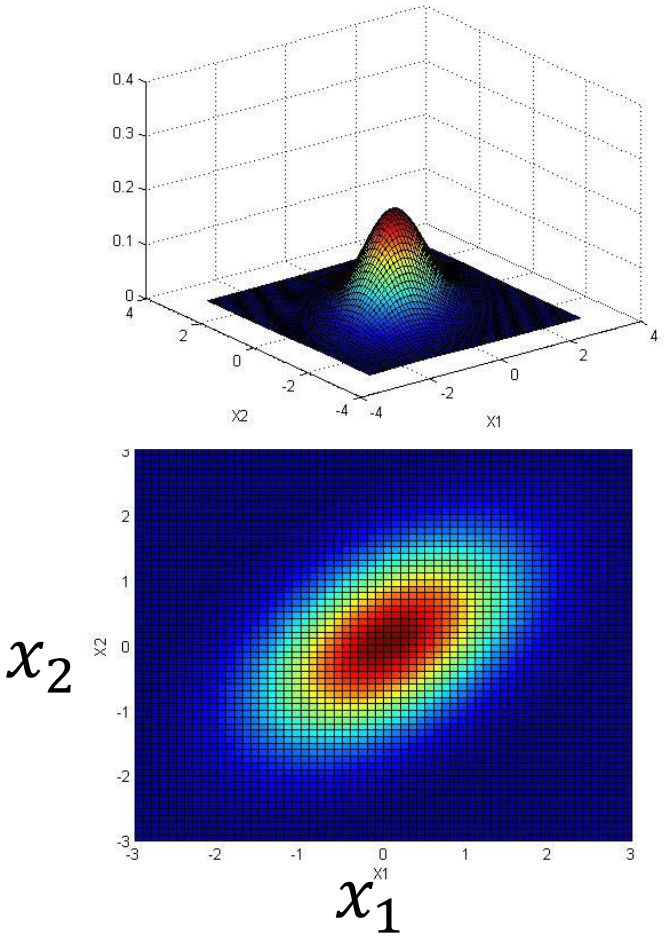


Multivariate Gaussian (normal) examples

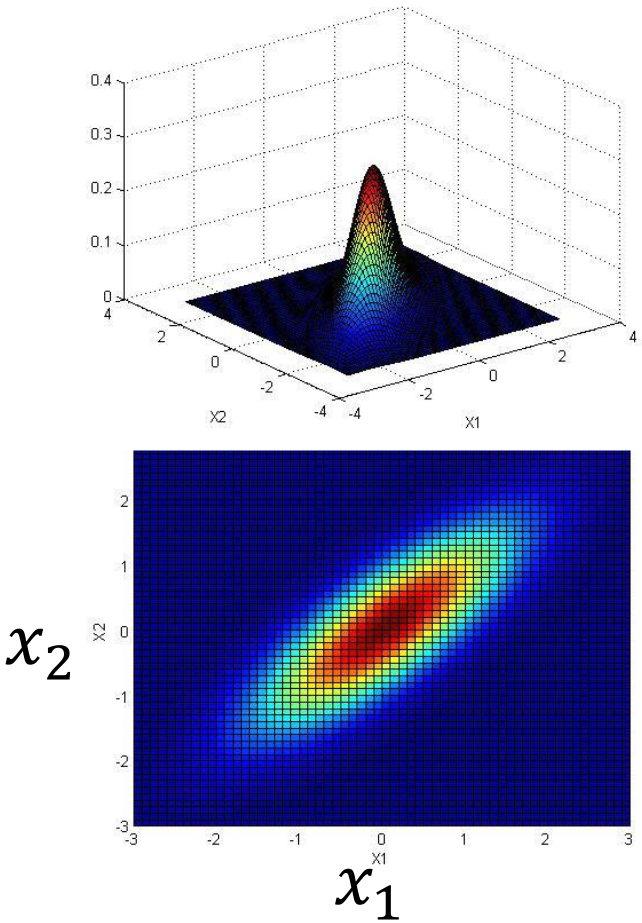
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



Anomaly detection using the multivariate Gaussian distribution

1. Fit model $p(x)$ by setting

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^\top$$

2 Give a new example x, compute

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp(-(x-\mu)^\top\Sigma^{-1}(x-\mu))$$

Flag an anomaly if $p(x) < \epsilon$

Original model

$$p\left(x_1; \mu_1, \sigma_1^2\right) p\left(x_2; \mu_2, \sigma_2^2\right) \cdots p\left(x_n; \mu_n, \sigma_n^2\right)$$

Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values

Computationally cheaper (alternatively, scales better)

OK even if training set size is small

Original model

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-(x-\mu)^\top \Sigma^{-1}(x$$

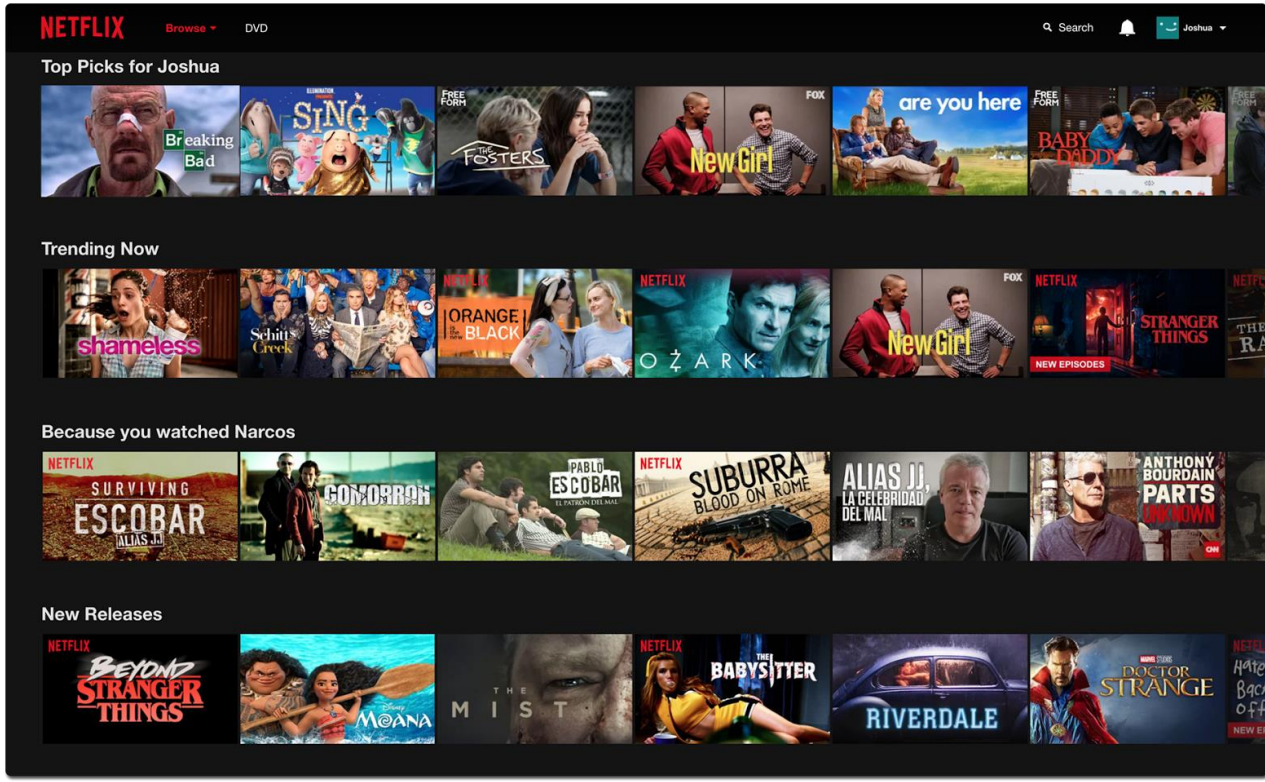
Recommender Systems

- Motivation
- Problem formulation
- Content-based recommendations
- Collaborative filtering
- Mean normalization

Recommender Systems

- **Motivation**
- Problem formulation
- Content-based recommendations
- Collaborative filtering
- Mean normalization

You may also like..?



2009
NETFLIX
DATE: 09·21·09
PAY TO THE ORDER OF: BellKor's Pragmatic Chaos
$ 1,000,000 00
AMOUNT: ONE MILLION
00/100
FOR: The Netflix Prize
Reed Hastings

Recommender Systems

- Motivation
- **Problem formulation**
- Content-based recommendations
- Collaborative filtering
- Mean normalization

Example: Predicting movie ratings

- User rates movies using zero to five stars

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

- n_u = no. users
- n_m = no. movies
- $r(i,j) = 1$ if user j has rated movie i
- $y^{(i,j)}$ = rating given by user j to movie i

Recommender Systems

- Motivation
- Problem formulation
- **Content-based recommendations**
- Collaborative filtering
- Mean normalization

Content-based recommender systems

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

For each user j, learn a parameter $\theta^{(j)} \in R^3$.

Predict user j as rating movie i with $(\theta^{(j)})^\top x^{(i)}$ stars.

Content-based recommender systems

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

$$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$$

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$$

$$(\theta^{(1)})^\top x^{(3)} = 5 * 0.99 = 4.95$$

For each user j, learn a parameter $\theta^{(j)} \in R^3$.

Predict user j as rating movie i with $(\theta^{(j)})^\top x^{(i)}$ stars.

Problem formulation

- $r(i,j) = 1$ if user j has rated movie i
- $y^{(i,j)}$ = rating given by user j to movie i
- $\theta^{(j)}$ = parameter vector for user j
- $x^{(i)}$ = feature vector for user i
- For each user j, predicted rating: $(\theta^{(j)})^\top x^{(i)}$

- $m^{(j)}$ = no. of movies rated by user j

Goal: learn $\theta^{(j)}$:

$$\min_{\theta^{(j)}} \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^\top \left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2$$

Optimization objective

- Learn $\theta^{(j)}$ (parameter for user j):

$$\min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^\top (x^{(i)}) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(\theta_k^{(j)} \right)^2$$

Learn $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$:

$$\min_{\theta^{(1)},\theta^{(2)},\cdots,\theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^\top (x^{(i)}) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)} \right)^2$$

Optimization algorithm

$$\min_{\theta^{(j)}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)^2+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

Gradient descent update:

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)x_k^{(i)} \qquad \text{(for } k=0\text{)}$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)x_k^{(i)}+\lambda\,\theta_k^{(j)}\right) \qquad \text{(for } k\neq 0\text{)}$$

Recommender Systems

- Motivation
- Problem formulation
- Content-based recommendations
- **Collaborative filtering**
- Mean normalization

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \qquad \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \qquad \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \qquad \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \qquad x^{(1)} = \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix}$$

Optimization algorithm

- Given $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$, to learn $x^{(i)}$:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^\top \left(x^{(i)} \right) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2$$

- Given $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$, to learn $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$:

$$\min_{x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^\top \left(x^{(i)} \right) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2$$

Collaborative filtering

- Given $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$ (and movie ratings),

  Can estimate $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$

- Given $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$

  Can estimate $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$

Collaborative filtering optimization objective

- Given $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$, estimate $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$

$$\min_{\theta^{(1)},\theta^{(2)},\cdots,\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

- Given $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$, estimate $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$

$$\min_{x^{(1)},x^{(2)},\cdots,x^{(n_m)}} \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

Collaborative filtering optimization objective

- Given $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$, estimate $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$

$$\min_{\theta^{(1)},\theta^{(2)},\cdots,\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u}\boxed{\sum_{i:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)^2}+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

- Given $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$, estimate $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$

$$\min_{x^{(1)},x^{(2)},\cdots,x^{(n_m)}} \frac{1}{2}\sum_{i=1}^{n_m}\boxed{\sum_{j:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)^2}+\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

- Minimize $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$ and $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$ simultaneously

$$J=\frac{1}{2}\sum_{j:r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right)-y^{(i,j)}\right)^2+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2+\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

Collaborative filtering optimization objective

$$J(x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}) =$$

$$\frac{1}{2}\sum_{r(i,j)=1}\left((\theta^{(j)})^\top\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

Collaborative filtering algorithm

- Initialize $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}$ to small random values
- Minimize $J(x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization algorithm). For every $j = 1 \cdots n_u, i = 1, \cdots, n_m$:

$$x_k^{(j)} := x_k^{(j)} - \alpha \left(\sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^\top \left(x^{(i)}\right) - y^{(i,j)} \right) \theta_k^{(i)} + \lambda\, x_k^{(i)} \right)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^\top \left(x^{(i)}\right) - y^{(i,j)} \right) x_k^{(i)} + \lambda\, \theta_k^{(j)} \right)$$

- For a user with parameter θ and movie with (learned) feature x, predict a star rating of $\theta^\top x$

Collaborative filtering

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Collaborative filtering

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

- Predicted ratings:

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

$$X = \begin{bmatrix} - & \left(x^{(1)}\right)^\top & - \\ - & \left(x^{(2)}\right)^\top & - \\ & \vdots & \\ - & \left(x^{(n_m)}\right)^\top & - \end{bmatrix} \qquad \Theta = \begin{bmatrix} - & \left(\theta^{(1)}\right)^\top & - \\ - & \left(\theta^{(2)}\right)^\top & - \\ & \vdots & \\ - & \left(\theta^{(n_u)}\right)^\top & - \end{bmatrix}$$

$$\mathrm{Y} = \mathrm{X}\Theta^\top$$

Low-rank matrix factorization

Finding related movies/products

- For each product i, we learn a feature vector $x^{(i)} \in R^n$

x_1: romance, x_2: action, x_3: comedy, …

- How to find movie j relate to movie i?

Small $\left\|x^{(i)} - x^{(j)}\right\|$ movie j and I are "similar"

Recommender Systems

- Motivation
- Problem formulation
- Content-based recommendations
- Collaborative filtering
- **Mean normalization**

Users who have not rated any movies

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	Eve (5)
Love at last	5	5	0	0	?
Romance forever	5	?	?	0	?
Cute puppies of love	?	4	0	?	?
Nonstop car chases	0	0	5	4	?
Swords vs. karate	0	0	5	?	?

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

$$\frac{1}{2}\sum_{r(i,j)=1}\left((\theta^{(j)})^{\top}\left(x^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

$$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Users who have not rated any movies

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	Eve (5)
Love at last	5	5	0	0	0
Romance forever	5	?	?	0	0
Cute puppies of love	?	4	0	?	0
Nonstop car chases	0	0	5	4	0
Swords vs. karate	0	0	5	?	0

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

$$\frac{1}{2}\sum_{r(i,j)=1}\left((\theta^{(j)})^{\top}(x^{(i)}) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

$$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Mean normalization

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \quad Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

Learn $\theta^{(j)}, x^{(i)}$

For user j, on movie i predict:

$$\left(\theta^{(j)}\right)^{\top}\left(x^{(i)}\right) + \mu_i$$

User 5 (Eve):

$$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \left(\theta^{(5)}\right)^{\top}\left(x^{(i)}\right) + \mu_i$$

Recommender Systems

- Motivation
- Problem formulation
- Content-based recommendations
- Collaborative filtering
- Mean normalization