# WEEK - 9

# ANAMOLY DETECTION

Example:

Aircraft engine features:

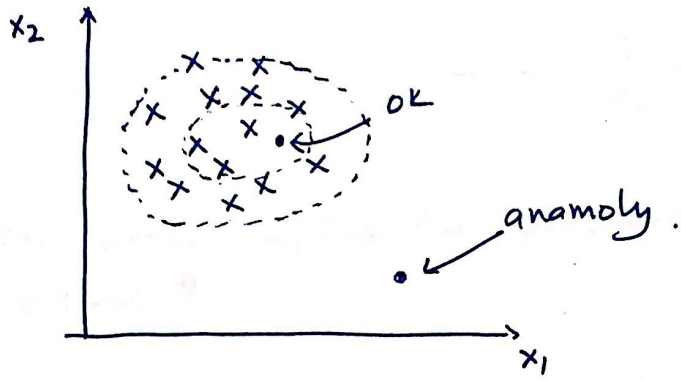
$x_1$ = heat generated

$x_2$ = vibration intensity

Dataset: $\{x^1, x^2 \ldots x^m\}$

New engine: $x_{test}$

is $x_{test}$ is anamalous?




Build a Model $p(x)$.

$p(x_{test}) < \epsilon \rightarrow$ flag anamoly

$p(x_{test}) \geq \epsilon \rightarrow$ ok

# Fraud detection example:

→ $x^i$ = features of user $i$'s activities

→ Model $p(x)$

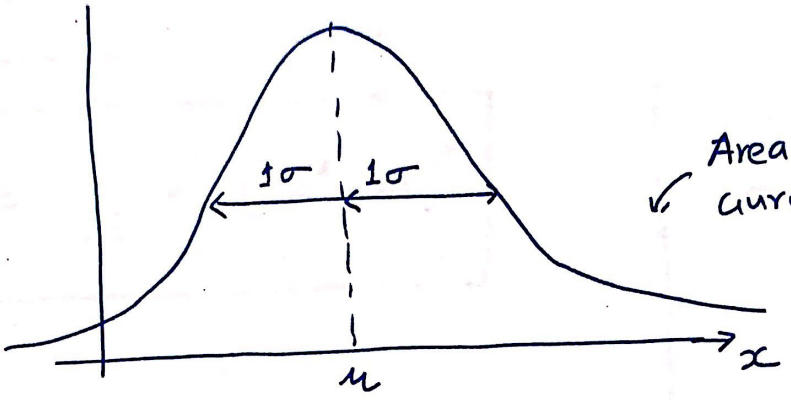
→ Identify unusual users by checking which have $p(x) < \epsilon$

- **Guassian distribution** (Normal)

$$X \sim N(\mu, \sigma^2) \quad : \quad X \in \mathbb{R}$$

$\sigma^2$ = variance

$\sigma$ = S.D




Area under the curve = 1

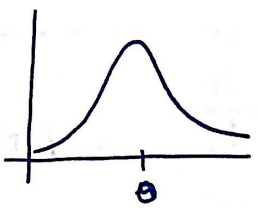
The curve defines the probability of $x$ taking different values.

$$p(x) \text{ or } p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

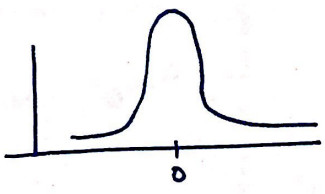
$$\Downarrow$$

$$* \text{ simil}(x, \ell) = \exp\left(-\frac{\|x-\ell\|^2}{2\cdot\sigma^2}\right)$$

① $\mu = 0$, $\sigma = 1$




② $\mu = 0$, $\sigma = 0.5$



- **Parameter Estimation :-**

Data set : $\{ x^1, x^2, \dots x^m \}$ $\quad x^i \in R$

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

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x^i - \mu)^2$$

∴ in statistics.

$$\sigma = \sqrt{\frac{\Sigma (x-\mu)^2}{n-1}}$$

↙ sample

As in machine learning $n >> 25$

∴ $n-1$ is not used

- **Density Estimation :**

$$p(x) = \underbrace{p(x_1; \mu_1, \sigma_1^2)\ p(x_2; \mu_2, \sigma_2^2) \dots p(x_n; \sigma_n^2, \mu_n)}_{\forall\ n \text{ features.}}$$

∴ ✱ $$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2)$$

## Anomaly detection algorithm

① choose features $x_i$ that you think might be indicative of anomalous examples.

② Fit parameters: $\mu_1, \dots \mu_n \mid \sigma_1^2, \dots \sigma_n^2$

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^i$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^i - \mu_j)^2$$

③ Given new example $x$, compute $p(x)$:

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2)$$

$$= \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi} \cdot \sigma_j} \cdot \exp\left( - \frac{(x_j - \mu_j)^2}{2 \cdot \sigma_j^2} \right)$$

④ Anamoly if $p(x) < \epsilon$

## • Developing & Evaluation of Anomaly detection System

→ Developing a learning algorithm, making decisions is much easier if we have a way of evaluating our learning algorithm.

→ Assume we have some labeled data, of anomalous & non-anomalous examples ($y \in \{0, 1 \ldots\}$)

→ Training set: $\{x^1, x^2 \ldots x^m\}$ : $x_{train}$

→ $\begin{cases} x_{test}, x_{cv} \\ y_{test}, y_{cv} \end{cases}$ ~~and $x_{train}$~~

example: 10000 good engines
20 anomalous.

| | CV | test |
|---|---|---|
| 6000 : Training set ($y=0$) | 2000 good engines<br>10 anomalous ($y=1$) | 2000 good ($y=0$)<br>10 anomalous ($y=1$) |

- **Algorithm Evaluation :**

→ Fit model $p(x)$ on trainging set $\{x^1, x^2 \ldots x^m\}$

→ on a cv/test $x$, predict

$$y = \begin{cases} 1 & \text{if } p(x) < \epsilon \quad \text{(anomaly)} \\ 0 & \text{if } p(x) \geq \epsilon \quad \text{(normal)} \end{cases}$$

→ Possible evaluation metrics

① 

| | 1 | 0 |
|---|---|---|
| 1 | True positive | False positive |
| 0 | False negative | True negative |

② precision / Recall

③ F1 - score

$$\text{precision} = \frac{\text{True positive}}{\text{True pos + False posi'}}$$

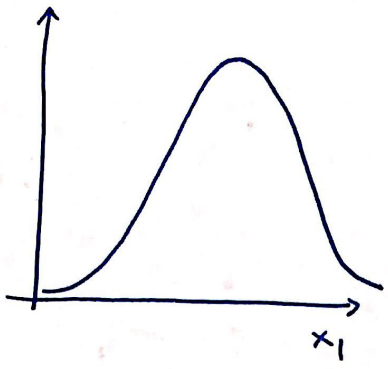
$$\text{Recall} = \frac{\text{True pos}}{\text{False neg + True pos.}}$$

$$\text{Accoracy} = \frac{\text{True pos + True neg}}{\text{All examples}}$$

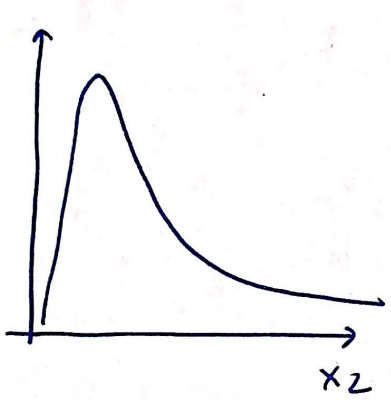
## Anamoly Detections vs. Supervised learning

| Anamoly Detections | Supervised learning |
| --- | --- |
| ① Very small number of positives examples ($y=1$) | Large number of positive & negative examples |
| ② Large number of negative examples ($y=0$) | |
| * Many different "type" of anomalies Hard for any algorithm to learn from positive examples what the anomalies look like; | * Enough positive examples for algorithm to get a sense of what positive examples are like, future positive examples likely to be similar to ones in training set. |
| •• Future anomalies may look nothing like any of the anomalous examples we've seen so far. | |
| Examples: | |
| - fraud detection | - Email classification |
| - Manufacturing (eg. aircraft) | - weather prediction |
| - Monitoring machines in a data center | - Cancer classification |

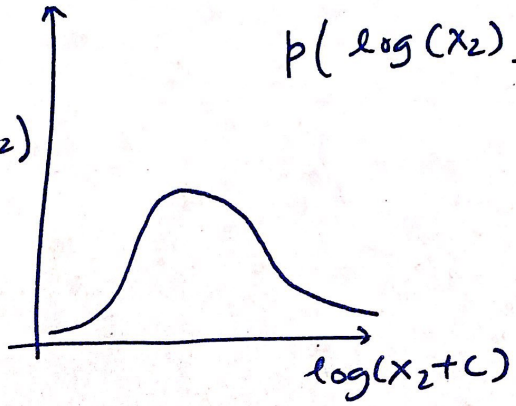
- **choosing what features to use** :-




$p(x_1; \mu, \sigma^2)$




$\log(x_2)$




$p(\log(x_2); \mu, \sigma^2)$

Octave

```
hist(x.^05, 50)
hist(log(x), 50)
```

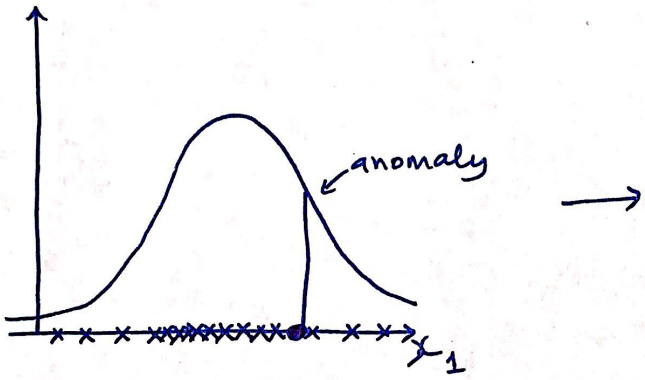
- **Error analysis for anomaly detection:**

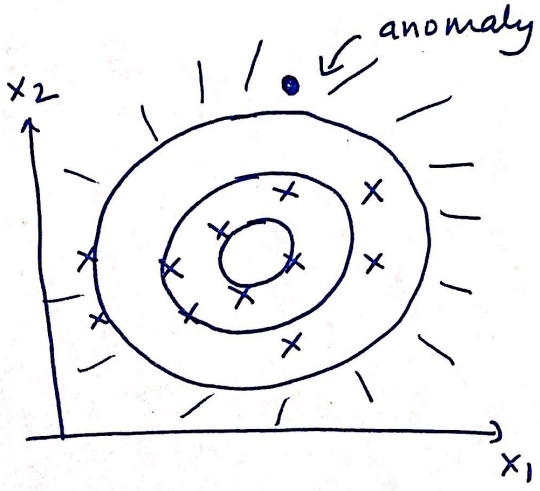
Want $p(x) \geq \epsilon$ : large for normal examples, $x$

$p(x) \leq \epsilon$ : small for anomalous examples, $x$

Most common problem:

$p(x)$ is comparable for normal & anomalous examples.




plot against a new feature, $x_2$ (or try/create a new feature)

Example: Monitoring computers in a data center

$x_1$ = memory use of a computer

$x_2$ = number of disk accesses/sec

$x_3$ = CPU load

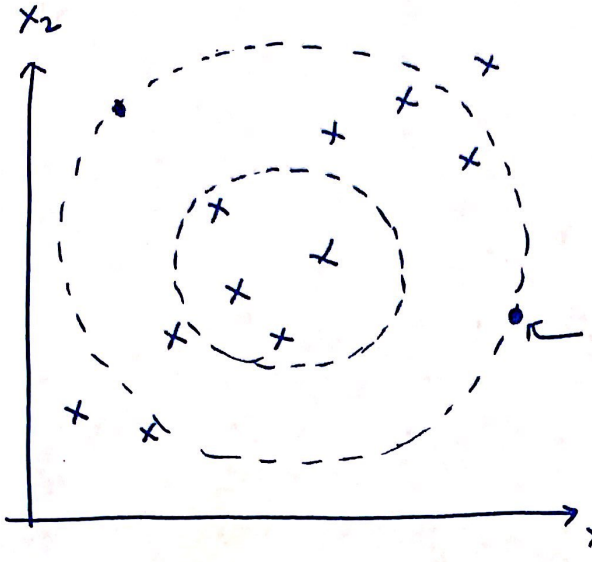
$x_4$ = network traffic.

- choose features that might take on unusually large or small values in the event of an anomaly.

$$x_5 = \frac{\text{CPU load}}{\text{N/W traffic}}$$

$$x_6 = \frac{(\text{CPU load})^2}{\text{N/W traffic}}$$

# Multi-variate Gaussian Distribution
## Anomaly Detection




$x_1 \sim N(\mu_1, \sigma_1^2)$

$x_2 \sim N(\mu_2, \sigma_2^2)$

points on this curve have same probability, which might not be actual case.

$\therefore$ $x \in \mathbb{R}^n$. Don't model $p(x_1), p(x_2)$ ... etc. seperately

Model $p(x)$ all in one go

Parameters: $\mu \in \mathbb{R}^n$

$\sum \in \mathbb{R}^{n \times n}$ : Covariance matrix

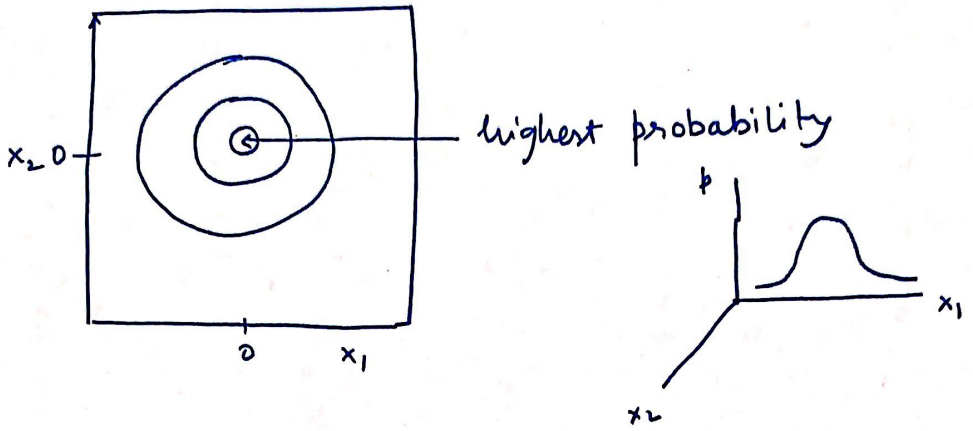
$\therefore$ $p(x) \Rightarrow p(x; \mu; \Sigma)$

$$= \frac{1}{(\sqrt{2\pi})^n |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \cdot \Sigma^{-1}(x-\mu)\right)$$

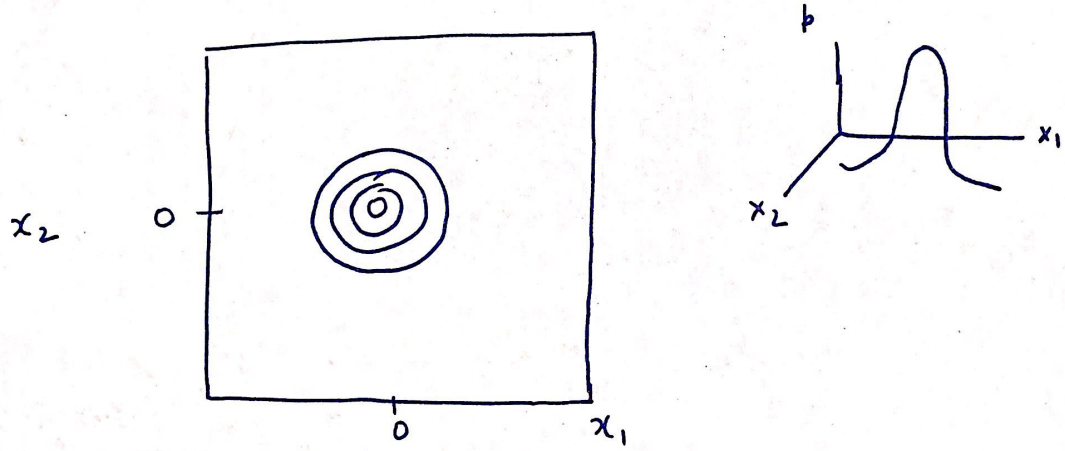
$|\Sigma|$ = determinat

Example:

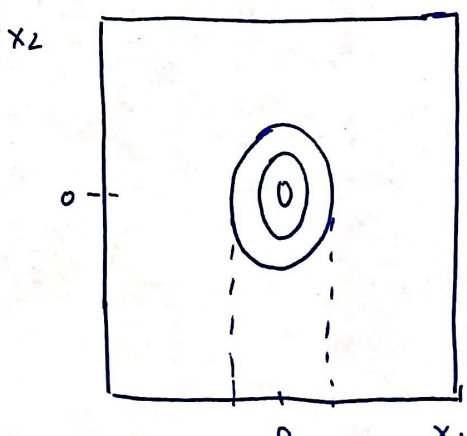
① $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ , $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$




② $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ , $\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$




③ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ , $\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$

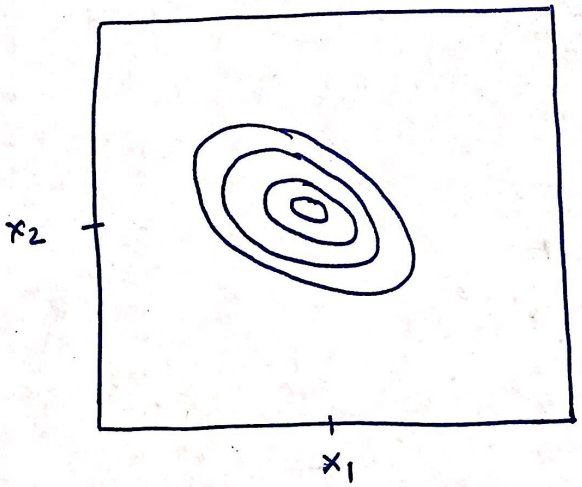


④ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & +0.5 \\ 0.5 & 1 \end{bmatrix}$

high/Medium/ low (+)vely corelation


⑤ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$

-vely corelated



# $p(x; \mu; \Sigma^2) = \frac{1}{(2\pi)^{n/2} \cdot |\Sigma|^{1/2}} \cdot \exp\left(-(x-\mu)^T \cdot \Sigma^{-1} (x-\mu)\right)$

parameters : $\mu, \Sigma$

# Parameter fitting:

(1) Given training set : $\{x^1, x^2 \ldots x^m\}$

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

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

Fix model $p(x)$

(2) Given a new example $x$, compute

$$p(x) = \frac{1}{(2\pi)^{n/2} \cdot |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \cdot \Sigma^{-1} (x-\mu)\right)$$

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

- **Relationship to Original Model**

Original model:

$$p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2; \sigma_2^2) \times \ldots p(x_n, \sigma_n^2, \mu_n)$$

corresponds to Multi-variate Gaussian model:

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

s.t.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \sigma_3^2 & \\ & & & \ddots \\ 0 & & & \sigma_n^2 \end{bmatrix}$$

This implies that ~~we~~ we aren't allowed to model co-relation between features.

- Original Model vs. Multi-variate

| Original Model | Multi-variate |
|---|---|
| ➡ Manually create features to capture anomalies where $x_1, x_2$ take unusual combination of values. | ➡ Automatically captures co-relations b/w features |
| ➡ Computationally cheaper (alternatively, scales better to large, n) | ➡ Computationally more expensive |
| ➡ OK even if m is small | ➡ Must-have $m > n$ or else $\Sigma$ is non-invertible ($m \geq 10n$) |

# Recommender Systems

Example: Predicting movie ratings

| Movie | Alice(1) | Bob(2) | Carol(3) | Dave(4) |
|---|---|---|---|---|
| Movie 1 | 5 | 5 | 0 | 0 |
| Movie 2 | 5 | ? | ? | 0 |
| Movie 3 | ? | 4 | 0 | ? |
| Movie 4 | 0 | 0 | 5 | 4 |
| Movie 5 | 0 | 0 | 5 | ? |

$n_u$ = no. of users.

$n_m$ = no. of movies

$r(i,j) = 1$ if user $j$ has rated movie $i$

$y(i,j)$ = rating given by user $j$ to movie $i$ (defined only if $r(i,j) = 1$)

Problem: develop a recommender system to come up with a learning algorithm that can automatically fill in the missing values And then try to predict what else might be interesting to a user.

## Approach 1: Content-based recommender systems.

| Movie | $x_1$ | $x_2$ |
|---|---|---|
| 1 | 0.9 | 0 |
| 2 | 1 | 0.01 |
| 3 | 0.99 | 0 |
| 4 | 0.1 | 1.0 |
| 5 | 0 | 0.9 |

Feature vector

$x_1$: Degree to which the movie is romantic

$x_2$: Degree to which the movie is Action

$x^{(i)}$: feature vector for movie $i$ $\Rightarrow x^1 = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$ ($x_0 = 1$)

$n$ = number of features (not counting $x_0$)

$\Rightarrow n = 2$

- for each user $j$, learn a parameter $\theta^{(j)} \in \mathbb{R}^{n+1}$
- Predict user $j$ as rating moving $i$ with $(\theta^{(j)})^T \cdot x^{(i)}$ stars.

$x^3$ = feature vector for movie 3.

$$= \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$$

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

{somehow got parameter $\theta^j$ for user j ; explored later}

$\therefore \quad (\theta^1)^T x^3 = 4.95$

---

Problem formulation
(sort of linear regression)

---

$r(i,j) = 1$ : if user j has rated movie i

$y^{(i,j)}$ = rating by user j for movie i (if defined)

$\theta^j$ = parameter vector for user j

$x^i$ = ~~parameter~~ feature vector for movie i

$m^j$ = no. of movies rated by user j

For user j predict rating for movie i : $(\theta^j)^T \cdot x^i$

- **Optimization Objective**

⊛ To learn parameter $\theta^j$ ( parameter for user $j$ )

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

$$\theta^j \in \mathbb{R}^{n+1}$$

⊛ To learn $\theta^1, \theta^2, \theta^3 \ldots \theta^{(n_u)}$

$$\min_{\theta^1, \theta^2 \ldots \theta^{n_u}} \left[ \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^j)^T x^i - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{K=1}^{n} (\theta_K^j)^2 \right] = J(\theta^1, \theta^2 \ldots \theta^{n_u})$$

- **Approach 2**: Collaborative filtering

| Movie | $x_1$ | $x_2$ |
|---|---|---|
| 1 | ? | ? |
| 2 | ? | ? |
| 3 | ? | ? |
| 4 | ? | ? |
| 5 | ? | ? |

$\Rightarrow$ we donot have the feature vector for each movie

and, we have gotten our users to provide us the parameter vector $\theta^j$ [based on their interest]

$$\theta^1 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \quad \theta^2 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad \theta^3 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \quad \theta^4 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

so, we are trying to find out $x^i$ for user $j$

$$(\theta^j)^T (x^i) = r(i,j)$$

ex: $(\theta^1)^T x^1 \approx 5$ [Because Alice rated movie 1 to be 5)

## Optimization Algorithm :

→ Given $\theta^1 \dots \theta^{(n_u)}$, to learn $x^i$ :

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

prediction — actual

min the error

$\star$

→ Given $\theta^1, \dots \theta^{n_u}$, to learn $x^1 \dots x^{(nm)}$ ;

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

- ~~Given~~ Guess: $\theta \rightarrow x \rightarrow \theta \rightarrow x \rightarrow \ldots\ldots$

New optimization objective:

Minimize $\{x^1 \ldots x^{(nm)}\}$ and $\{\theta^1 \ldots \theta^{nu}\}$

simultaneously: $x \in \mathbb{R}^n$, $\theta \in \mathbb{R}^n$

$$\Rightarrow \left[ J(\theta^1 \ldots \theta^{nu}, x^1 \ldots x^{nm}) = \frac{1}{2} \sum_{i,j : r(i,j)=1} \left((\theta^j)^T x^i - y^{i,j}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^j)^2 + \frac{\lambda}{2} \sum_{i=1}^{nm} \sum_{k=1}^{n} (x_k^i)^2 \right]$$

$$\Rightarrow \left[ \therefore \min_{\substack{x^1 \ldots x^{nm} \\ \theta^1 \ldots \theta^{nu}}} J(x^1 \ldots x^{nm}, \theta^1 \ldots \theta^{nu}) \right]$$

## → Steps for Collaborative filtering Algorithm

① : Initialize $x, \theta \in \mathbb{R}^n$ to small random values.

② Min. $J(\ldots)$ using gradient descent (or any advanced optimization algo). $\forall\, i, j$

$$x_k^i = x_k^i - \alpha \underbrace{\left( \sum_{j : r(i,j)=1} ((\theta^j)^T x^i - y^{i,j}) \cdot \theta_k^j + \lambda\, x_k^i \right)}_{\frac{\partial J(\ldots)}{\partial x_k^i}}$$

$$\theta_k^j = \theta_k^j - \alpha \underbrace{\left( \sum_{i : r(i,j)=1} ((\theta^j)^T x^i - y^{i,j})\, x_k^i + \lambda\, \theta_k^j \right)}_{\frac{\partial J(\ldots)}{\partial \theta_k^j}}$$

③ For a user with parameter $\theta$ & movie with feature $x$, predict a star rating of $\theta^T x$

- **Vectorization** : Low rank matrix factorization
  (collaborative filtering)

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

Predicted ratings: $(\theta^j)^T x^i$

$$\begin{bmatrix} (\theta^1)^T x^1 & (\theta^2)^T x^1 & \cdots & (\theta^{n_u})^T x^1 \\ (\theta^1)^T x^2 & & & \\ \vdots & & & \\ (\theta^1)^T x^{n_m} & & & (\theta^{n_u})^T x^{n_m} \end{bmatrix} = X * \theta^T$$

$n_m \times n_u$

$$X = \begin{bmatrix} -(x^1)^T- \\ -(x^2)^T- \\ \vdots \\ -(x^{n_m})^T- \end{bmatrix} \quad ; \quad \theta = \begin{bmatrix} -(\theta^1)^T- \\ -(\theta^2)^T- \\ \vdots \\ -(\theta^{n_u})^T- \end{bmatrix}$$

$(n_m \times 1)$ $(n_u \times 1)$

## Application: Recommender System

for each product $i$ / movie, we learn a feature vector $x^i \in \mathbb{R}^{n_m}$

→ How to find movies $j$ related to movie $i$ ?

small $\|x^i - x^j\|$ : movie $j$ and $i$ are similar

→ 5 most similar movies to movie $i$

↳ Find the 5 movies $j$ with the smallest $\|x^j - x^i\|$

→→ Users who have not rated any movies !

$$J(\theta^{(1)}, \ldots, \theta^{(n_u)}, x^{(1)}, \ldots, x^{(n_m)}) = \min_{\theta, x} \frac{1}{2}\left(\sum\sum (\theta_k^j x_k^i - y^{(i,j)})^2 + \frac{\lambda}{2}\sum\sum (\theta_k^j)^2\right)$$

$$\Downarrow$$

$$\frac{\lambda}{2}\left[(\theta_1^5)^2 + (\theta_2^5)^2\right]$$

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

$$\therefore (\theta^5)^T x^i = 0.$$

* How to solve this problem?

↳ Mean Normalization

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad ; \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \quad \begin{matrix} 10/4 \\ 5/2 \\ 4/2 \\ 9/4 \\ 5/4 \end{matrix}$$

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

* essentially making mean of all movies = 0

⇓

learn $\theta^j, x^i$ from this data

for user $j$, on movie $i$ predict

$$(\theta^j)^T x^i + \mu_i$$

for user 5 (Eve):

$$\theta^5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \therefore \underbrace{(\theta^5)^T x^i}_{0} + \mu^i$$

* In above case: we have normalized row: user not having rated movie.

* Alternatively, if we have a movie with no ratings we can normalize column: