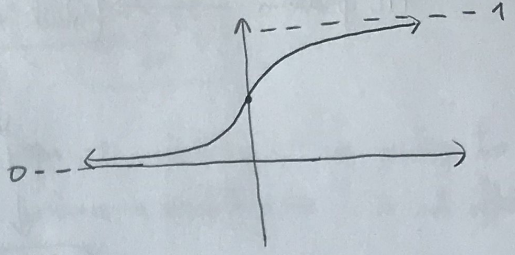
WEEK (7)

FROM LR TO SVM :

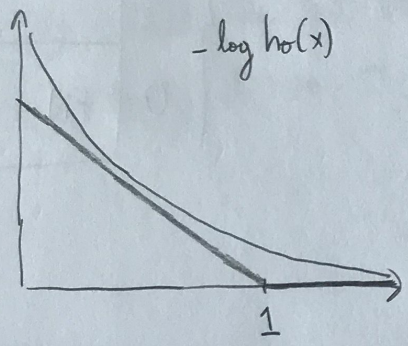
$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \rightarrow h_\theta(x) = g(z) \quad \text{where } z = \theta^T x$$



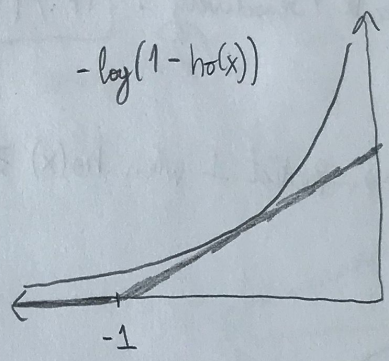
$$\begin{cases} y=1 \rightarrow h_\theta(x) \approx 1 \rightarrow \theta^T x \gg 0 \\ y=0 \rightarrow h_\theta(x) \approx 0 \rightarrow \theta^T x \ll 0 \end{cases}$$

$$\text{Cost} \rightarrow \underbrace{-y \log\left(\frac{1}{1+e^{-\theta^T x}}\right)}_{\text{cost}_1(\theta^T x^{(i)})} \underbrace{- (1-y) \log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)}_{\text{cost}_0(\theta^T x^{(i)})}$$

For $y=1$ → we want $\theta^T x \gg 0$



For $y=0$ → we want $\theta^T x \ll 0$



$$\text{SVM} \rightarrow \underbrace{\min_\theta \frac{1}{m} \sum_i^m y^{(i)} \text{cost}_1 + (1-y^{(i)}) \text{cost}_0}_{A} \underbrace{+ \frac{\lambda}{2m} \sum_{j=0}^n \theta_j^2}_{B}$$

$$\begin{cases} \text{LR} \rightarrow A + \lambda B \\ \text{SVM} \rightarrow CA + B \rightarrow \quad C \text{ is like } \frac{1}{\lambda} \end{cases}$$

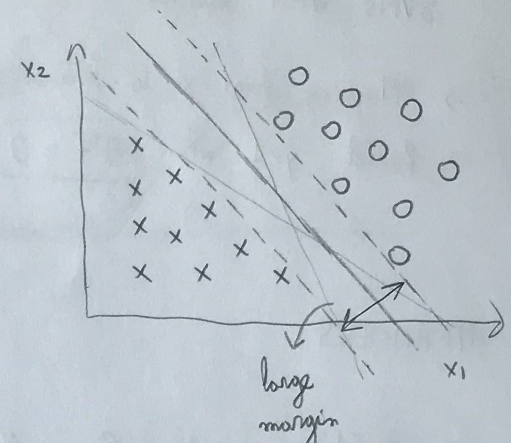
$$\hookrightarrow \boxed{\min_\theta C \sum_{i=1}^m \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^n \theta_j^2}$$

$$h_\theta(x) \begin{cases} 1 \rightarrow \theta^T x \geq 0 \\ 0 \rightarrow \text{other} \end{cases}$$

LARGE MARGIN INTUITION :

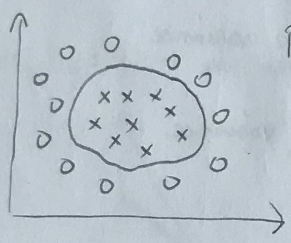
$$\begin{cases} y=1 \rightarrow \theta^T x \geq 1 \quad (\text{not just } \geq 0) \\ y=0 \rightarrow \theta^T x \leq -1 \quad (\text{not just } < 0) \end{cases}$$

↓ extra margin factor



→ C very large → sensitive to outliers

KERNELS :



Predict $y=1 \rightarrow \theta_0 + \theta_1 \underbrace{x_1}_{f_1} + \theta_2 \underbrace{x_2}_{f_2} + \theta_3 \underbrace{x_1 x_2}_{f_3} + \theta_4 \underbrace{x_1^2}_{f_4} + \theta_5 \underbrace{x_2^2}_{f_5} + \ldots \geq 0$

↓ How to choose the features ?

Ex: Proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$.

$$\begin{cases} f_1 = \text{similarity}(x, l^{(1)}) = e^{-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}} \\ f_2 = \ldots \\ f_3 = \ldots \end{cases}$$

↳ kernels (gaussian kernel here)

$$\begin{cases} \text{When } x \approx l^{(1)} \rightarrow f_1 \approx 1 \\ \text{When } x \text{ far from } l^{(1)} \rightarrow f_1 \approx 0 \end{cases}$$

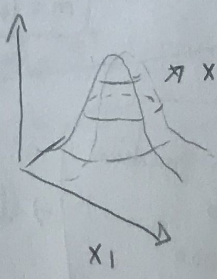
Plotting these similarity functions :

$$\begin{cases} l^{(1)} \rightarrow \text{centre of the gaussian} \\ \sigma^2 \rightarrow \text{larger } \sigma^2 \text{ means smoother function} \end{cases}$$

Ex: $\theta = \begin{bmatrix} -0.5 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ $\quad \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

→ x close to l^1 or l^2 → Predict 1

→ x far → Predict 0



SVMs WITH KERNELS

→ Mapping from x to features f

→ Predict $y=1$ when $\theta^T f \geq 0$

$x^{(i)}$ —MAPPING / KERNELS→ $f^{(i)}$

PARAMETERS:

→ $C\ (\frac{1}{\lambda})$
- → Large C → lower bias / high variance
- → Small C → higher bias / low variance

→ σ^2
- → Large σ^2 → more smooth = higher bias / lower variance
- → Small σ^2 → more abrupt = low bias / high variance

To avoid overfitting → pick small C and large σ^2

SVM OVERVIEW:

→ Specify C and kernel (if needed)
- → No kernel (i.e. linear) → $\theta^T x \geq 0$
- → Kernel → $\theta^T f \geq 0$

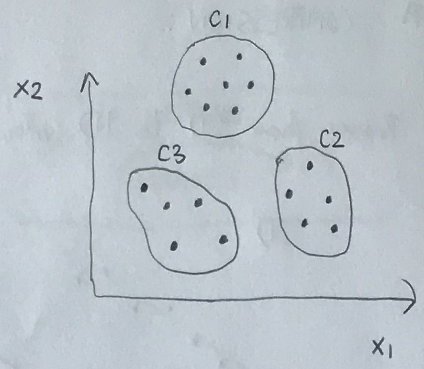
→ Use feature scaling before applying a kernel.

→ Multiclass SVM → Train K SVMs if K classes, and pick largest one

n → features
m → training samples

- $n \geq m$, $n = 10000$, $m = 10 \rightarrow 1000$ → Use LR or SVM without kernel
- n small, m medium: $n = 1 \rightarrow 1000$, $m = 10 \rightarrow 10000$ → Use SVM with gaussian kernel
- n small m large: $n = 1 \rightarrow 1000$, $m = 50000+$ → Add more features and use LR or SVM without kernel

WEEK (8)



→ K-MEANS CLUSTERING :

Input : → K (number of clusters)
→ Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

① Random init. of K cluster centroids $\mu_1, \mu_2 \ldots \mu_K$

② Do [for each sample : assign $c^{(i)}$ → index of cluster centroid closest to $x^{(i)}$] ← cluster assignment

[for each cluster : compute μ_K → average of points assigned to cluster k.] ← move centroids

OPT. OBJECTIVE (DISTORTION) →

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \| x^{(i)} - \mu_{c^{(i)}} \|^2$$

→ RANDOM INIT :

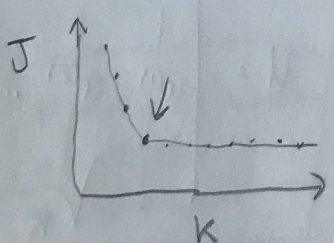
RUN ALGO MULTIPLE TIMES :

↳ Random. init. of μs to training data pts
↳ Run algo and get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$
↳ Compute $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

Pick clustering that gave the lowest cost J().

→ CHOOSING NUMBER OF CLUSTERS :

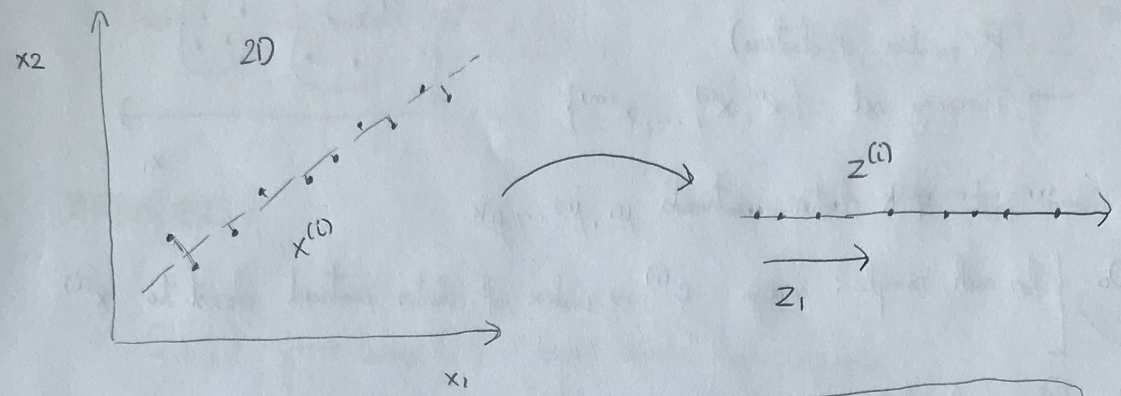
↳ Elbow method:



↳ Based on later purpose : use metric on how well it performs on a later task

DATA COMPRESSION :

→ Reduce from 2D to 1D when correlated features



$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^2$ → $z^{(1)}, z^{(2)}, \dots, z^{(m)} \in \mathbb{R}$

→ 3D from 2D : From x_1, x_2, x_3 $\in \mathbb{R}^3$ to z_1, z_2 $\in \mathbb{R}^2$

DATA VISUALIZATION :

Ex : Reduce from 50D (50 features) to 2D (2 features).

PRINCIPAL COMPONENT ANALYSIS (PCA) :

→ Reduce from n-dim to k-dim : find k vectors $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ onto which to project the data to minimize the projection error.

① Data processing :

TRAINING SET : $x^{(1)}, x^{(2)}, \dots, x^{(m)}$

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

$$x_j^{(i)} \rightarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

② ALGO : (from n-d to k-d)

COVARIANCE MATRIX : $\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T$ (n x n)

COMPUTE EIGENVECTORS : (n x n)

$$U = \begin{bmatrix} | & | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} & \dots & u^{(n)} \\ | & | & | & & | \end{bmatrix}$$

(k)

COMPUTE z

$$z = U_k^T x$$

Example 3D to 2D:

$\{x_1, x_2, x_3\} \in \mathbb{R}^3$

$$U_3 = \underbrace{\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{bmatrix}}_{3\times 3} \rightarrow U_2 = [U^{(1)}\ U^{(2)}] = \underbrace{\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \\ U_{31} & U_{32} \end{bmatrix}}_{3\times 2}$$

$$z = U_2^T x = \underbrace{\begin{bmatrix} U_{11} & U_{21} & U_{31} \\ U_{12} & U_{22} & U_{32} \end{bmatrix}}_{2\times 3} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{3\times 1} = \underbrace{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}_{(2\times 1)}$$

RECONSTRUCTION :

$$z = U_{reduced}^T \cdot x \longrightarrow x_{approx.} = U_{reduced} \cdot z$$

CHOOSING NUMBER OF PCs :

↳ Choose smallest value k so that

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - x^{(i)}_{approx.}\|^2}{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)}\|^2} \leq 0{,}01 \rightarrow 1\%$$

(numerator: average squared projection error; denominator: total variation in the data)

99% of variance is retained

PCA ADVICE :

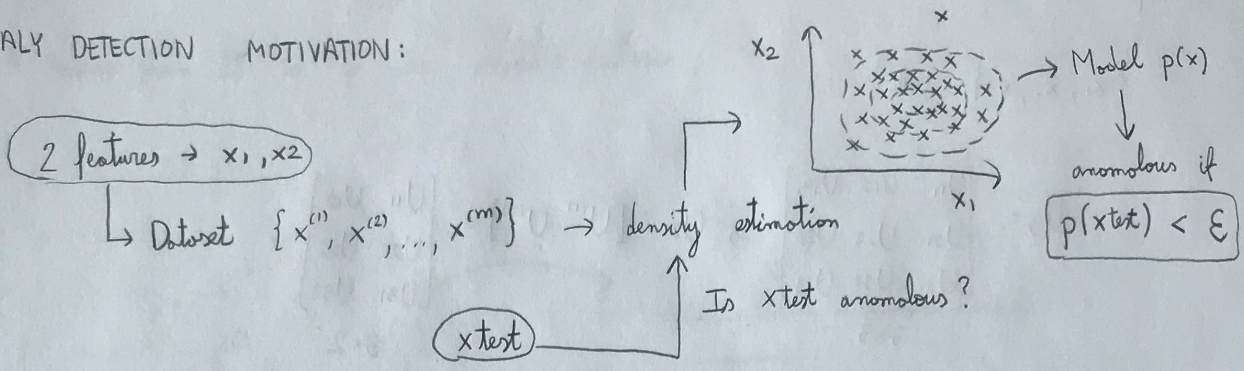
SUPERVISED LEARNING DATASET :
$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

$x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ → PCA → $z^{(1)}, z^{(2)}, \ldots, z^{(m)}$

TRAIN ALGO ON:
$(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \ldots (z^{(m)}, y^{(m)})$

TEST ON:
$x_{cv}^{(i)} \rightarrow z_{cv}^{(i)}$
$x_{test}^{(i)} \rightarrow z_{test}^{(i)}$

ANOMALY DETECTION MOTIVATION:



GAUSSIAN (NORMAL) DISTRIBUTION:

$$x \sim \mathcal{N}(\mu, \sigma^2) \rightarrow p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

PARAM. ESTIMATION

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

DENSITY ESTIMATION:

$$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\} \rightarrow \begin{cases} x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2) \\ x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2) \\ \vdots \\ x_n \sim \mathcal{N}(\mu_n, \sigma_n^2) \end{cases} \rightarrow p(x) = p(x_1; \mu_1, \sigma_1^2) \cdot p(x_2; \mu_2, \sigma_2^2) \cdot \dots \cdot p(x_n; \mu_n, \sigma_n^2)$$

ANOMALY DETECTION ALGO:

1. Relevant features selection
2. Fit params. $\mu_1 \rightarrow \mu_n$ and $\sigma_1^2 \rightarrow \sigma_n^2$

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

3. Given new example x:

Compute $$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

ANOMALY IF $p(x) < \varepsilon$

ANOMALY DETECTION DEV. AND. EVAL.

- $y=0$ → normal
- $y=1$ → anomalous

TRAINING SET
- $x^{(1)}, x^{(2)}, \dots, x^{(m)}$
- Only normal examples

E.g. Training: 6000 normal
CV → 2000 normal, 10 anom.
Test → 2000 normal, 10 anom.

CV SET
- $(x_{cv}^{(1)}, y_{cv}^{(1)}) \dots (x_{cv}^{(m\,cv)}, y_{cv}^{(m\,cv)})$
- ↳ Some labelled $y=1$

TEST SET
- $(x_t^{(1)}, y_t^{(1)}) \dots (x_t^{(mt)}, y_t^{(mt)})$
- ↳ Some labelled $y=1$

① Fit $p(x)$ on the training set → only normal samples

② On CV / test sample x → predict
- $y=1$ when $p(x) < \varepsilon$
- $y=0$ when $p(x) \geq \varepsilon$

→ Use CV to choose value of ε

③ Use relevant metrics to evaluate algo:
- → TP / FP / FN / TN
- → Precision, recall
- → F1 score

ANOMALY DETECTION VS SUPERVISED LEARNING:

- → Very small number of anomalous samples
  - ↳ estimate $p(x)$ from $y=0$
- → Many different types of anomalies
- → Future anomalies may be very different

- → Large number of both labels
- → Future positive samples likely to be similar to those in the training set.

FEATURE SELECTION:

→ If non gaussian: $\begin{cases} x_1 \longrightarrow \log(x_1) \\ x_1 \rightarrow x_1^{1/3} \dots \end{cases}$

→ Choose new features that might be small/large for anomalies: $\begin{cases} x_3 \rightarrow \frac{x_1}{x_2} \\ x_3 \rightarrow \frac{x_1^2}{x_2} \end{cases}$

MULTIVARIATE GAUSSIAN DISTRIBUTION:

↳ When there may be a correlation between features

↳ Model $p(x)$ in one go with $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n \times n}$ (covariance matrix)

2 features

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

→ μ → translation of centre of the distribution

→ diagonals in Σ : larger = smoother

→ other elements in Σ : correlation between x_1 and x_2 → elliptic

↓ ALGO.

① Fit model $p(x)$ using $\begin{cases} \mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \\ \Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T \end{cases}$

② Given new sample x, compute: $\boxed{p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}$

③ Flag anomaly if $\boxed{p(x) < \varepsilon}$

WEEK 10

TYPES OF GRADIENT DESCENT ALGOS

① BATCH GD

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

↓ REPEAT

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

for every $j = 0 \dots n$

↓

One iteration = all m samples

② STOCHASTIC GD

↓ Randomly shuffle training samples

↓ REPEAT 1-10x

FOR $i = 1, \dots, m$

$$\theta_j := \theta_j - \alpha\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

FOR $j = 0 \dots n$

↓

One iteration = only one sample

③ MINI-BATCH GD

mini batch size = b

$1 \leq b \leq m$

E.g. $\begin{cases} b = 10 \\ m = 100 \end{cases}$

LOOP $\begin{cases} \text{for } i = 1, 11, 21, 31, \dots, 91 \\ \quad \theta_j := \theta_j - \frac{\alpha}{10}\sum_{k=1}^{i+9}\left(h_\theta(x^{(k)}) - y^{(k)}\right)x_j^{(k)} \\ \quad (\text{for every } j = 0, \dots, n) \end{cases}$

MAP - REDUCE

↓ Batch GD

- Machine 1 → use samples 1 to 100 → $temp_j^{(1)} = \sum_{i=1}^{100} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$
- Machine 2 → 101 to 200 → $temp_j^{(2)}$
- Machine 3 → 201 to 300 → $temp_j^{(3)}$
- Machine 4 → 301 to 400 → $temp_j^{(4)}$

↘ COMBINE ALL

$$\theta_j := \theta_j - \alpha \frac{1}{400} \left[temp_j^{(1)} + temp_j^{(2)} + temp_j^{(3)} + temp_j^{(4)} \right]$$

for $j = 0 \to n$