Stanford-CS229 Machine Learning

Lecturer: Andrew Ng
(吴恩达)

Lecture 1

App Feild: Computer Vision. Biology. Economy. Robotics.
NLP...

MATLAB. Statistics. Linear Algebra. Algorithm & Data Structure

1. Supervised Learning $\begin{cases} \text{Regression} \\ \text{Classfication : SVM} \end{cases}$

2. Learning Theory: How & Why algo.'s work.

3. Unsupervised Learning. $\begin{cases} \text{Image Processing} \\ \text{Cooktail Party Problem: ICA.} \end{cases}$

```
[w,s,v]=svd((repmat(sum(x.*x,1), size(x,1), 1).*x)*x');
```

4. Reinforcement Learning: Good dog / Bad dog.

Lecture 2

- Linear Regression
- Gradient Descent
- Normal Equations

autonomous driving: regression.

$$h(x) = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$= \sum_{j=0}^{n} \theta_j x_j = \theta^T x \quad (n=2)$$

$$x_0 \stackrel{\text{def}}{=} 1$$

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

$$J(\theta) \stackrel{def}{=} \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

Batch Gradient Descent

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial\theta_j} J(\theta)$$

Repeat until convergence:

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

Stochastic Gradient Descent (much faster)

Repeat {

For $i = 1$ to m {

$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)} \quad (\text{For All } j$$

}

}

$$\nabla_\theta J = \begin{pmatrix} \frac{\partial J}{\partial\theta_0} \\ \vdots \\ \frac{\partial J}{\partial\theta_n} \end{pmatrix} \in \mathbb{R}^{n+1}$$

Gradient Descent:

$$\vec{\theta} := \vec{\theta} - \alpha \nabla_\theta J \quad \in \mathbb{R}^{n+1}$$

design matrix $X \stackrel{def}{=} \begin{pmatrix} \text{---} (x^{(1)})^T \text{---} \\ \text{---} (x^{(2)})^T \text{---} \\ \vdots \\ \text{---} (x^{(m)})^T \text{---} \end{pmatrix}$

$$X\theta = \begin{pmatrix} x^{(1)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{pmatrix} = \begin{pmatrix} h_\theta(x^{(1)}) \\ \vdots \\ h_\theta(x^{(m)}) \end{pmatrix} \qquad \vec{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

$f: \mathbb{R}^{m \times n} \mapsto \mathbb{R}$

$f(A), \quad A \in \mathbb{R}^{m \times n}$

$$\nabla_A f(A) = \begin{pmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{pmatrix}$$

If $A \in \mathbb{R}^{n \times n}$, $\operatorname{tr} A = \sum_{i=1}^{n} A_{ii}$

Fact: $\boxed{\begin{array}{l} \operatorname{tr} AB = \operatorname{tr} BA \\ \operatorname{tr} ABC = \operatorname{tr} CAB = \operatorname{tr} BCA \end{array}}$

$f(A) = \operatorname{tr} AB$: $\boxed{\nabla_A \operatorname{tr} AB = B^T}$

$\boxed{\operatorname{tr} A = \operatorname{tr} A^T}$

If $a \in \mathbb{R}$: $\boxed{\operatorname{tr} a = a}$

$\boxed{\nabla_A \operatorname{tr} ABA^T C = CAB + C^T AB^T}$

$$X\theta - y = \begin{pmatrix} h(x^{(1)}) - y^{(1)} \\ \vdots \\ h(x^{(m)}) - y^{(m)} \end{pmatrix}$$

$$\frac{1}{2}(X\theta - y)^T (X\theta - y) = \frac{1}{2} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)}\right)^2 = J(\theta)$$

$$\nabla_\theta J(\theta) \stackrel{\text{set}}{=} \vec{0}$$

3

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

$$= \frac{1}{2}\nabla_\theta\left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y\right)$$

$$= \frac{1}{2}\nabla_\theta \, tr\left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y\right) \quad (\text{scaler elem})$$

$$= \frac{1}{2}\left(\nabla_\theta tr\, \theta\theta^T X^T X - \nabla_\theta tr\, y^T X\theta - \nabla_\theta tr\, y^T X\theta\right)$$

$$\nabla_\theta tr\, \underbrace{\theta}_{A}\underbrace{I}_{B}\underbrace{\theta^T}_{A^T}\underbrace{X^T X}_{C} = \underbrace{X^T X}_{C}\underbrace{\theta}_{A}\underbrace{I}_{B} + \underbrace{X^T X}_{C^T}\underbrace{\theta}_{A}\underbrace{I}_{B^T}$$

$$\nabla_\theta tr(\underbrace{y^T X}_{B}\underbrace{\theta}_{A}) = \underbrace{X^T y}_{B^T}$$

$$\nabla_\theta J(\theta) = \frac{1}{2}\left(X^T X\theta + X^T X\theta - X^T y - X^T y\right)$$

$$= X^T X\theta - X^T y \overset{set}{=} 0$$

$$\boxed{X^T X\theta = X^T y}$$ Normal Eq'ns

$$\theta = (X^T X)^{-1} X^T y$$

Lecture 3

Linear Regression → Locally weighted regression

↓

Probabilistic Interpretation

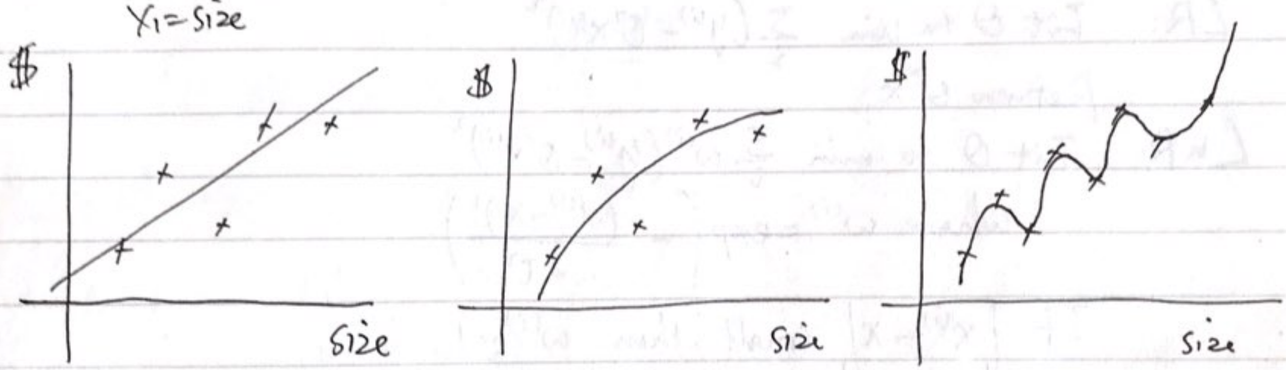
↓

Logistic Regression → Digression: Perceptron

↓

Newton's Method

$X_1 = \text{size}$

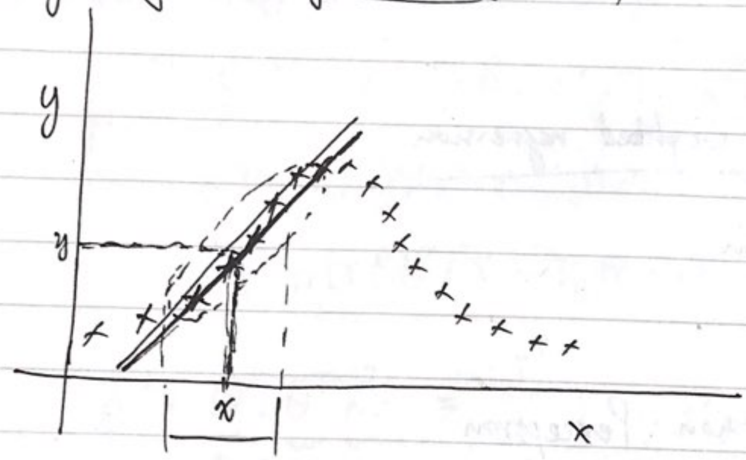


$\theta_0 + \theta_1 x_1$ $\qquad$ $\theta_0 + \theta_1 x_1 + \theta_2 \underset{\uparrow \, x_2}{x_1^2}$ $\qquad$ $\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \cdots + \theta_7 x_1^7$

"underfitting" $\qquad$ "overfitting"

{ "Parametric Learning algorithm" (θ's - fixed set of parameters)
"Non-parametric Learning algorithm" (# of parameters grows with m)

Locally weighted regression (Loess)



To evalute h at a certain x

LR: Fit θ to min $\sum_i (y^{(i)} - \theta^T x^{(i)})^2$

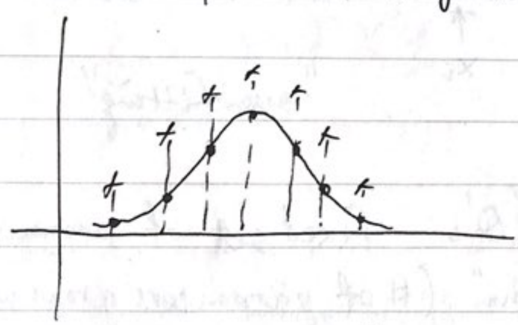
Return $\theta^T x$

LWR: Fit θ to min $\sum_i \omega^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$

where $\omega^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$

If $|x^{(i)} - x|$ small, then $\omega^{(i)} \approx 1$

If $|x^{(i)} - x|$ large, then $\omega^{(i)} \approx 0$.

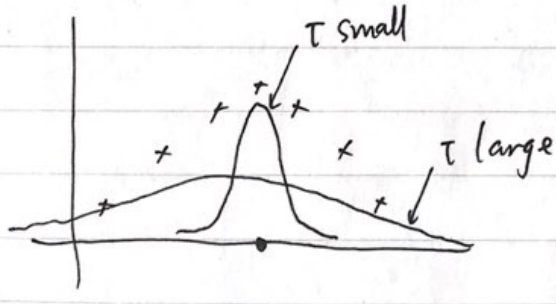


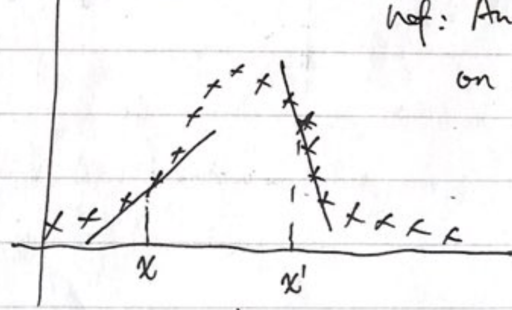
τ: bandwidth parameter

(model selection)

For large datas[et]

ref: Andrew Moore on KD-trees





(autonomous helicopter use this algo.)

Probabilistic interpretation

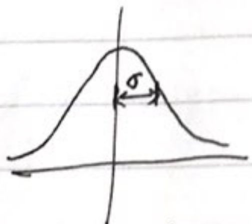
(present one set of assumptions)

Assume $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$

$\varepsilon^{(i)}$ = error

$\varepsilon^{(i)} \sim N(0, \sigma^2)$

$$P(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\varepsilon^{(i)})^2}{2\sigma^2}}$$



$\cancel{P(y^{(i)} | x^{(i)}, \theta)}$

θ is parameter, not random variable here.

$$P(y^{(i)} | x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

$$y^{(i)} | x^{(i)}; \theta \sim N(\theta^T x^{(i)}, \sigma^2)$$

$\varepsilon^{(i)}$s are IID:

(independently identically distributed).

likelihood function: different view of probability function

$L(\theta) = P(\vec{y} | X; \theta)$ ← probability of data

likelihood of parameter

$$= \prod_{i=1}^{m} P(y^{(i)} | x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

Maximum Likelihood: (MLE)

Choose θ to maximum $L(\theta)$

$$= P(\vec{y} \mid X ; \theta)$$

$$\ell(\theta) = \log L(\theta)$$

$$= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

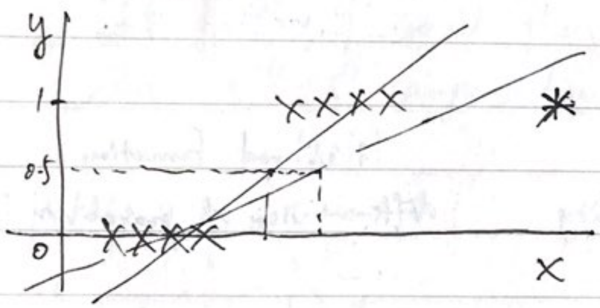
$$= \sum_{i=1}^{m} \log \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\cdots} \right)$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} - \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

so, maximum $\ell(\theta)$ is the same as minimize

$$\sum_{i=1}^{m} \frac{1}{2} (y^{(i)} - \theta^T x^{(i)})^2 = J(\theta)$$

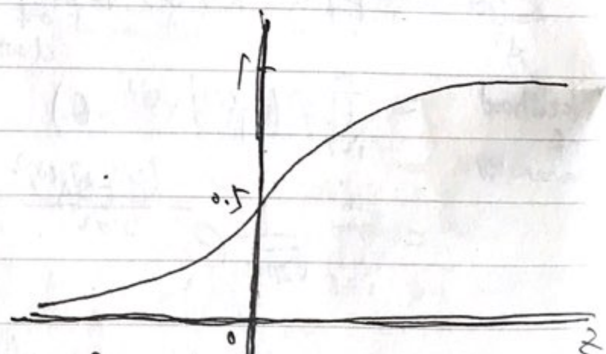
Classification $\quad y \in \{0, 1\}$



$h_\theta(x) \in [0, 1]$.

Choose

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



$$g(z) = \frac{1}{1 + e^{-z}}$$

sigmoid function

logistic function

$$P(y=1 \mid x;\theta) = h_\theta(x)$$

$$P(y=0 \mid x;\theta) = 1 - h_\theta(x)$$

$$P(y \mid x;\theta) = h_\theta(x)^y (1-h_\theta(x))^{1-y}$$

$$L(\theta) = P(\vec{y} \mid X;\theta) = \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)};\theta)$$

$$= \prod_{i=1}^{m} h(x^{(i)})^{y^{(i)}} (1-h(x^{(i)}))^{1-y^{(i)}}$$

$$l(\theta) = \log L(\theta)$$

$$= \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))$$

$$\boxed{\theta := \theta + \alpha \nabla_\theta l(\theta)} \quad (\text{maximize } l(\theta))$$

$$\frac{\partial}{\partial\theta_j} l(\theta) = \sum_{i=1}^{m} \cdots = \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \cancel{(h_\theta(x^{(i)}))(1-h_\theta(x^{(i)}))} \overset{\text{set}}{=} 0.$$

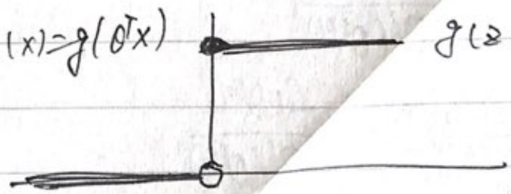
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - \underline{h_\theta(x^{(i)})}) x_j^{(i)}$$

nonlinear!
logistic function

Digression: Perceptron

$h_\theta(x) = g(\theta^T x)$

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$



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

Lecture 4

Logistic Regression
- Newton's method

Exponential Family

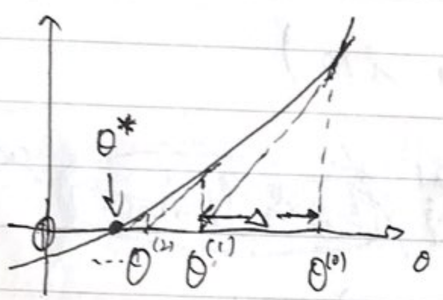
Generalized Linear Model (GLM)

$$P(y=1|x;\theta) = h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$\ell(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))$$

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h(x^{(i)})) x_j^{(i)}$$

Newton's Method (much faster)



$f(\theta)$ Find θ s.t. $f(\theta)=0$

$$f'(\theta^{(0)}) = \frac{f(\theta^{(0)})}{\Delta}$$

$$\Delta = \frac{f(\theta^{(0)})}{f'(\theta^{(0)})}$$

$$\theta^{(1)} = \theta^{(0)} - \frac{f(\theta^{(0)})}{f'(\theta^{(0)})}$$

$$\theta^{(t+1)} = \theta^{(t)} - \frac{f(\theta^{(t)})}{f'(\theta^{(t)})}$$

max $\ell(\theta)$ want θ st. $\ell'(\theta)=0$

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\ell'(\theta^{(t)})}{\ell''(\theta^{(t)})}$$

0.01 error $\rightarrow$ 0.001 error

$\rightarrow$ 0.000 001 error

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1} \nabla_\theta \ell$$

where H is the Hessian Matrix

$$H_{ij} = \frac{\partial^2 \ell}{\partial \theta_i \partial \theta_j}.$$

$p(y|x;\theta)$

$y \in \mathbb{R}$: Gaussian → Least Squares

$y \in \{0,1\}$: Bernoulli → Logistic Regression. $g(z) = \frac{1}{1+e^{-z}}$

Bernoulli (ϕ): $P(y=1;\phi) = \phi$.

$N(\mu, \sigma^2)$

<u>Exponential Family</u>

$$P(y;\eta) = b(y)\, e^{\eta^T T(y) - a(\eta)}$$

η: natural parameter

$T(y)$: sufficient Statistic

(usually $T(y) = y$)

(a, b, T).

① $\text{Ber}(\phi)$ $\quad P(y=1;\phi) = \phi$.

$$
\begin{aligned}
p(y;\phi) &= \phi^y (1-\phi)^{1-y} \\
&= \exp\left(\log \phi^y (1-\phi)^{1-y}\right) \\
&= \exp\left(y \log\phi + (1-y)\log(1-\phi)\right) \\
&= \exp\Big(\underbrace{\log\frac{\phi}{1-\phi}}_{\eta}\ \underbrace{y}_{T(y)} + \underbrace{\log(1-\phi)}_{-a(\eta)}\Big)
\end{aligned}
$$

$b(y) = 1$

$$
\begin{cases}
\eta = \log\frac{\phi}{1-\phi} \Rightarrow \phi = \frac{1}{1+e^{-\eta}} & \text{Attention!} \\
a(\eta) = -\log(1-\phi) = \log(1+e^{\eta}) \\
T(y) = y \\
b(y) = 1
\end{cases}
$$

② Gaussian:

$N(\mu, \sigma^2)$ set $\sigma^2 = 1$ for simplicity.

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2}} = \cdots$$

$$= \underbrace{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2}}_{b(y)} \cdot e^{\mu y - \frac{1}{2}\mu^2}$$

$\eta = \mu$ $T(y) = y$ $a(\eta) = \frac{1}{2}\mu^2 = \frac{1}{2}\eta^2$

GLMs

Assume:

(1) $y | x; \theta \sim \text{ExpFamily}(\eta)$

(2) Given x, goal is to output $E[T(y)|x]$.

want $h(x) = E[T(y)|x]$

(3) design choice (assumption):

$$\eta = \theta^T x$$

$$(\eta_i = \theta_i^T x \text{ if } \eta \in \mathbb{R}^k)$$

Bernoulli:

$y | x; \theta \sim \text{ExpFamily}(\eta)$ $\quad$ $\underline{y \in \{0, 1\}}$

For fixed x, θ. algorithm output:

$$h_\theta(x) = E[y | x; \theta] = p(y=1 | x; \theta)$$

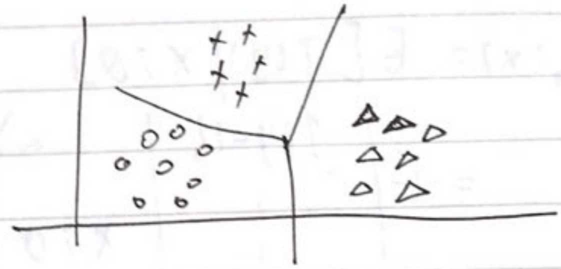
$$= \phi = \frac{1}{1+e^{-\eta}} = \frac{1}{1+e^{-\theta^T x}}$$

$g(\eta) = E[y; \eta] = \frac{1}{1+e^{-\eta}}$: canonical response function

g^{-1} : canonical link function

Multinomial:

$y \in \{1, \dots, k\}$



Parameters: $\phi_1, \phi_2, \dots, \phi_k$

$\therefore P(y=i) = \phi_i$

$\phi_k = 1 - (\phi_1 + \dots + \phi_{k-1})$

Parameters: $\phi_1, \phi_2, \dots, \phi_{k-1}$

$T(1) = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad T(2) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \cdots \in \mathbb{R}^{k-1}$

$T(k-1) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad T(k) = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

$1\{\text{True}\} = 1 \quad 1\{2+2=4\} = 1$

$1\{\text{False}\} = 0 \quad 1\{4=5\} = 0$

$T(y)_i = 1\{y=i\}$

$$P(y) = \phi_1^{1\{y=1\}} \phi_2^{1\{y=2\}} \cdots \phi_k^{1\{y=k\}}$$

$$= \phi_1^{T(y)_1} \phi_2^{T(y)_2} \cdots \phi_{k-1}^{T(y)_{k-1}} \cdot \phi_k^{1 - \sum_{j=1}^{k-1} T(y)_j}$$

$$= \dots$$

$$= b(y) e^{\eta^T T(y) - a(\eta)}$$

where $\eta = \begin{pmatrix} \log(\phi_1/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{pmatrix} \in \mathbb{R}^{k-1}$

$$\phi_i = \frac{e^{\eta_i}}{1 + \sum_{j=1}^{k-1} e^{\eta_j}} \quad (i = 1, \dots, k-1)$$

$$= \frac{e^{\theta_i^T x}}{1 + \sum_{j=1}^{k-1} e^{\theta_j^T x}} \quad (i = 1, \dots, k-1)$$

$a(\eta) = -\log(\phi_k)$

$b(y) = 1$

$$h_\theta(x) = E[T(y) \mid x;\theta]$$

$$= E\left(\begin{matrix} 1\{y=1\} \\ \vdots \\ 1\{y=k-1\} \end{matrix} \middle| \; x;\theta \right) = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \end{pmatrix}$$

$$= \begin{pmatrix} e^{\theta_1^T x} / \left(1+\sum_{j=1}^{k-1} e^{\theta_j^T x}\right) \\ \vdots \\ e^{\theta_{k-1}^T x} / \left(1+\sum_{j=1}^{k-1} e^{\theta_j^T x}\right) \end{pmatrix}$$

↳ <u>Softmax Regression</u>

$y \in \{1, \dots, k\}$ $\quad (x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$.

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \phi_1^{1\{y^{(i)}=1\}} \phi_2^{1\{y^{(i)}=2\}} \cdots \phi_k^{1\{y^{(i)}=k\}}$$

$$\phi_i = \frac{e^{\theta_i^T x}}{1+\sum_{j=1}^{k-1} e^{\theta_j^T x}} \quad (i = 1, \dots, k-1)$$

$\theta_1, \dots, \theta_{k-1} \in \mathbb{R}^{n+1}$

$\boxed{\text{Lecture 5}}$

Generative Learning Algorithms

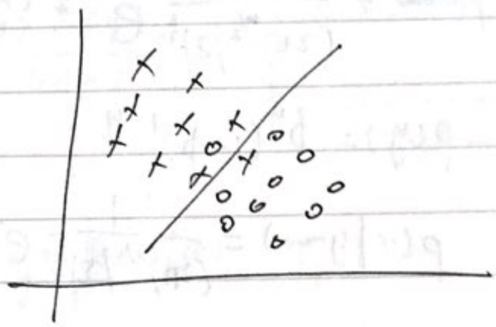
GDA

Disgression: Gaussians

Generative & Discriminative comparison

Naive Bayes

Laplace Smoothing



Discriminative

- learns $p(y|x)$ conditional p.d.
- or learns $h_\theta(x) \in \{0, 1\}$ directly

Generative

$p(x|y)$, $p(y)$. - learns joint p.d.

features class label

$$p(y=1|x) = \frac{p(x|y=1)\,p(y=1)}{p(x)}$$

$$p(x) = p(x|y=0)\,p(y=0) + p(x|y=1)\,p(y=1)$$

Assume $x \in \mathbb{R}^n$, continuous valued.

Gaussian Discriminant Analysis:

Core Assumption: $p(x|y)$ is Gaussian

$z \sim N(\vec{\mu}, \Sigma)$ — mean, covariance $\quad \Sigma = E[(x-\mu)(x-\mu)^T]$

$$p(z) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)}$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)}$$

Joint Likelihood (generative ~~discriminative~~ model)

$$\ell(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^m p(x^{(i)}, y^{(i)})$$

$$= \log \prod_{i=1}^m p(x^{(i)}|y^{(i)})\, p(y^{(i)})$$

logistic regression:

$$\ell(\theta) = \log \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta)$$

conditional likelihood (~~generative~~ discriminant model)

max ℓ w.r.t. $\phi, \mu_0, \mu_1, \Sigma$.
(with respect to)

$$\phi = \frac{\sum_{i=1}^m y^{(i)}}{m} = \frac{\sum_{i=1}^m 1\{y^{(i)}=1\}}{m}$$

$$\mu_0 = \frac{\sum_{i=1}^m 1\{y^{(i)}=0\}\, x^{(i)}}{\sum_{i=1}^m 1\{y^{(i)}=0\}}$$

sum of $x^{(i)}$ for which $y^{(i)}=0$ (label 0)

examples with label 0

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$$

$\Sigma = \ldots$ [See the lecture notes].

$\min (x-5)^2 = 0$

$\arg\min_x (x-5)^2 = 5$

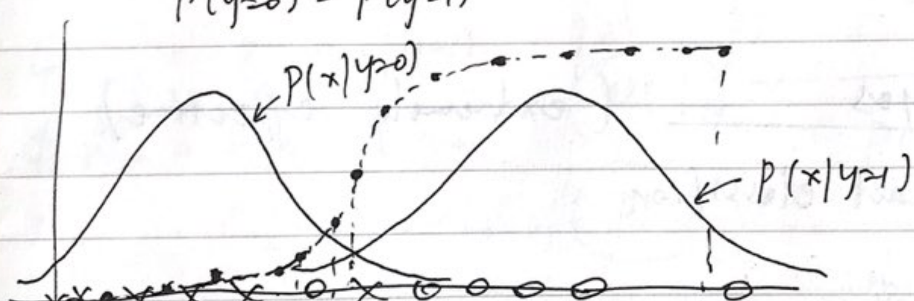
Predict:

$$\arg\max_y p(y|x) = \arg\max_y \frac{p(x|y)\,p(y)}{p(x)}$$

$$= \arg\max_y p(x|y)\,p(y)$$

If $p(y)$ is uniform: $\arg\max_y p(x|y)$

$p(y=0) = p(y=1)$



$$p(y=1|x) = \frac{p(x|y=1)\,p(y=1)}{p(x)} \quad \phi$$

$p(x|y)$ - Gaussian

$p(y|x)$ - sigmoid

$$p(x) = p(x|y=0)\,p(y=0) + p(x|y=1)\,p(y=1)$$

[Advantage of generative learning algo.]
/ use more information of data.
require less data.

Assume $x|y \sim$ Gaussian (more generally, Exp Family)

$\Downarrow$ $\not\Uparrow$

logistic posterior for $p(y=1|x)$

Gamma
Beta
⋮

[Advantages of discriminative...
more robust on making assumption

$$\left\{\begin{array}{l} x|y=1 \sim \text{Poisson}(\lambda_1) \\ x|y=0 \sim \text{Poisson}(\lambda_0) \end{array}\right. \quad \begin{array}{l} \text{Exp Family}(\eta_1) \\ \text{Exp Family}(\eta_0) \end{array}$$

$\Rightarrow p(y=1|x)$ is logistic

Naive Bayes (extremely effective)

Spam email classifier

$y \in \{0, 1\}$

feature vector

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \text{a} \\ \text{ardvark} \\ \text{ausnorth} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{cs229} \\ \vdots \\ \text{zymurgy} \end{array}$$

$p(x|y)$

$x \in \{0,1\}^n$

$n = 50{,}000$ (dictionary)

$2^{50{,}000}$

$\therefore$ multinomial X

#parameter $= 2^{50{,}000}$

Assume: x_i's are conditionally independent given y.

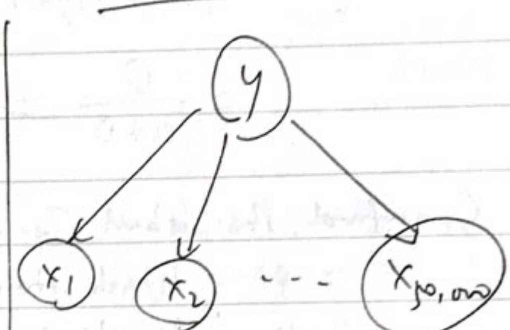
(Naive Bayes Assumption)

$$p(x_1, x_2, \dots, x_{50,000} \mid y)$$

$$= p(x_1 \mid y)\, p(x_2 \mid y, x_1) \cdots p(x_{50,000} \mid y, x_1, x_2, \dots, x_{49,999})$$

$$\quad\; p(x_1 \mid y) \quad p(x_2 \mid y) \quad\quad p(x_{50,000} \mid y)$$

$$= \prod_{i=1}^{n} p(x_i \mid y)$$



Parameters:

$$\phi_{i|y=1} = p(x_i = 1 \mid y=1)$$

$$\phi_{i|y=0} = p(x_i = 1 \mid y=0)$$

$$\phi_y = p(y=1)$$

$p(x|y)$

$p(y)$

$$p(y|x) = p(x|y)\,p(y)$$

Joint Likelihood: $\mathcal{L}(\phi_y, \phi_{i|y=1}, \phi_{i|y=0}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$.

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1,\, y^{(i)}=1\}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}} \qquad \phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1,\, y^{(i)}=0\}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}{m}$$

Training set: $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$.

Around June: deadline of ~~next~~ conference. $\boxed{\textbf{NIPS}}$

"first appear" word

$$\underline{p(x_{30,000}=1 \mid y=1) = 0} \leftarrow \text{PROBLEM!!}$$

$$p(x_{30,000}=1 \mid y=0) = 0.$$

$$P(y=1 \mid x) = \frac{p(x \mid y=1)\, p(y=1)}{p(x \mid y=1)p(y=1) + p(x \mid y=0)p(y=0)} \qquad \prod_{i=1}^{50,000} p(x_i \mid y=1) = 0.$$

$$= \frac{0}{0+0} = \frac{0}{0}.$$

Stanford Basketball Team		Win
2-8	Washington	0
2-11	Washington	0
2-22	USC	0
2-24	UCLA	0
3-8	USC	0
3-15	Louisville	?

[in fact: 0]

Laplace Smoothing

Bayes prior

$$p(y=1) = \frac{\#\text{"1"s} + 1}{\#\text{"0"s}_{+1} + \#\text{"1"s}_{+1}} = \frac{0+1}{5_{+1} + 0_{+1}} = \frac{1}{7}.$$

More generally:

If $y \in \{1, 2, \dots, k\}$: ~~$P(y=j) = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1, y^{(i)}=1\}+1}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}+2}$~~

$$P(y=j) = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=j\} + 1}{m + k}$$

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1, y^{(i)}=1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)}=1\} + 2}$$

Lecture 6

- Naive Bayes
  - Event Models
- Neural Networks
- Support Vector Machines (most effective supervised learning algo.)

NB: Generative Learning Algorithm.

$$P(x|y) = \prod_{i=1}^{n} p(x_i|y)$$

$$p(y)$$

$$\arg\max_y p(y|x) = \arg\max_y p(x|y)p(y)$$

$$X = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ \vdots \end{pmatrix} \begin{matrix} a \\ ab \\ abc \\ \vdots \\ \vdots \end{matrix}$$

Multi-variate Bernoulli Event Model.

$X_i \in \{0, 1\}$

n = #words in dict (50,000)

generally:

$x_i \in \{1, 2, \ldots, k\}$. $k = n$

$$p(x|y) = \prod_{i=1}^{n} p(x_i|y)$$

multinomial

Living area	<500	500-1000	1000-1500	1500-2000	>2000
x_i =	1	2	3	4	5

Enough for Text Classification

Multinomial Event Model

① do better than Multi-variate Bernoulli E.M

② Unigram Model in NLP (against to High-order Markov Model)

$$(x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)})$$

n_i = #words in this email

$x_j \in \{1, 2, \ldots, 50000\}$.

index of dict where the word sit.

$$p(x, y) = \left(\prod_{j=1}^{n_i} p(x_j|y)\right) p(y)$$

parameters:

$$\phi_{k|y=1} = P(x_j = k \mid y=1)$$

$$\phi_{k|y=0} = P(x_j = k \mid y=0)$$

$$\phi_y = P(y=1)$$

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\left(1\{y^{(i)}=1\} \cdot \sum_{j=1}^{n_i} 1\{x_j^{(i)}=k\}\right) + 1}{\sum_{i=1}^{m}\left(1\{y^{(i)}=1\} \cdot n_i\right) + 50000\ ?}$$

$x \in \{1, 2, \dots, l\}$

$$P(x=k) = \frac{\#\text{observation of } "x=k" + 1}{\#\text{observation of } x + l}$$

✓ Laplace Smoothing

MLE:

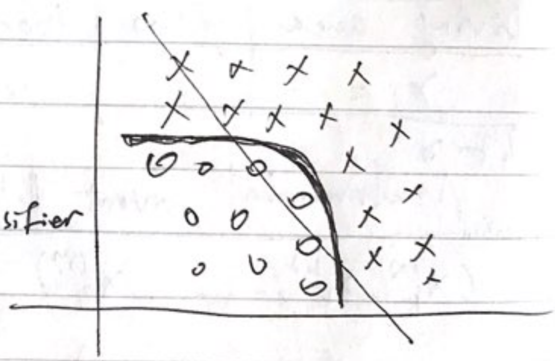
$$\ell\left(\phi_{k|y=1}, \phi_{k|y=0}, \phi_y\right)$$

$$= \log \prod_{i=1}^{m} P\left(x^{(i)}, y^{(i)}; \phi_{k|y=1}, \phi_{k|y=0}, \phi_y\right)$$

$$= \log \prod_{i=1}^{m} \prod_{j=1}^{n_i} P\left(x_j^{(i)} \mid y^{(i)}; \phi_{k|y=1}, \phi_{k|y=0}\right) P\left(y^{(i)} \mid \phi_y\right)$$

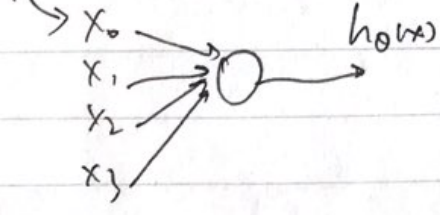
Nonlinear Classifiers

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$$ linear classifier

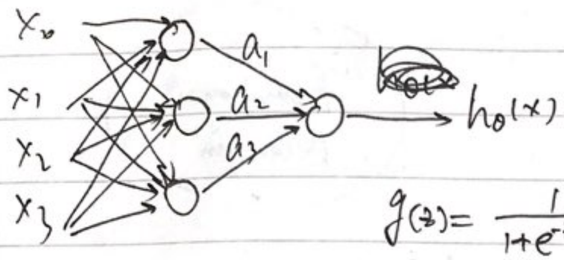


$x \mid y=1 \sim \text{Exp Family}(\eta_1)$

$x \mid y=0 \sim \text{Exp Family}(\eta_0)$



Neural Network.



$$g(z) = \frac{1}{1+e^{-z}}$$

$$a_1 = g(x^T\theta^{(1)})$$
$$a_2 = g(x^T\theta^{(2)})$$
$$a_3 = g(x^T\theta^{(3)})$$
$$h_\theta(x) = g(\vec{a}^T\theta^{(4)}) \qquad \vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

↳ $\text{Function}(\theta^{(1)}, \theta^{(2)}, \theta^{(3)})$

cost function:
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))^2$$
gradient descent : back propagation in Neural Network

* Yann LeCun @NYU

① Hammerton Digit Recognition
② Convolutional Neural Network

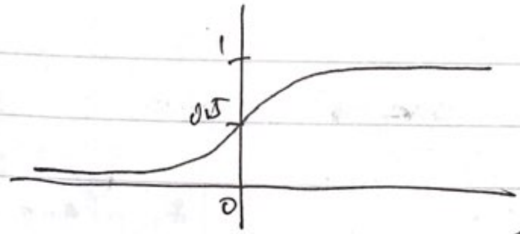
his system called LeNet

* Terry Sejnowski

NETtalk read text

landmark in NN early history

SVM



Intuition

① Compute $\theta^T x$.

Predict "1" iff. $\theta^T x \geq 0$

Predict "0" iff $\theta^T x < 0$

If $\theta^T x \gg 0$ very "confident" that $y=1$.

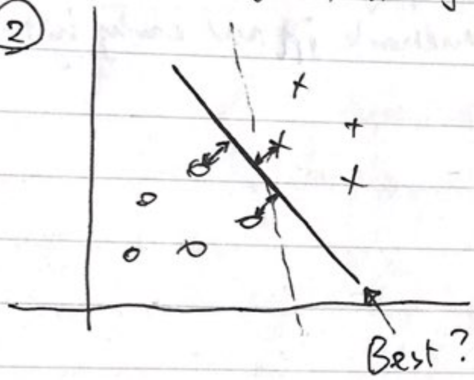
If $\theta^T x \ll 0$ very "confident" that $y=0$.

"Functional Margin"

Nice: $\forall i$, st. $y^{(i)}=1$, have $\theta^T x^{(i)} \gg 0$.

$\forall i$, st. $y^{(i)}=0$, have $\theta^T x^{(i)} \ll 0$.

②



Assume: linearly seperatable training set

"Geometric Margin".

Notation:

$y \in \{-1, +1\}$

Have h, output values in $\{-1, +1\}$

$$g(z) = \begin{cases} 1, & \text{if } z \geq 0. \\ -1, & \text{if } z < 0. \end{cases}$$

$$h_\theta(x) = g(\theta^T x) \quad x \in \mathbb{R}^{n+1} \Big\} \text{ Drop}$$

$x_0 = 1$

$$h_{w,b}(x) = g(w^T x + b) \qquad w \in \mathbb{R}^n . x \in \mathbb{R}^n$$

$w = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}$, $b = \theta_0$