Monday, September 8, 2014   3:19 PM

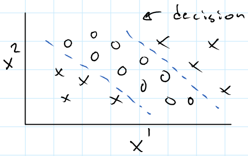
Supervised Learning

Input: X, Y (X ← design matrix; Y ← target)

Output function: $f(x_i) \mapsto y_i$

Evaluation compare: $f(x_i) = \hat{y}_i$ to (real) y_i on new (x_i, y_i)

Notation: x_i is training example 'i' <vector> features
x^i is variable 'i' <scalar> feature



model throw away data points, but retain lines

Location based on individual features x^1, x^2

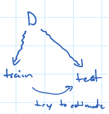
x/o based on class label y: +1 = 'x', or −1 = 'o'

KNN 'closest' − euclidean distance $d(\vec{w}, \vec{v}) = \sqrt{\sum_{i=1}^{n} (w^i - v^i)^2} = \|w - v\|$
− others possible (?) L1 norm?

KNN: 'cost'
- checking $x_i - x_j$ is $O(D)$ (?, D) = distance
- do this N times for each of T test params $O(D \cdot N T)$
- sorting $O(N \log N)$
  selecting $O(KN)$

Possible course project: Improve KNN
- Better distance define or learn distance
- faster classification - gpu, tree, sparse

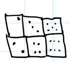
Q: why can we learn?



- Probability

assume $(x_i, y_i) \sim D$ Sampled by some distribution ∴ we learn the distribution

$0 \le P(A) \le 1$ (definitely not ... definitely)



$P(3) = 1/6$

$P(\neg 3) = 5/6$

$\sum_{i=1}^{6} p(i) = 1$

$P(\neg A) = 1 - P(A)$

$p(x) \triangleq p(X = x)$ random variable **X** takes the value of x

$\sum_{x \in X} p(x) = 1$

$P(A, B) = P(A \wedge B)$

$P(A \cup B) = P(A) + P(B) + P(A, B)$

$P(A \mid B) = \dfrac{P(A, B)}{P(B)}$ (for $p(B) > 0$)

$P(A, B) = P(A \mid B) P(B) = P(B \mid A) P(A)$

because sum across $P(i) = 1$

$$\sum_{i=1}^{6} \frac{\prod_{j=1}^{6} \Gamma p(x_j)\, p(i)}{\Gamma\left(\sum_{j=1}^{6} \alpha_j\right)} = \frac{\prod_{j=1}^{6} \Gamma(\alpha_j)}{\Gamma\left(\sum_{j=1}^{6} \alpha_j\right)}$$

$P(3, \text{even}) = 0$

$P(3, \text{odd}) = 1/6$

$P(1 \cup 2)$ $p(1) + p(2) - p(1, 2)$
$1/6 + 1/6 - 0$
$= 1/3$
(intuitive)

$P(3 \mid \text{odd}) = \dfrac{P(3, \text{odd})}{P(\text{odd})} = \dfrac{1/6}{1/2} = 1/3$ (also intuitive)

<u>Bayes Rule</u>

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

$\propto P(B \mid A) P(A)$ ← $p(B)$ is fixed

$P(A) = \sum_b p(A, B = b)$

$p(\text{even}) = \sum_{i=1}^{6} (\text{even}, i) = 0 + 1/6 + \ldots$
$= 1/2$

We can always add a conditioning case.

$$P(A \mid B, C) = \frac{P(B \mid A, C)\, P(A \mid C)}{P(B \mid C)}$$

$X \perp Y \iff P(x, y) = p(x) + p(y)$

Probabilistic Classifiers

discriminative

$p(y_i \mid x_i)$

- no model of x
- lots of labelled data, higher accuracy

<logistic regression>

generative

$p(y_i, x_i)$

- need to model x
- small amount of labeled data

<naïve bayes>

Training

$p(y_i, x_i) = p(x_i \mid y_i)\, p(y_i)$

↑ easy — times we saw +1, ratio times we saw −1

Testing

$$p(y_i \mid x_i) = \frac{p(x_i \mid y_i)\, p(y_i)}{p(x_i)} \propto p(x_i \mid y_i)\, p(y_i)$$

x_i^1	x_i^2	y_i
0	0	0
0	0	1
0	1	0
⋮		

a lot, 2^{D+1} parameters

Naive Bayes

x^j are "mutually independent"

$\forall$ subsets A, B (all) $\quad X^A \perp X^B \mid y$ — each x event is seperate

$$p(x_i^{1:D} \mid y_i) = p(x_i^1 \mid x_i^{2:D}, y_i)\, p(x^{2:D} \mid y_i)$$

$$= p(x_i^1 \mid y_i)\, p(x^{2:D}, y_i)$$

$$= \prod_{j=1}^{p} p(x_i^j \mid y_i)$$ "easy"

x^j	y
0	0
0	1
1	0
1	1

Parametric model. 4D as opposed to 2^{D+1}

Maximum Likelihood

$\operatorname{argmax}_\theta p(\mathbb{D}_{train} \mid \theta)$ ← how we parameterize our data

$\Updownarrow$

$\operatorname{argmax}_\theta \log(\mathbb{D} \mid \theta)$

i.i.d : independent identically distributed

$\Rightarrow \operatorname{argmax}_\theta \prod_{i=1}^{n} p(\mathbb{D}_i \mid \theta)$

$\Rightarrow \operatorname{argmax}_\theta \sum_{i=1}^{n} \log p(\mathbb{D}_i \mid \theta)$

$$\log p(y_i \mid x_i) = \sum_{i=1}^{n} \log p(y_i \mid \theta) + \sum_{j=1}^{D} \log\left(p(x_i^j \mid y_i, \theta_j^{x_i})\right)$$

$(y_i \mid \theta) \sim \text{Bernoulli}(\theta) \quad [0,1]$

$p(y_i \mid \theta) = \theta^{\mathbb{1}y_i=1} (1-\theta)^{\mathbb{1}y_i=0}$

t — f — one or the other

$$\operatorname{argmax}_\theta \sum_{i=1}^{N} \log p(y_i \mid \theta)$$

$$= \sum_{i=1}^{N} \mathbb{1}(y_i = 1) \log\theta + \mathbb{1}(y_i = 0) \log(1-\theta)$$

N_1 — times we saw 1 — N_0 — times we saw 0

derivative (gradient)

$$\nabla = \frac{N_1}{\theta} - \frac{N_0}{1-\theta}$$

find $\nabla = 0$:

$$(1-\theta) N_1 = N_0(\theta) \qquad \frac{\theta}{1-\theta} = \frac{N_1/N}{N_0/N} \qquad N_1 + N_0 = N$$

$$\theta = \frac{N_1}{N}$$

$y_i \in \{1, 2, 3, 4\}$

$\theta^c = \frac{N_c}{N}$

$(x_i \mid y_i, \theta_j^{y_i})$

$\theta_j^c = \frac{N_{jc}}{N_c}$ — feature j and class c / class c