CS182 Discussion 2

Jianguo Huang
huangjg@shanghaitech.edu.cn
ShanghaiTech university
School of Information Science and Technology

March 16, 2023

Presentation Outline

Bayes' rule

Naïve Bayes

Bayes Nets

Perceptron Algorithm

Bayes' rule

Principles for Estimating Probabilities

Statistics

- (maximum likelihood): choose parameters θ that maximize $P(data|\theta)$.
- (maximum a posteriori prob.): $P(\theta|data) = \frac{P(data|\theta)P(\theta)}{p(data)} \infty P(data|\theta)P(\theta)$.

Statistics

- Expected values. $E[X] = \sum_x xP(X = x)$ or $E[X] = \int_x xP(X = x)$
- Covariance. $Cov(X, Y) = E(X - E(X)(Y - E(Y)))$

Naïve Bayes

Conditionally independent

$$P(X_1, \ldots, X_n|Y) = \prod_i^n P(X_i|Y) \tag{1}$$

Definition: X is <u>conditionally independent</u> of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Maximum likelihood estimates:

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_j | Y = y_k) = \frac{\#D\{X_i = x_j \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

MAP estimates (Beta, Dirichlet priors):

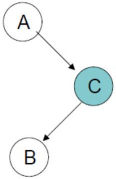
$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + (\beta_k - 1)}{|D| + \sum_m (\beta_m - 1)}$$

Only difference:
"imaginary" examples

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_j | Y = y_k) = \frac{\#D\{X_i = x_j \wedge Y = y_k\} + (\beta_k - 1)}{\#D\{Y = y_k\} + \sum_m (\beta_m - 1)}$$

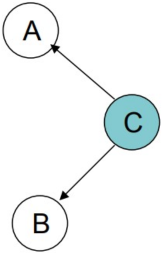
Bayes Nets

Head to Tail



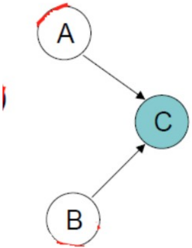
A cond indep of B given C.

Tail to Tail



A cond indep of B given C.

Head to Head



A is not cond indep of B given C.

Perceptron Algorithm

$$sign(w^T x)$$

$$\min_{w} \quad L(w) = -\sum_{i \in M} y_i (w^T x)$$

Algorithm:

- Set t=1, start with all-zeroes weight vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$.
  - On a mistake, update as follows:
    - Mistake on positive, update $w_{t+1} \leftarrow w_t + x$
    - Mistake on negative, update $w_{t+1} \leftarrow w_t - x$

Dual Algorithm

Easy to kernelize since w_t is weighted sum of incorrectly classified examples $w_t = a_{i_1} x_{i_1} + \cdots + a_{i_k} x_{i_k}$

Replace $w_t \cdot x = a_{i_1} x_{i_1} \cdot x + \cdots + a_{i_k} x_{i_k} \cdot x$ with

$a_{i_1} K(x_{i_1}, x) + \cdots + a_{i_k} K(x_{i_k}, x)$

if data not linearly separable

Kernelizing the Perceptron Algorithm

- Given x, predict + iff

$\phi(x_{i_{t-1}}) \cdot \phi(x)$

$$a_{i_1} K(x_{i_1}, x) + \cdots + a_{i_{t-1}} K(x_{i_{t-1}}, x) \geq 0$$

- On the t th mistake, update as follows:
  - Mistake on positive, set $a_{i_t} \leftarrow 1$; store x_{i_t}
  - Mistake on negative, $a_{i_t} \leftarrow -1$; store x_{i_t}