

Euclidean distance

$$D(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

Manhattan distance

$$D(A, B) = |a_1 - b_1| + |a_2 - b_2| + \dots + |a_n - b_n|$$

SMC

$$\frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

Normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Standardization

$$x' = \frac{x - \mu(x)}{\sigma(x)}$$

Jaccard coefficient

$$\frac{f_{11}}{f_{01} + f_{10} + f_{11}} \quad \# \text{ 没有 } f_{00}$$

Cos similarity

$$A \cdot B$$

$$\frac{A \cdot B}{\|A\| \|B\|}$$

1 high related

0 not related

Correlation

$$\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \times \sqrt{\sum (y_i - \bar{y})^2}}$$

Ridge Regression

— linear regression

$$f(w) = \min ||Xw - y_h|| + \lambda ||w||^2$$

$$w = (X^T X + \lambda I)^{-1} X^T y$$

Gradient descent

$$w_{i+1} = w_i - \alpha_i \nabla f(w)$$

Logistic regression

$$\begin{matrix} O(nk^2 + k^2) & O(nk) & O(k) & O(bk) \\ O(nk + k^2) & O(k) & & \end{matrix}$$

$$\sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \quad \left\{ \begin{array}{l} > 0.5 \Rightarrow \hat{y} = 1 \\ < 0.5 \Rightarrow \hat{y} = 0 \end{array} \right.$$

Naive Bayes

$$P(H|E) = \frac{P(E|H) P(H)}{P(E)}$$

→ Prior Probability (before)

posteriori probability

$$P(H|E_i) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(E_i - \mu)^2}{2\sigma^2}}$$

or
conditional probability

$$\text{Variance} = \sqrt{\frac{\sum_{i=1}^n (E_i - \mu)^2}{n-1}}$$

(after)

Confusion Matrix

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + F_p + T_n + F_n}$$

$$\text{Precision} = \frac{T_p}{T_p + F_p}$$

$$R = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2PR}{P + R}$$

Information Gain

$$\log_2 0 = 0$$

$$H(S) = - \sum_i P_i \log_2 P_i$$

$$\text{Gain} = H(S) - T_2$$

$$\rightarrow W_1 H(S_1) + \dots + W_n H(S_n)$$

ada boosting

$$\epsilon_t = \sum W_t(n) \mathbb{I}[y_n \neq h_t(x_n)]$$

$$\beta_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}$$

$$W_{t+1} = W_t(n) e^{-\beta_t y_n h_t(x_n)} \quad \xrightarrow{\text{True label}}$$

$$h(x)^* = \text{Sign} \left[\sum \beta_t h_t(x) \right]$$

SVM

$$W_k = \sum_{i=1}^N \lambda_i y_i x_i$$

linear

$\hookrightarrow k(x_i, z)$ kernel trick for non linear dot product problem

$$\frac{1}{2} \|w\|^2 \rightarrow \text{maximize margin}$$

• Training: Dot product of pair of training vectors

Classify: Dot product of the new vector z and the support vector

$$f = w \cdot z + b = \sum \lambda_i y_i x_i \cdot z + b$$

\uparrow w 求出后直接计算

SVD

$$U_{n \times m} \Lambda_{m \times m} V_{m \times k} \Rightarrow U_{n \times m} \Lambda_{m \times m} V_{m \times k}$$

Compression ratio

$$r = \frac{k(1+m+n)}{n \times m}$$

Perceptron

$$W^{new} = W^{old} + e x^T$$

$$b^{new} = b^{old} + e \cancel{x}$$

\rightarrow True output t - actual output a

back propagation

$$W_{pq}^{new} = W_{pq}^{old} + \Delta W_{pq}$$

momentum:
current update depend on
the previous weight

$$W_{pq} = \eta \cdot \delta_q \cancel{O_p}$$

\rightarrow error of neuron q

$$\textcircled{1} \delta_q = (t_q - o_q) f'(z_q)$$

$$\textcircled{2} \delta_q = f'(z_q) \sum_i W_{qi} \delta_i$$

$$\cdot \text{bias}_q^{\text{new}} = \text{bias}_q^{\text{old}} + \eta \times \delta_q$$

time based

exponential

$$\eta_n = \frac{\eta_{n-1}}{1 + d \times n}$$

$$\eta_n = \eta_0 e^{-dn}$$

Xavier

$$\sigma = \sqrt{\frac{2}{N_{in} + N_{out}}}$$

Soft Max

$$\frac{e^{o_i}}{\sum e^{o_i}}$$

RNN

$$h_t = f_w(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

$$y_t = W_{hy}h_t + b_y$$

LSTM



0 close gate
1 open gate

$$\sigma \text{ or } \tanh \left(W_z \cdot [h_{t-1}, x_t] + b_z \right)$$

Transformers

$$O(n^2)$$

$$Z = \text{softmax} \left(\frac{Q \times K^T}{\sqrt{d_k}} \right) V$$

$$W^O \times Z$$

GMM

$\sum w_j = 1$ 每个 distri 的 weight
 \nwarrow \nearrow pdf

$$P(\text{cluster } i \text{ distri} \mid x_i, \theta_i) = \frac{w_i \times P(x_i \mid \theta_i)}{\sum w_j P(x_j \mid \theta_j)}$$

\rightarrow 有 k 个 distri 就 k 个 θ
 \downarrow 不变

Hierarchical clustering

$$O(n^3)$$

$$O(n^2)$$

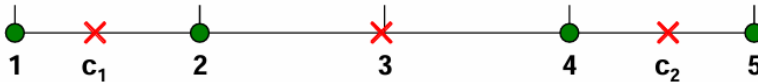
SSE BSE

\hookrightarrow cohesion \hookrightarrow separation

- We can use squared distances to express cohesion and separation:

- SSE = distance within a cluster $SSE = \sum_{i=1}^k \sum_{\mathbf{x} \in K_i} (c_i, \mathbf{x})^2$

- BSE = distance between clusters $BSE = \sum_{i=1}^k |K_i| (c_i, c)^2$



$k=2$ clusters

$(\{1, 2\}, \{3, 4\})$

$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSE = 2 \cdot (3-1.5)^2 + 2 \cdot (4.5-3)^2 = 9$$

Silhouette

$$s_i = \frac{b_i - a_i}{\max(b_i, a_i)}$$

$$k_i = \text{avg}(s_i, \dots, s_n)$$

$$\text{all} = \text{avg}(k_1, \dots, k_k)$$

HMM | Forward $O(N^2T)$

$$f_k(i) = e_k(x_i) A_o(k)$$

$$f_k(i) = e_k(x_i) \sum_j^{\text{\# of label}} f_j(i-1) a_{j \rightarrow k}$$

$$P(x) = \sum_{\hat{y}}^{\text{\# of label}} f_{\hat{y}}(m-1)$$

HMM2 - Viterbi: $O(N^2)$

$$V_k(1) = e_k(x_1) A_{0 \rightarrow k}$$

$$\begin{cases} V_k(i) = e_k(x_i) \max_j \{ f_j(i-1) A_{j \rightarrow k} \} \text{ for all } j \in \text{label} \\ P_{tr}_k(i) = \arg \max_j \{ \checkmark \} \end{cases}$$

$$\begin{aligned} \pi_m &= \arg \max_j \{ f_j(m) \} \\ \text{max } i &= m \\ \pi_{i-1} &= P_{tr}_k(i) \quad i = m, \dots, 2 \end{aligned}$$

State-Value Func

$$V^{\pi}(s) = E \left[\sum r^t r_t \mid s_0 = s, \bar{\pi} \right]$$

Action-Value Function

$$Q(s, k) = E[\sum r^t r_t \mid s_0 = s, a_0 = a, \pi]$$

Bellman Equation

$Q^{\star}(s, k)$ = immediate reward + next state cumulative reward

$$= E[\bar{r} + r_{\max} Q^{\star}(s', a') \mid s, a]$$

Q-learning

$$\Delta Q(s, a) = \underbrace{R(s, a, s') + \max_{a'} Q(s', a')}_{\text{target}} - \underbrace{Q(s, a)}_{\text{current}}$$

Deep Q-learning

$$L = [r + r_{\max_a} Q(s', a') - Q(s, a)]^2$$

avg link

$$d(A, B) = \frac{1}{m \cdot n} \sum_{x \in A} \sum_{y \in B} \text{dist}(x, y)$$