Euclidean distance

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

Manhattan distance

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

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

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

Cos similarity

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

1 high related

0 not related

Correlation

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

Ridge Regression

— linear regression

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

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

Gradient descent

$$W_{i+1} = W_i - \alpha_i \nabla_s f(w)$$

Logistic regression

$O(nk^3+k^2)$ $O(nk)$ $O(k)$ $O(bk)$

$O(nk+k^2)$ $O(k)$

$$\sigma(w^T x + b) = \frac{1}{1 + e^{-\underbrace{(w^T x + b)}}} \rightarrow \begin{cases} > 0.5 \Rightarrow \hat{y} = 1 \\ < 0.5 \Rightarrow \hat{y} = 0 \end{cases}$$

Naive Bayes

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

→ Prior Probability (before)

posteriori probability

or

Conditional probability

(after)

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

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

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

$\log_2 0 = 0$

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

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

$$\hookrightarrow W_{S_1} H(S_1) + \cdots W_{S_n} H(S_n)$$

ada boosting

$$\varepsilon_t = \sum W_t(n)\, I[y_n \neq h_t(x_n)]$$

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

True label

$$W_{t+1} = W_t(n)\, e^{-\beta_t y_n h_t(x_n)}$$

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

SVM

- $W_k = \sum_{i=1}^{N} \lambda_i y_i x_{i,k}$ linear

  dot product problem

  ↳ $K(x_i, z)$ kernel trick for non linear

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

- Training: Dot product of pair of training vectors

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

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

  w求出后再直接计算

SVD

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

$\boxed{\text{compression ratio}}$ $r = \dfrac{k(1+m+n)}{n \times m}$

$\boxed{\text{Perceptron}}$

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

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

$\hookrightarrow$ True output t − actual output a

$\boxed{\text{back propagation}}$

momentum:
current update depend on the previous weight

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

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

error of neuron q

① $\delta_q = (t_q - O_q) f'(z_q)$

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

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

time based

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

exponential

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

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

$\otimes$

0 close gate

1 open gate

$$\begin{matrix} \sigma \\ \text{or} \\ \tanh \end{matrix} \left(W_? \cdot [h_{t-1}, x_t] + b_? \right)$$

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

pdf

$$P(\text{cluser} \mid \text{distri} \mid x_i, \theta_i) = \frac{W_i \times P(x_i \mid \theta_i)}{\sum W_j \, P(x_i \mid \theta_j)}$$

有几个 distri 就加几次

不变

Hierarchical clustering

$O(n^3)$

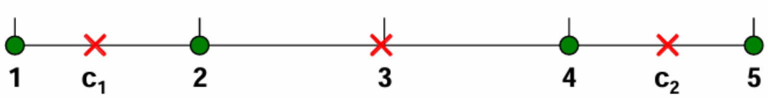
$O(n^2)$

SSE BSE

cohesion

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 = $2^*(3-1.5)^2 + 2^*(4.5-3)^2 = 9$

Silhouette

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

$$k_i = \text{avg}(s_1, \cdots, s_n)$$

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

HMM 1 — Forward $O(N^2 T)$

$$f_k(1) = e_k(x_1) A_0(k)$$

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

$$P(x) = \sum_{j}^{\#\text{ of label}} f_j(m-1)$$

HMM2 - Viterbi: $O(NT)$

$$V_k(1) = e_k(x_1)\, A_0(k)$$

$$\begin{cases} V_k(i) = e_k(x_i)\ \text{Max}\{ V_j(i-1)\, a_{j \to k} & \text{for all } j \in \text{label} \\ \text{Ptr}_k(i) = \text{argmax}\{\ \ \} \end{cases}$$

$$\pi_m = \text{argmax}\{ V_j(m) \}$$

maxi = m

$$\pi_{i-1} = \text{Ptr}_k(i) \qquad i = m, \cdots, 2$$

State-value Func

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

Action-Value Function

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

Bellman Equation

$Q^{*}(s,k)$ = immediate reward + next state cummulative reward

$$= E\left[r + \gamma \max Q^{*}(s',a') \mid s,a\right]$$

Q-learning

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

Deep Q-learning

$$L = \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]^2$$