

CS 529 - Midterm Notes

① $\{ \text{Eng.}, \text{French} \}^n$

machine translation \rightarrow RNN

$$P(w_t | w_1, w_2, \dots, w_{t-1}) = ?$$

②



Image \rightarrow caption

x

y

Encoder - Decoder Arch.

e.g.: CNN + RNN

KNN - $O(knd)$

d = dimensionality

kmeans - non-convex

DT vs LR - opp is probability
($\geq LR$)

LINEAR REGRESSION

$$L = \frac{1}{2} \sum_i (w_i x_i - y)^2 + \lambda w^2$$

Huber loss: $L = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & |y - \hat{y}| \leq s \\ s(|y - \hat{y}| - \frac{1}{2}s), & \text{otherwise} \end{cases}$

Matrix Form

$$L = \frac{1}{2} \|Xw - y\|_2^2$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = (X^T X)^{-1} X^T y$$

gradient

$$w = w - \alpha \left((w_i x_i - y) x_i + 2\lambda w \right)$$

DECISION TREES

$$H(Y) = - \sum p_i \log p_i$$

$$IG(x) = H(Y) - H(x)$$

$$= H(Y) - \left[\frac{x_1}{t} H(Y|x_1) + \frac{x_2}{t} H(Y|x_2) \right]$$

$$(t = \text{total at root}) + \dots \]$$

NAIVE BAYES

$\mathbf{x} = [x_1, x_2, \dots]$

$$P(y|x) = \text{argmax}_y P(y) \prod_i P(x_i|y)$$

$$\mu = \frac{1}{n} \sum_i x_i$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_i (x_i - \mu)^2}$$

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

LOGISTIC REGRESSION

$$L = \frac{1}{m} \sum_i \text{cost}(h(x_i), y_i)$$

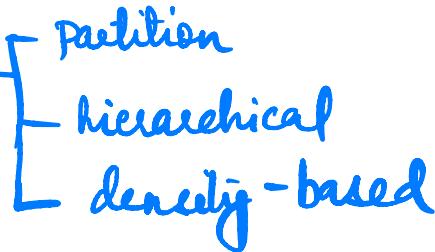
$$\text{cost}(h(\mathbf{x}), y) = \begin{cases} -\log(h(\mathbf{x})), & y=0 \\ -\log(1-h(\mathbf{x})), & y=1 \end{cases}$$

$$L = -\frac{1}{m} \sum_i (y_i \log(h(x_i)) + (1-y_i) \log(1-h(x_i)))$$

$$w = w + \alpha ((y - h(x)) \mathbf{x})$$

k-means

clustering


 partition
 hierarchical
 density-based

Issues :

- locally optimized clusters
- empty clusters
- data distribution (density-based)
 - kernelized k-means

k-means++

any given time, let $D(x)$ denote the shortest distance from a data point x to the closest center we have already chosen. Then, we define the following algorithm, which we call **k-means++**.

- 1a. Choose an initial center c_1 uniformly at random from \mathcal{X} .
- 1b. Choose the next center c_i , selecting $c_i = x' \in \mathcal{X}$ with probability $\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
- 1c. Repeat Step 1b until we have chosen a total of k centers.
- 2-4. Proceed as with the standard **k-means** algorithm.

We call the weighting used in Step 1b simply “ D^2 weighting”.

3 k-means++ is $O(\log k)$ -competitive

Hierarchical Clustering

- └ Agglomerative (bottom-up)
- └ Divisive (top-down)

Dendrogram

Distance betn clusters

- Single linkage (min)
- Complete linkage (max)
- Average

PERCEPTRON

$$w_i = w_i + \alpha y_i x_i \quad \text{if } \text{sign}(w \cdot x) \neq y$$

NEURAL NETWORKS

$$a_i(z) = b_i + W_i h_{i-1}(z)$$

$$h_i(z) = g(a_i(z))$$

SECURITY