CS 529 - Midterm Notes

① $\{Eng., French\}^N$

machine translation → RNN

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

②  Image → caption

x → y

Encoder - Decoder Arch.

eg: CNN + RNN

KNN — $O(knd)$

↳ d = dimensionality

Kmeans — non-convex

DT vs LR — op is probability (>LR)

LINEAR REGRESSION

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

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

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

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

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

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

(t = total at root)

NAIVE BAYES

$x = [x_1, x_2 \cdots]$

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

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

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

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

LOGISTIC REGRESSION

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

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

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

$$w = w + \alpha \left((y - h(x)) x \right)$$

k-means

Clustering
- Partition
- Hierarchical
- density-based

Issues:
- locally optimized clusters
- empty clusters
- data distribution (density-based)
  → kernalized 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) ↓ Dendrogram
- Divisive (top-down)

Distance betⁿ clusters

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

PERCEPTRON

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

NEURAL NETWORKS

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

$h_i(x) = g(a_i(x))$

SECURITY