COMP551 Notes

Chany Ahn

1 KNN

1.1 Real-Valued Feature-Vector Distance Metrics

- Euclidean Distance

$$D_{Euclid}(x, x') = \sqrt{\sum_{d=1}^{D}(x_d - x'_d)^2}$$

- Manhattan Distance

$$D_{Manhattan}(x, x') = \sum_{d=1}^{D}|x_d - x'_d|$$

- Minkowski distance

$$D_{Minkowski}(x, x') = \left(\sum_{d=1}^{D}|x_d - x'_d|^p\right)^{\frac{1}{p}}$$

- Cosine similarity

$$D_{Cosine}(x, x') = \frac{x^\top x'}{||x||||x'||}$$

1.2 Discrete Feature-Vector Distance Metrics

- Hamming Distance

$$D_{Hamming}(x, x') = \sum_{d=1}^{D}\mathbb{I}(x_d \neq x'_d)$$

1.3 Label by Majority

Estimate the probability that an input should be classified by a given class:

$$p(y^{new} = c|x^{new}) = \frac{1}{K}\sum_{x^{(k)} \in KNN(x^{new})}\mathbb{I}(y^{(k)} = c)$$

2 Decision Trees

3 Important Concepts

3.1 Evaluation Metrics

For binary classifiers, we have these metrics based on the confusion table:

$$Accuracy = \frac{TP + TN}{P + N} \tag{1}$$

$$Error\ rate = \frac{FP + FN}{P + N} \tag{2}$$

$$Precision = \frac{TP}{RP} \tag{3}$$

$$Recall = \frac{TP}{P} \tag{4}$$

$$F_1 score = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

4 Linear Regression

4.1 Linear Model

Assuming a scalar output, $f_w : \mathbb{R}^D \to \mathbb{R}$ where:

$$f_w(\vec{x}) = w_0 + w_1 x_1 + \ldots + w_D x_D \tag{6}$$

where w is the model parameters. A better generalization is letting $\vec{x}^\top = [1, x_1, \ldots, x_D]$ such that $f_w(\vec{x}) = w^\top x$.

4.2 Loss

To fit the data, must minimize a loss function.

4.2.1 L2 Loss

Loss for a single instance in the data.

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \tag{7}$$

4.2.2 Cost Function

Sum of squared errors (over all instances).

$$J(w) = \frac{1}{2} \sum_{n=1}^{N} \left(y^{(n)} - w^\top x \right) \tag{8}$$

4.3 Linear Least Squares

$$w^* = min_w \sum_n \left(y^{(n)} - w^\top x^{(n)} \right)^2 \tag{9}$$

4.4 Matrix Form

$$\hat{y} = Xw \tag{10}$$

where X is $N \times D$, $\hat{y}$ is $N \times 1$, and w is $D \times 1$.

4.4.1 Linear Least Squares: Matrix Form

$$\text{argmin}_w \frac{1}{2}||y - Xw||_2^2 = \frac{1}{2}(y - Xw)^\top (y - Xw) \tag{11}$$

4.5 Optimal w: $D = 1$ Case

$$w^* = \frac{\sum_n x^{(n)} y^{(n)}}{\sum_n x^{(n)2}} \tag{12}$$

4.6 Optimal w: Any D

$$\sum_n (w^\top x^{(n)} - y^{(n)}) x_d^{(n)} = 0 \;\; \forall d \in \{1, \ldots, D\} \tag{13}$$

4.7 Normal Equation

$$X^\top (y - Xw) = 0 \tag{14}$$

4.7.1 Closed Form Solution

$$w^* = (X^\top X)^{-1} X^\top y \tag{15}$$

$$\hat{y} = Xw = X(X^\top X)^{-1} X^\top y \tag{16}$$

where (16) is the projection into column space of X.

4.8 Miltiple Targets

Instead of $y \in \mathbb{R}^N$, we have $Y \in \mathbb{R}^{N \times D'}$. Then we have

$$\hat{Y} = XW \tag{17}$$

where W is $D \times D'$. W^* is found by

$$W^* = (X^\top X)^{-1} X^\top Y \tag{18}$$

4.9 Nonlinear Basis Functions

Now denote the features by $\phi_d(x), \forall d$. So, the linear regression problem becomes $f_w = \sum_d w_d \phi_d(x)$. Thus, the solution becomes

$$(\phi^\top \phi) w^* = \phi^\top y \tag{19}$$

4.9.1 Nonlinear Basis Functions

- Polynomial bases

$$\phi_k(x) = x^k$$

- Gaussian bases

$$\phi_k(x) = e^{-\frac{(x-\mu_k)^2}{s^2}}$$

- Sigmoid bases

$$\phi_k(x) = \frac{1}{1 + e^{-\frac{x-\mu_k}{s}}}$$

5 Logistic Regression

5.1 Squashing Function

$$w^\top x \to \sigma(w^\top x)$$

The desirable properties of this function $\sigma : \mathbb{R} \to \mathbb{R}$:

- all $w^\top x > 0$ are squashed close together.
- all $w^\top x < 0$ are squashed close together.

5.2 Logistic Function

$$\sigma(z) = \frac{1}{1 + e^{-x}} \tag{20}$$

where the decision boundary is

$$w^\top x = 0 \iff \sigma(w^\top x) = \frac{1}{2}. \tag{21}$$

This interprets the prediction as a class probability

$$\hat{y} = p_w(y = 1|x) = \sigma(w^\top x) \tag{22}$$

where the log-ratio of class probabilities is linear

$$\log \frac{\hat{y}}{1 - \hat{y}} = \log \frac{\sigma(w^\top x)}{1 - \sigma(w^\top x)} = \log \frac{1}{e^{-w^\top x}} = w^\top x \tag{23}$$

5.3 The Loss

- Misclassification Error

$$L_{0/1}(\hat{y}, y) = \mathbb{I}\left(y \neq \text{sign}\left(\hat{y} - \frac{1}{2}\right)\right)$$

- L2 Loss (see (7))
- Cross-Entropy Loss (Loss considered for Logistic Regression =))

$$L_{CE}(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

5.4 Cost Function

$$J(w) = \sum_{n=1}^{N} y^{(n)} \log(1 + e^{-w^\top x}) + (1 - y^{(n)}) \log(1 + e^{w^\top x}) \tag{24}$$

6 Maximum Likelihood

6.1 Likelihood

$$\mathcal{L}(\theta;\mathcal{D}) = \prod_{x\in\mathcal{D}} p(x;\theta) \tag{25}$$

Using the product creates extreme values.

6.2 Log-Likelihood

$$l(\theta;\mathcal{D}) = \log(\mathcal{L}(\theta;\mathcal{D})) = \sum_{x\in\mathcal{D}} \log(p(x;\theta)) \tag{26}$$

Has the same behaviour, but log-likelihood is well-behaved. To find $\theta^* = \arg\max_\theta l(\theta;\mathcal{D})$, must solve for $\frac{\partial}{\partial\theta} l(\theta;\mathcal{D}) = 0$ (if there is an analytical solution).

$\theta^{MLE} = \arg\max_\theta p(\mathcal{D}|\theta)$.

6.3 Categorical Distributions

- Likelihood: $p(\mathcal{D}|\theta) = \prod_{x\in\mathcal{D}} Cat(x|\theta) = \prod_{x\in\mathcal{D}} \prod_k \theta_k^{\mathbb{I}(x=k)}$
- Log-Likelihood: $l(\theta,\mathcal{D}) = \sum_{x\in\mathcal{D}} \sum_k \mathbb{I}(x=k)\log(\theta_k)$

$\theta_k^{MLE} = \frac{N_k}{N}$.

6.4 Bayesian Approach

Max-likelihood does not reflect uncertainty. The Bayesian approach is as follows:

- We maintain a distribution over the parameters: $p(\theta)$.
- After observing $\mathcal{D}$, update the distribution given $\mathcal{D}$: $p(\theta|\mathcal{D})$.

Use Baye's Theorem:

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \tag{27}$$

where $p(\theta) = \int p(\theta)p(\mathcal{D}|\theta)d\theta$.

6.5 Maximum a Posteriori (MAP)

Use the parameter with the highest posterior probability:

$$\theta^{MAP} = \arg\max_\theta p(\theta|\mathcal{D}) = \arg\max_\theta p(\theta)p(\mathcal{D}|\theta) \tag{28}$$

6.6 Maximum Likelihood in ML

Consider linear regression and logistic regression. The learning that ML algos involve finding w that maximizes the likelihood of the training data (many algorithms assume some probabilistic model).

$$w^* = \arg\max_w \sum_n \log p(y^{(n)}|x^{(n)};w) \tag{29}$$

6.6.1 Probabilistic Interpretation of Linear Regression

Assume $p(y|x;w)$ with following form:

$$p_w(y|x) = \mathcal{N}(y|w^\top x, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-w^\top x)^2}{w\sigma^2}} \tag{30}$$

The likelihood is as follows:

- $\mathcal{L}(w) = \prod_{n=1}^{N} p(y^{(n)}|x^{(n)};w)$
- $l(w) = \sum_n -\frac{1}{2\sigma^2}(y^{(n)} - w^\top x^{(n)})^2 + \text{constants}$.
- Max-Likelihood Parameters: $w^* = \arg\max_w l(w) = \arg\min_w \frac{1}{2}\sum_n (y^{(n)} - w^\top x^{(n)})^2$ (linear least squares!).

6.6.2 Probabilistic View of Logistic Regression

Interpret the prediction as class probability: $\hat{y} = p(y = 1|x;w) = \sigma(w^\top x)$, so we have a Bernoulli likelihood:

$$p(y^{(n)}|x^{(n)};w) = \text{Bernoulli}(y^{(n)}; \sigma(w^\top x^{(n)})) = \hat{y}^{(n)^{y^{(n)}}}(1-\hat{y}^{(n)})^{1-y^{(n)}} \tag{31}$$

The w that maximizes the log likelihood:

$$\begin{aligned} w^* &= \max \sum_{n=1}^{N} \log p(y^{(n)}|x^{(n)};w) && (32)\\ &= \max_w \sum_{n=1}^{N} y^{(n)} \log(\hat{y}^{(n)}) + (1-y^{(n)})\log(1-\hat{y}^{(n)}) && (33)\\ &= \min_w J(w) && (34) \end{aligned}$$

The last equality is the cross entropy cost function, thus cross entropy loss maximizes the conditional likelihood in logistic regression!

6.6.3 Multiclass Classification for Logistic Regression

If we have C classes (rather than a binary classification), we use the categorical likelihood:

$$\text{Cat}(y|\hat{y}) = \prod_{c=1}^{C} \hat{y}_c^{\mathbb{I}(y=c)} \tag{35}$$

The **softmax** takes a vector of real numbers and produces probabilites:

$$\hat{y}_c = \text{softmax}(z)_c = \frac{e^{z_c}}{\sum_{c'=1}^{C} e^{z_{c'}}} \tag{36}$$

so $\sum_c \hat{y}_c = 1$. If we produce the input to softmax using a linear model:

$$\hat{y}_c = \text{softmax}([w_1^\top x, \ldots, w_C^\top x])_c = \frac{e^{w_c^\top x}}{\sum_{c'} e^{w_{c'}^\top x}} \tag{37}$$

We can put these vectors as the *columns of the weight matrix* W.

$$\hat{y} = \text{softmax}(Wx) = \frac{e^{Wx}}{\mathbf{1}^\top e^{Wx}} \tag{38}$$

where $dim(W) = C \times D$, $dim(x) = D \times 1$, and $dim(\hat{y}) = C \times 1$. This produces a vector of class probabilities $\hat{y}$ for each input x.

6.7 Cost Function

The likelihood of the data as a function of model parameters:

$$\mathcal{L}(\{w_c\}) = \prod_{n=1}^{N} \text{softmax}(Wx^{(n)})^\top y^{(n)} \tag{39}$$

$y^{(n)}$ is one-hot encoded.
Softmax cross entropy cost function is the negative of the log-likelihood:

$$J(\{w_c\}) = -\left(\sum_{n=1}^{N} (Wx^{(n)})^\top y^{(n)} - \log \sum_c e^{w_c x^{(n)}}\right) \tag{40}$$

The naive implementation of log-sum-exp cause over/underflow.

$$\log \sum_c e^{z_c} = \overline{z} + \log \sum_c e^{z_c - \overline{z}} \tag{41}$$

where $\overline{z} \leftarrow \max_c z_c$.

7 Gradient Descent Methods

$$\nabla J(w) = \left[\frac{\partial}{\partial w_1} J(w), \ldots, \frac{\partial}{\partial w_D} J(w)\right] \tag{42}$$

7.1 Iterative Algorithm

- Starts from some $w^{\{0\}}$
- update using gradient: $w^{\{t+1\}} \leftarrow w^{\{t\}} - \alpha \nabla J(w^{\{t\}})$

This converges to a local minima.

7.2 Convex Function