CS229-Cheatsheet

Supervised Learning

- **Gradient Descent:** to minimize $J(\theta)$, we perform $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta)$
- $\nabla_A AB = B^T$, $\nabla_{A^T} f(A) = (\nabla_A f(A))^T$, $\nabla_A trABA^T C = CAB + C^T AB^T$, $\nabla_A |A| = |A|(A^{-1})^T$
- **Normal Equations and Least Squares** $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)^2 \to \nabla_\theta J(\theta) =$ $\nabla_\theta \frac{1}{2}(X\theta - y)^T(X\theta - y) = X^T X\theta - X^T y = 0 \to$ $X^T X\theta = X^T y \to \theta = (X^T X)^{-1} X^T y.$
- **Locally Weighted Regression** Fit θ to minimize $\sum_{i=0}^{m}(y^i - \theta^T x^i)^2$ where $w^i = e^{-\frac{(x^i - x)^2}{2\tau^2}}$
- **Logistic Regression:** $h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$, $g(z) = \frac{1}{1+e^{-z}}$, $g'(z) = \frac{d}{dz}\frac{1}{1+e^{-z}} = g(z)(1 - g(z))$, $p(y|x;\theta) = (h_\theta(x))^y(1 - h_\theta(x))^{1-y}$. $l(\theta) = logL(\theta) = \sum_{i=1}^{m} y^i logh(x^i) + (1 - y^i)log(1 - h(x^i))$, $\frac{\partial}{\partial \theta_j} l(\theta) = (y - h_\theta(x))x_j$
- **Perceptron Learning Algorithm** $\theta_j := \theta_j + \alpha(y^i - h_\theta(x^i))x_j^i$
- **Newton's Method:** $\theta := \theta - \frac{f(\theta)}{f'(\theta)}$, we want the first derivative to be zero, then $\theta := \theta - \frac{l'(\theta)}{l''(\theta)}$, if θ is a vector then $\theta := \theta - H^{-1}\nabla_\theta l(\theta)$ where $H_{ij} = \frac{\partial^2 l(\theta}{\partial\theta_i \partial\theta_j}$
- **Exponential Family** $p(y;\eta) = b(y)exp(\eta^T T(y) - a(\eta))$
- **General Linear Model Assumptions:** 1. $y|x;\eta \sim ExponentialFamily(\eta)$. 2. Given x our goal is to predict the expected value of $T(y)$ which is usually just y, so we would like our hypothesis to satisfy $h(x) = E(y|x)$. 3. The natural parameter η and inputs x are related linearly. $\eta = \theta^T x$.
- **Canonical response function:** the distribution's mean as a function of the natural parameter $g(\eta) = E(T(y);\eta)$.

Generative Learning Algorithm

Gaussian Discriminant Analysis

- $y \sim Bernoulli(\phi), x|y = 0 \sim N(\mu_0, \Sigma), x|y = 1 \sim N(\mu_1, \Sigma)$.
- $p(y) = \phi^y(1 - \phi)^{1-y}$
- $p(x|y = 0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x - \mu_0)^T\Sigma^{-1}(x - \mu_0))$
- $p(x|y = 1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x - \mu_1)^T\Sigma^{-1}(x - \mu_1))$
- $l(\phi, \mu_0, \mu_1, \Sigma) = log \prod_{i=1}^{m} p(x^i, y^i; \phi, \mu_0, \mu_1, \Sigma) =$ $log \prod_{i=1}^{m} p(x^i|y^i; \phi, \mu_0, \mu_1, \Sigma)p(y^i, \phi)$.
- By maximizing l with respect to the parameters, we find the maximum likelihood of the parameters to be: $\phi = \frac{1}{m}1\{y^i = 1\}$ $\mu_0 = \frac{\sum_{i=1}^m \{y^i = 0\}x^i}{\sum_{i=1}^m \{y^i = 0\}}$ $\mu_1 = \frac{\sum_{i=1}^m \{y^i = 1\}x^i}{\sum_{i=1}^m \{y^i = 1\}}$ $\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu_{y^i})^T(x^i - \mu_{y^i})$

Naive Bayes

- **Naive Assumption:** $p(x_1, x_2, \ldots |y) = p(x_1|y)p(x_2|y) \ldots = \prod_{i=1}^{n} p(x_i|y)$