

Probability

Bayes Theorem:

$$P(Y = \pm 1|X) = \frac{P(X|Y=\pm 1)P(Y=\pm 1)}{P(X|Y=+1)P(Y=+1)+P(X|Y=-1)P(Y=-1)}$$

X and **Y** are **independent** iff $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X})P(\mathbf{Y})$
X and **Y** are **uncorrelated** iff $\mathbb{E}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}(\mathbf{X})\mathbb{E}(\mathbf{Y})$

Matrix calculus

$$f(\mathbf{x}) = \mathbf{Ax} + \mathbf{x}'\mathbf{A} + \mathbf{x}'\mathbf{x} + \mathbf{x}'\mathbf{Ax} \Rightarrow \frac{df(\mathbf{x})}{d\mathbf{x}} = \mathbf{A}' + \mathbf{A} + 2\mathbf{x} + \mathbf{Ax} + \mathbf{A}'\mathbf{x}$$
$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}; \nabla_x(a\mathbf{x}) = a\mathbf{I}; \mathbf{J} = |\frac{\partial \mathbf{x}}{\partial \mathbf{y}}| \Leftrightarrow \mathbf{J}^{-1} = |\frac{\partial \mathbf{y}}{\partial \mathbf{x}}|$$

Perceptron

$$f(\mathbf{x}) = \boldsymbol{\theta} \cdot \mathbf{x} + \theta_0 = \sum_{i=1}^d \theta_i x_i + \theta_0, \hat{y} = \begin{cases} 1, & \text{if } f(x) \geq 0 \\ -1, & \text{if } f(x) < 0 \end{cases}$$

Decision boundary, a **hyperplane** in \mathbb{R}^d :
 $H = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) = 0\} = \{\mathbf{x} \in \mathbb{R}^d : \boldsymbol{\theta} \cdot \mathbf{x} + \theta_0 = 0\}$

θ is the **normal** of the hyperplane,
θ₀ is the **offset** of the hyperplane from origin,
 $-\frac{\theta_0}{\|\boldsymbol{\theta}\|}$ is the **signed distance** from the origin to hyperplane.

Perceptron algorithm,
Input: $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$
while some $y_i \neq \text{sign}(\boldsymbol{\theta} \cdot \mathbf{x}_i)$
 pick some misclassified (\mathbf{x}_i, y_i)
 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y_i \mathbf{x}_i$

Given a **linearly separable data**, perceptron algorithm will take no more than $\frac{R^2}{\gamma^2}$ updates to **converge**, where $R = \max_i \|\mathbf{x}_i\|$ is the radius of the data, $\gamma = \min_i \frac{y_i(\boldsymbol{\theta} \cdot \mathbf{x}_i)}{\|\boldsymbol{\theta}\|}$ is the margin.
Also, $\frac{\boldsymbol{\theta} \cdot \mathbf{x}}{\|\boldsymbol{\theta}\|}$ is the signed distance from H to **x** in the direction **θ**.

θ = $\sum_i \alpha_i y_i \mathbf{x}_i$, thus any inner product space will work, this is a **kernel**.

Gradient descent view of perceptron, minimize margin cost function $J(\boldsymbol{\theta}) = \sum_i (-y_i(\boldsymbol{\theta} \cdot \mathbf{x}_i))_+$ with $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla J(\boldsymbol{\theta})$

Support Vector Machine

Hard margin SVM,
 $\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|^2$, such that $y_i \boldsymbol{\theta} \cdot \mathbf{x}_i \geq 1 (i = 1, \dots, n)$
Soft margin SVM,
 $\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^n (1 - y_i \boldsymbol{\theta} \cdot \mathbf{x}_i)_+$

Regularization and SVMs: Simulated data with many features $\phi(\mathbf{x})$; C controls trade-off between margin $1/\|\boldsymbol{\theta}\|$ and fit to data; Large C: focus on fit to data (small margin is ok). More overfitting. Small C: focus on large margin, less tendency to overfit. Overfitting increases with: less data, more features.

$\boldsymbol{\theta} = \sum_j \alpha_j y_j \mathbf{x}_j$, $\alpha_j \neq 0$ only for support vectors.

Width of the margin is $\frac{2}{\|\boldsymbol{\theta}\|}$.

$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, K is called a kernel.
Solve α_j to determine $\sum_j \alpha_j y_j \phi(\mathbf{x}_j)$.
Compute the classifier for a test point **x** via
 $\boldsymbol{\theta} \cdot \phi(\mathbf{x}) = \sum_j \alpha_j y_j K(\mathbf{x}_j, \mathbf{x})$

degree-m polynomial kernel: $K_m(\mathbf{x}, \tilde{\mathbf{x}}) = (1 + \mathbf{x} \cdot \tilde{\mathbf{x}})^m$
RBF kernel (infinite dimention): $K_{rbf}(\mathbf{x}, \tilde{\mathbf{x}}) = \exp(-\gamma \|\mathbf{x} - \tilde{\mathbf{x}}\|^2)$

Decision Theory

Loss function: $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, and $l(\hat{y}, y)$ is the cost of predicting \hat{y} when the outcome is y .

Risk for a given class: $R(\alpha_i|x) = \sum_{j=1}^C \lambda_{ij} P(w = j|x)$

Assume **(X, Y)** are chosen i.i.d according to some probability distribution on $\mathcal{X} \times \mathcal{Y}$. **Risk** is misclassification probability:
 $R(f) = \mathbb{E}l(f(\mathbf{X}), \mathbf{Y}) = Pr(f(\mathbf{X}) \neq \mathbf{Y})$

Bayes Decision Rule is
 $f^*(x) = \begin{cases} 1, & \text{if } P(\mathbf{Y} = 1|x)L_{f(x)=-1} > P(\mathbf{Y} = -1|x)L_{f(x)=1}, \\ -1, & \text{otherwise.} \end{cases}$,
and the optimal risk (Bayes risk) $R^* = \inf_f R(f) = R(f^*)$

Excess risk is for any $f : \mathcal{X} \rightarrow \{-1, +1\}$,
 $R(f) - R^* = \mathbb{E}(1[f(x) \neq f^*(x)]|2P(\mathbf{Y} = +1|\mathbf{X}) - 1|)$

Risk in Regression is expected squared error:
 $R(f) = \mathbb{E}l(f(\mathbf{X}), \mathbf{Y}) = \mathbb{E}\mathbb{E}[f(\mathbf{X}) - \mathbf{Y}^2|\mathbf{X}]$

Bias-variance decomposition:
 $R(f) = \mathbb{E}[\underbrace{(f(\mathbf{X}) - \mathbb{E}[\mathbf{Y}|\mathbf{X}])^2}_{\text{bias}^2} + \mathbb{E}[\underbrace{(\mathbb{E}[\mathbf{Y}|\mathbf{X}] - \mathbf{Y})^2}_{\text{variance}}]]$
 $R(f) = \mathbb{E}[(f(\mathbf{X}) - f^*(\mathbf{X}))^2] + \mathbb{E}[(f^*(\mathbf{X}) - \mathbf{Y})^2]$
 $R(f) = \mathbb{E}[(f(\mathbf{X}) - f^*(\mathbf{X}))^2] + R(f^*)$
 $R(f) - R(f^*) = \mathbb{E}[(f(\mathbf{X}) - f^*(\mathbf{X}))^2], f^*(\mathbf{X}) = \mathbb{E}[\mathbf{Y}|\mathbf{X}]$

Generative and Discriminative

Discriminative models: $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X})P(\mathbf{Y}|\mathbf{X})$.
Estimate $P(\mathbf{Y}|\mathbf{X})$, then pretend out estimate $\hat{P}(\mathbf{Y}|\mathbf{X})$ is the actual $P(\mathbf{Y}|\mathbf{X})$ and plug in bayes rule expression.

Generative model: $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y})P(\mathbf{X}|\mathbf{Y})$.
Estimate $P(\mathbf{Y})$ and $P(\mathbf{X}|\mathbf{Y})$, then use bayes theorem to calculate $P(\mathbf{Y}|\mathbf{X})$ and use discriminative model.

Gaussian class conditional densities $P(\mathbf{X}|Y = +1), P(\mathbf{X}|Y = -1)$ (with the same variance), the posterior probability is **logistic**:
 $P(Y = +1|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x} \cdot \boldsymbol{\beta} - \beta_0)}$,
 $\boldsymbol{\beta} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \beta_0 = \frac{\boldsymbol{\mu}_0' \Sigma^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1' \Sigma^{-1} \boldsymbol{\mu}_1}{2} + \log \frac{P(Y=1)}{P(Y=0)}$

Estimation

Method of moments: Match moments of the distribution to momemnts measured in the data.

Maximum likelihood: Choose parameter so that the distribution it defines gives the observed data the highest probability (likelihood).

Maximum log likelihood: Log of maximum likelihood, equivalent to maximum likelihood since log is monotonically increase; it is useful since it can change \prod to \sum .

Penalized maximum likelihood: Add a penalty term in the maximum (log) likelihood equation; treat the penalty term as some imaginary data points crafted for desired probability.

Bayesian estimate: Treat parameter as a random variable, then update based on observed value (data).
Prior: $\pi(p) = 1$,
Posterior: $P(p|\mathbf{X}_1 = 1) = P(\mathbf{X}_1 = 1|p)\pi(p)/\int P(X_1 = 1|q)d\pi(q)$

Maximum a posterior probability: the mode of the posterior. If uniform prior, MAP is MLE; if not uniform prior, MAP is Penalized MLE.

Multivariate Normal Distribution

$$\mathbf{x} \in \mathbb{R}^d : p(x) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} e^{(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))}$$

Covariance matrix: $\boldsymbol{\Sigma} = \mathbb{E}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'$
Symmetric: $\boldsymbol{\Sigma}_{i,j} = \boldsymbol{\Sigma}_{j,i}$
Non-negative diagonal entries: $\boldsymbol{\Sigma}_{i,i} \geq 0$
Positive semidefinite: $\forall \mathbf{v} \in \mathbb{R}^d, \mathbf{v}' \boldsymbol{\Sigma} \mathbf{v} \geq 0$

Super-level sets of pdf:
 $\boldsymbol{\xi}_r = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq r^2\}$.
Volume of $\boldsymbol{\xi}_r \propto \prod_{i=1}^d \sigma_i = \sqrt{|\boldsymbol{\Sigma}|}$

Spectral Theorem for non-diagonal covariance:
 $U = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n], \boldsymbol{\Lambda} = \text{diag}([\lambda_1, \lambda_2, \dots, \lambda_n])'$
We can eigen decompose $\boldsymbol{\Sigma}^{-1} = U \boldsymbol{\Lambda}^{-1} U'$, this is like to change to a different eigen spaces, where covariances (**Λ**) diagonal axis-alianed.

Assume independent,
 $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}) + \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y)$

Given a *d*-dimensaional Gaussian $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,
write $\mathbf{X} = \begin{bmatrix} \mathbf{Y} \\ \mathbf{Z} \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_Y \\ \boldsymbol{\mu}_Z \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{YY} & \boldsymbol{\Sigma}_{YZ} \\ \boldsymbol{\Sigma}_{ZY} & \boldsymbol{\Sigma}_{ZZ} \end{bmatrix}$,
where $\mathbf{Y} \in \mathbb{R}^m$, and $\mathbf{Z} \in \mathbb{R}^{d-m}$. Then $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_{YY})$

Given a *d*-dimensaional Gaussian $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,
matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ and vector $\mathbf{b} \in \mathbb{R}^m$, define $\mathbf{Y} = \mathbf{AX} + \mathbf{b}$. Then $\mathbf{Y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}')$

Given a *d*-dimensaional Gaussian $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,
with $\boldsymbol{\Sigma}$ positive definite,
 $\mathbf{Y} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{X} - \boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$

Gaussian maximum likelihood estimation:
Sample mean: $\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$;
Sample covariance: $\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})'$

Linear Regression

Given $\mathbf{X} \in \mathbb{R}^p, Y \in \mathbb{R}$, consider linear(affine) prediction rules,
 $F_{\text{lin}} := \{\mathbf{x} \mapsto \mathbf{x}'\boldsymbol{\beta} + \beta_0 : \boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}\}$

Empirical risk minimization

Empirical risk is the sample average of squared error:
 $\hat{R}(f) = \hat{\mathbb{E}}_n \ell(f(\mathbf{X}), Y) = \frac{1}{n} \sum_{i=1}^n n(f(\mathbf{X}_i) - Y_i)^2$
Choose $\hat{f} := \arg \min_{f \in F_{\text{lin}}} \hat{\mathbb{E}}_n \ell(f(\mathbf{X}), Y)$

Find $\hat{f}: \mathbf{x} \mapsto \mathbf{x}'\hat{\beta}$, such that
 $\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^n (\mathbf{X}'_i \beta - Y_i)^2 = \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\|\mathbf{X}\beta - \mathbf{y}\|^2}_{\text{RSS}}$
where **design matrix** $\mathbf{X} \in \mathbb{R}^{n \times p}$ and **response vector** $\mathbf{y} \in \mathbb{R}^n$.

Normal equations: $\mathbf{X}'\mathbf{X}\beta = \mathbf{X}'\mathbf{y} \Rightarrow \hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$

Projection Theorem also leads to normal equations:
 $(\mathbf{y} - \hat{\mathbf{y}})^{-1}\mathbf{X} = 0 \Leftrightarrow \mathbf{X}'(\mathbf{y} - \mathbf{X}\beta) = 0 \Leftrightarrow \mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}\beta$

Linear model with additive Gaussian noise

Model the conditional distribution of Y given $\mathbf{X} = \mathbf{x}$ as:
 $P(Y|\mathbf{X} = \mathbf{x}) = \mathcal{N}(\mathbf{x}'\beta, \sigma^2)$
Equivalently: $Y = \mathbf{x}'\beta + \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2)$

Maximum likelihood is least square:
 $L(\beta) = \prod_{i=1}^n p(Y_i|\mathbf{X}_i, \beta) \Leftrightarrow \ell(\beta) = g(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mathbf{X}'_i \beta)^2$

Fix \mathbf{X} . Provided $\mathbb{E}\mathbf{y} = \mathbf{X}\beta$ and $\text{Cov}(\mathbf{y}) = \sigma^2\mathbf{I}$

Bayesian analysis: Treat β as a r.v. with prior distribution $\mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$, then compute posterior distribution $P(\beta|\mathbf{X}, Y)$.

$P(\beta|\mathbf{X}_1, Y_1, \dots, \mathbf{X}_n, Y_n) \propto P(Y_1, \dots, Y_n|\beta, \mathbf{X}_1, \dots, \mathbf{X}_n)P(\beta)$
 $P(\beta|\mathbf{X}_1, Y_1, \dots, \mathbf{X}_n, Y_n) \propto \exp(-\frac{1}{2}(\sum_{i=1}^n \frac{(Y_i - \mathbf{X}'_i \beta)^2}{\sigma^2} + \frac{1}{\tau^2}\|\beta\|^2))$

Linear Regression Regularization

Trading off bias and variance: some increase in bias can give a big decrease in variance.

Subset selection is like L_0 regularization: RSS decreases as the complexity increases because the best fit with a smaller subset is always possible with a larger subset.

Find a path through subset space: using cross-validation and forward-stepwise selection or backward-stepwise selection (need $n > p$).

Ridge regression is like L_2 regularization:
 $\hat{\beta} = \arg \min_{\beta} (\sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda \sum_{j=1}^p p\beta_j^2)$
 $\hat{\beta}^{\text{ridge}} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{x}'\mathbf{y}$

Lasso regression is like L_1 regularization:
 $\hat{\beta} = \arg \min_{\beta} (\sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda \sum_{j=1}^p p|\beta_j|)$
While ridge regression leads to reduced, but non-zero values of the coefficients, Lasso regression forces some coefficients to be zero.

Bayesian analysis: Ridge regression is equivalent to a MAP estimate with a gaussian prior. Lasso regression is equivalent to a MAP estimate with a Laplace prior.

Logistic Regression

Model **log odds** $(\log p/(1-p))$ as an affine function of \mathbf{x} .
 $P(Y = 1|\mathbf{x}) = \frac{1}{1+\exp(\beta'\mathbf{x})}$ Given data
 $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n) \in \mathbb{R}^p \times \{0, 1\}$, estimate β with maximum likelihood.

Log likelihood:
 $\ell(\beta) = \sum_{i=1}^n y_i \log \mu_i(\beta) + (1 - y_i) \log(1 - \mu_i(\beta))$,
where $\mu_i(\beta) = P(Y = 1|\mathbf{X} = \mathbf{x}_i, \beta) = \frac{1}{1+\exp(-\beta'\mathbf{x}_i)}$

$\nabla_{\beta} \mu_i(\beta) = \mu_i(\beta)(1 - \mu_i(\beta))\mathbf{x}_i$
 $\nabla_{\beta} \ell(\beta) = \sum_{i=1}^n (y_i - \mu_i(\beta))\mathbf{x}_i = \mathbf{X}'(\mathbf{y} - \boldsymbol{\mu})$
 $\nabla_{\beta}^2 \ell(\beta) = \sum_{i=1}^n -\mu_i(\beta)(1 - \mu_i(\beta))\mathbf{x}_i \mathbf{x}'_i = -\mathbf{X}'\text{diag}(\boldsymbol{\mu}(1 - \boldsymbol{\mu}))\mathbf{X}$
 $\hat{\beta}^{\text{ml}}$ solves: $\sum_{i=1}^n y_i \mathbf{x}_i = \sum_{i=1}^n \mu_i \beta \mathbf{x}_i$

Gradient ascent:
 $\beta^{(t+1)} = \beta^{(t)} + \eta \nabla_{\beta} \ell(\beta^{(t)}) : O(np)/\text{step}$
Stochastic gradient ascent:
 $\beta^{(t+1)} = \beta^{(t)} + \eta (y_{it} - \mu_{it}(\beta^{(t)}))\mathbf{x}_{it} : O(p)/\text{step}$
Newton-Raphson method:
 $\beta^{(t+1)} = \beta^{(t)} - [\nabla_{\beta}^2 \ell(\beta^{(t)})]^{-1} \nabla_{\beta} \ell(\beta^{(t)})$

Newton's method for root finding: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

Prediction $\hat{p}(y|\mathbf{x}) = \begin{cases} P(Y = 1|\mathbf{x}), & \text{if } y = 1 \\ P(Y = -1|\mathbf{x}), & \text{if } y = -1 \end{cases}$

Log loss (Binomial Deviance): $\ell_{\log}(\hat{p}(\cdot|\mathbf{x}), y) = -\log(\hat{p}(y|\mathbf{x}))$
Minimize: $\frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \beta' \mathbf{x}_i))$

Linear Discriminant Analysis

Linear discriminant functions:
 $\delta_k(\mathbf{x}) = \mu'_k \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \mu'_k \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k$

Estimate with Maximum likelihood:
 $\pi_k = P(Y = k) \Leftrightarrow \hat{\pi}_k = \frac{n_k}{n}$
 $\mu_k = \mathbb{E}[\mathbf{X}|Y = k] \Leftrightarrow \hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i = k} \mathbf{x}_i$
 $\boldsymbol{\Sigma} = \text{Var}[\mathbf{X}|Y = k] \Leftrightarrow \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_k \sum_i i : y_i = k (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)'$

SVM with Convex Optimization

Lagrangian: rewrite constraint as penalties for a convex optimization problem such that $L(x, \lambda) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x)$.

Weak duality: $p^* = \min_x \max_{\lambda \geq 0} L(x, \lambda) \geq \max_{\lambda \geq 0} \min_x L(x, \lambda) = d^*$
primal dual

Strong duality:
if there is a saddle point (x^*, λ^*) such that for all x and $\lambda \geq 0$, $L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$, then primal and dual have the same value ($p^* = d^*$).

Karush-Kuhn-Tucker optimality conditions:
Primal feasibility: $f_i(x) \leq 0$; Dual feasibility: $\lambda_i \geq 0$
Complementary slackness: $\lambda_i f_i(x) = 0$
Stationarity: $\nabla f_0(x) + \sum_i \lambda_i \nabla f_i(x) = 0$

Hard margin SVM:
 $L(\theta, \alpha) = \frac{1}{2} \|\theta\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i \theta' \mathbf{x}_i)$
 $g(\alpha) = \min_{\theta} L(\theta, \alpha)$
setting $\theta^* = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$,
 $g(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$

Hard margin SVM dual problem:
 $\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$, s.t. $\alpha_i \geq 0$ ($i = 1, \dots, n$).
 $\min_{\alpha} \frac{1}{2} \alpha' \text{diag}(\mathbf{y}) \mathbf{K} \text{diag}(\mathbf{y}) \alpha - \alpha' \mathbf{1}$, s.t. $\alpha \geq \mathbf{0}$.

Soft margin SVM:
 $L(\theta, \xi, \alpha, \lambda) = \frac{1}{2} \|\theta\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - y_i \theta' \mathbf{x}_i - \xi_i) - \sum_{i=1}^n \lambda_i \xi_i$

Soft margin SVM dual problem:
 $\min_{\alpha} \frac{1}{2} \alpha' \text{diag}(\mathbf{y}) \mathbf{K} \text{diag}(\mathbf{y}) \alpha - \alpha' \mathbf{1}$, s.t. $\mathbf{0} \leq \alpha \leq \frac{C}{n}$.

Clustering

Group similarity $\text{dist}\{S_a, S_b\}$:
Single link: $\min_{i,j} d_{i,j}$ Complete link: $\max_{i,j} d_{i,j}$
Average link: $\sum_i \sum_j d_{i,j}$ Centroid link: $d(\bar{x}_a, \bar{x}_b)$

EM algorithm:
E step:
 $P(\mu_i | x_k) = \text{frac} P(\mu_i) P(x_k | \mu) P(x_k) = \frac{P(\mu_i) P(x_k | \mu_i)}{\sum_j P(\mu_j) P(x_k | \mu_j)}$

M step: $P(c_i) = \frac{\sum_{k=1}^{n_e} P(\mu_i | x_k)}{n_e}$
 $\mu_i = \frac{\sum_{k=1}^{n_e} x_k P(\mu_i | x_k)}{\sum_{k=1}^{n_e} P(\mu_i | x_k)}$
 $\sigma_i^2 = \frac{\sum_{k=1}^{n_e} (x_k - \mu_i)^2 P(\mu_i | x_k)}{\sum_{k=1}^{n_e} P(\mu_i | x_k)}$

Association rules

Rule $X \rightarrow Y$: $\text{supp}(X) = P(X)$, $\text{conf}(X \rightarrow Y) = P(Y | X)$ and $\text{lift}(X \rightarrow Y) = \frac{P(X \cup Y)}{P(X)P(Y)}$.

Decision Tree

Classification tree entropy: $\sum_i -p_i * \log(p_i)$ Maximize entropy gain

- Adaboost:**
- Assign uniform weight for each training sample $w_{1,i} = \frac{1}{N}$ for i in $\{1, 2...N\}$
2a. For each round of training m in $\{1, 2...M\}$ train classifier G_m
2b. Calculate weighted error rate $e_m = \sum_{i=1}^N w_{m,i} I(G_m(x_i) \neq y_i)$
2c. The final voting weight for G_m is $a_m = \frac{1}{2} \log(\frac{1-e_m}{e_m})$
2d. Update training sample weight $w_{m+1,i} = \frac{w_{m,i}}{Z_m} \exp(-a_m y_i G_m(x_i))$, $i = 1, 2..., N$ Where $Z_m = \sum_{i=1}^N w_{m,i} \exp(-a_m y_i G_m(x_i))$
 - The final classifier is $f(x) = \sum_{m=1}^M a_m G_m(x)$

Neural Network

Useful gradient:
 $\nabla_h \left[-y \log h - (1-y) \log(1-h) \right] = \frac{1-y}{1-h} - \frac{y}{h}$
Sigmoid g : $\nabla_x g(x) = g(x) * (1 - g(x))$ $\nabla_x \tanh(x) = 1 - \tanh^2(x)$
Fully connected layer i with bias:
 $\nabla_W L = [x_{i-1} \dots x_{i-1}] * [\vec{0}, \nabla_{c_i} A(c_i)] * W_{i+1} * \nabla_{c_{i+1}} L$

PCA

Matrix decomposition:
 $X = U \Sigma W^T$ so $X^T X = W \Sigma^T \Sigma W^T$ and $XX^T = U \Sigma \Sigma^T U^T$
Hence, for row vector design matrix, the projection matrix is W^T