

Probability

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

Matrix calculus

$$f(\mathbf{x}) = \mathbf{x}'\mathbf{A}\mathbf{B} + \mathbf{A}\mathbf{B}\mathbf{x} + \mathbf{x}'\mathbf{A}\mathbf{B}\mathbf{x} \Leftrightarrow \frac{df(\mathbf{x})}{d\mathbf{x}} = \mathbf{A}\mathbf{B} + \mathbf{A}\mathbf{B} + 2\mathbf{A}\mathbf{B}\mathbf{x}$$

$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}; \nabla_x(a\mathbf{x}) = a\mathbf{I}$$

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  $\mathbb{R}^d$ :  
 $H = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) = 0\} = \{\mathbf{x} \in \mathbb{R}^d : \boldsymbol{\theta} \cdot \mathbf{x} + \theta_0 = 0\}$

$\boldsymbol{\theta}$  is the **normal** of the hyperplane,  
 $\theta_0$  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  $\mathbf{x}$  in the direction  $\boldsymbol{\theta}$ .

$\boldsymbol{\theta} = \sum_i \alpha_i y_i \mathbf{x}_i$ , 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.

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

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

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

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  $y$ .

Assume  $(\mathbf{X}, \mathbf{Y})$  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

$$f^*(x) = \begin{cases} 1, & \text{if } P(\mathbf{Y} = 1|x) > P(\mathbf{Y} = -1|x) \\ -1, & \text{otherwise.} \end{cases}$$

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

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 ( $\boldsymbol{\Lambda}$ ) 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  $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$ .  
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,  
 $\mathbf{Y} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{X} - \boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

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  $\hat{f} : \mathbf{x} \mapsto \mathbf{x}'\hat{\boldsymbol{\beta}}$ , such that  
 $\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \underbrace{\sum_{i=1}^n (\mathbf{X}'_i \boldsymbol{\beta} - Y_i)^2}_{\text{RSS}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2$

where **design matrix**  $\mathbf{X} \in \mathbb{R}^{n \times p}$  and **response vector**  $\mathbf{y} \in \mathbb{R}^n$ .

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

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

Linear model with additive Gaussian noise

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

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

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

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

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

Linear Regression Regularization

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

**Subset selection** is like  $L0$  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  $L2$  regularization:  
 $\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} (\sum_{i=1}^n (y_i - \mathbf{x}'_i \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p p\beta_j^2)$   
 $\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{x}'\mathbf{y}$

**Lasso regression** is like  $L1$  regularization:  
 $\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} (\sum_{i=1}^n (y_i - \mathbf{x}'_i \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p p|\beta_j|)$

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  $\mathbf{x}$ .

$P(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp(\boldsymbol{\beta}'\mathbf{x})}$  Given data  
 $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n) \in \mathbb{R}^p \times \{0, 1\}$ , estimate  $\boldsymbol{\beta}$  with maximum likelihood.

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

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

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

**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:**  $\frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \boldsymbol{\beta}'\mathbf{x}_i))$

Linear Discriminant Analysis

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

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

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

**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.  $\alpha_i \geq 0$  ( $i = 1, \dots, n$ ).  
 $\min_{\alpha} \frac{1}{2} \boldsymbol{\alpha}' \text{diag}(\mathbf{y}) \mathbf{K} \text{diag}(\mathbf{y}) \boldsymbol{\alpha} - \boldsymbol{\alpha}' \mathbf{1}$ , s.t.  $\boldsymbol{\alpha} \geq \mathbf{0}$ .

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

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