Exam Cheat Sheet

Discriminative Functions for Classifiers

Naive Bayes

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{1}$$

$$P(x|y) = \prod_{i=1}^{n} P(x_i|y) \tag{2}$$

Fisher's Linear Discriminant Analysis

$$w = S_W^{-1}(\mu_1 - \mu_2) \tag{3}$$

$$S_W = \frac{1}{N_1}\sum_{n \in C_1}(x_n - \mu_1)(x_n - \mu_1)^T + \frac{1}{N_2}\sum_{n \in C_2}(x_n - \mu_2)(x_n - \mu_2)^T \tag{4}$$

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{5}$$

Logistic Regression

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

$$y(x) = \phi(w^T x) \tag{7}$$

$$J(\theta) = -\frac{1}{m}[y^T \log(\sigma(X\theta)) + (1-y)^T \log(1 - \sigma(X\theta))] \tag{8}$$

Perceptron Algorithm

$$\mathcal{E}_p(\mathbf{w}, w_0) = -\sum_{n \in M} t_n(\mathbf{w} \cdot \mathbf{x}_n + w_0), \text{ where } M \text{ is misclass} \tag{9}$$

MLP

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \phi(\mathbf{W}^T\mathbf{X} + \mathbf{b}) = \phi(\mathbf{W}^T\phi(\mathbf{V}^T\mathbf{X} + \mathbf{c}) + \mathbf{b}) \tag{10}$$

Gradient Descent

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla J(\mathbf{w}^{(t)}) \tag{11}$$

Regularization & Optimization

Pooling Techniques

Max Pooling

$$\text{Max Pooling: } y_{i,j} = \max_{m,n} x_{i+m,j+n} \tag{12}$$

Average Pooling

$$\text{Average Pooling: } y_{i,j} = \frac{1}{mn}\sum_{m,n} x_{i+m,j+n} \tag{13}$$

Gradient Descent Methods

Gradient descent methods are foundational in optimizing neural network parameters, minimizing the loss function iteratively.

Stochastic Gradient Descent (SGD)

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla J(\mathbf{w}^{(t)}, \mathbf{x}_n, t_n) \tag{14}$$

Where $\mathbf{x}_n$ is a randomly selected training example, η is the learning rate.

Nesterov's Accelerated Gradient (NAG)

$$\mathbf{v}^{(t+1)} = \gamma\mathbf{v}^{(t)} + \eta\nabla J(\mathbf{w}^{(t)} - \gamma\mathbf{v}^{(t)}) \tag{15}$$

Where γ is the momentum term.

Adaptive Moment Estimation Methods

These methods adapt the learning rates based on lower-order moments of gradients and provide an efficient way to converge faster.

Adam

$$m^{(t+1)} = \beta_1 m^{(t)} + (1 - \beta_1)\nabla J(\mathbf{w}^{(t)}) \tag{16}$$

$$v^{(t+1)} = \beta_2 v^{(t)} + (1 - \beta_2)\nabla J(\mathbf{w}^{(t)})^2 \tag{17}$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\frac{m^{(t+1)}}{\sqrt{v^{(t+1)} + \epsilon}} \tag{18}$$

Where m and v are the first and second moment estimates, respectively, β_1 and β_2 are the decay rates, and ϵ is a small constant to prevent division by zero.

Nadam

$$m^{(t+1)} = \beta_1 m^{(t)} + (1 - \beta_1)\nabla J(\mathbf{w}^{(t)}) \tag{19}$$

$$v^{(t+1)} = \beta_2 v^{(t)} + (1 - \beta_2)\nabla J(\mathbf{w}^{(t)})^2 \tag{20}$$

Where m and v are the first and second moment estimates, respectively, β_1 and β_2 are the decay rates, and ϵ is a small constant to prevent division by zero.

Support Vector Machines

Kernel Machines

$$K(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2}) = \exp(-\gamma\|x - x'\|^2) \tag{21}$$

$$\gamma = \frac{1}{2\sigma^2} \tag{22}$$

Hard Margin SVM

$$\min_{w,b} \frac{1}{2}\|w\|^2 \text{ s.t. } y_i(w^T x_i + b) \geq 1 \tag{23}$$

$$\mathcal{L}(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N}\alpha_i[t_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1] \tag{24}$$

Soft Margin SVM

$$\mathcal{L}(\mathbf{w}, w_0, \alpha, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i \tag{25}$$

Performance Metrics

Confusion Matrix

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{26}$$

Precision

$$\frac{TP}{TP+FP} \tag{27}$$

Recall

$$\frac{TP}{TP+FN} \tag{28}$$

F1 Score

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{29}$$

Accuracy

$$\frac{TP+TN}{TP+FP+FN+TN} \tag{30}$$

ROC Curve

$$\text{TPR} = \frac{TP}{TP+FN} \tag{31}$$

$$\text{FPR} = \frac{FP}{FP+TN} \tag{32}$$

Dimensionality Reduction

PCA

$$\mathbf{X} = \mathbf{X} - \bar{\mathbf{X}} \tag{33}$$

$$\mathbf{Cov} = \frac{1}{N}\mathbf{X}^T\mathbf{X} \tag{34}$$

Eigenvectors and Eigenvalues

Eigendeomposition of the 3×3 covariance matrix.

$$\mathbf{Cov} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T \tag{35}$$

Where $\mathbf{Q}$ is the matrix of eigenvectors and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues.

$$\mathbf{C_{ov}}\mathbf{v} = \lambda\mathbf{v} \tag{36}$$

$$C_{ov} = \mathbf{E}[\mathbf{X}\mathbf{X}^{\mathbf{T}}] \tag{37}$$

When $\mathbf{X}$ is mean-centered.

Manifold Learning

$$\mathbf{d}_{Euclidean} = \sqrt{\sum_{i=1}^{N}(x_i - x_j)^2} \tag{38}$$

$$\mathbf{d}_{Geodesic} = \min_{\mathbf{p}} \sum_{i=1}^{N-1} \sqrt{\sum_{j=1}^{N}(p_{i,j} - p_{i+1,j})^2} \tag{39}$$

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,N} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N,1} & d_{N,2} & \cdots & d_{N,N} \end{bmatrix} \tag{40}$$

$$\mathbf{D}^2 = \begin{bmatrix} d_{1,1}^2 & d_{1,2}^2 & \cdots & d_{1,N}^2 \\ d_{2,1}^2 & d_{2,2}^2 & \cdots & d_{2,N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{N,1}^2 & d_{N,2}^2 & \cdots & d_{N,N}^2 \end{bmatrix} \tag{41}$$

$$d_{i,j}^2 = b_{i,i} + b_{j,j} - 2b_{i,j} \tag{42}$$

Where $b_{i,j}$ is the element of the matrix $\mathbf{B}$.

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}^2\mathbf{J} \tag{43}$$

$$\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T \tag{44}$$

Where $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector of ones and $\mathbf{1}\mathbf{1}^T$ is the outer product of $\mathbf{1}$.