Statistical Learning & Inference

Based on lectures by Li Niu

Notes taken by Shaobo Wang

CS7335 2022

Contents

1 Curse of dimensionality

1. Temporal and spatial complexity: storage cost and time cost
2. Require more training samples to fill in the space
3. Feature dimension represents model complexity.
    (a) High model complexity, few training samples: overfitting
    (b) Low model complexity, abundant training samples: underfitting
    (c) With fixed number of training samples, high dimension can easily cause overfitting.
4. For a high-dimensional object, most of its volume is near the surface.

    Example. Given a d-dim hypercube, to capture a fraction s of the volume, you need the edge length to be $r' = s^{1/d}$.

5. Distances to near and to far neighbors become more and more similar with increasing dimensionality. Therefore, distance metric starts losing their effectiveness to measure dissimilarity in high-dim space.
6. Since classifiers depend on these distance metrics, it is more difficult to learn a good classifier.

    Example. $\mathbf{x}, \mathbf{y}$ are two independent variables, with uniform distribution on $[0, 1]^d$ The mean square distance $\|\mathbf{x} - \mathbf{y}\|^2$ satisfies $\mathbb{E}\left[\|\mathbf{x} - \mathbf{y}\|^2\right] = \frac{d}{6}$

7. How to avoid the curse of dimensionality?
    (a) Sample enough training data
    (b) Reduce feature dimensionality

2 Bias/variance, overfitting/underfitting

2.1 Meaning

1. Bias: many samples are classified as wrong labels. Bias refers to the error rate of the training data. The difference between the error rate of different datasets is called variance.
2. Variance: although the model achieves good performance on set A, but has bad performance (low accuracy) on set B. If a model performances very differently on different sets, we think that the model has a large variance.
3. Underfit: Underfitting refers to a model that can neither performs well on the training data nor generalize to new data. **Solution: Increase model parameters; Reduce training set**
4. Overfit: Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data. **Solution: Reduce model parameters; Enlarge training set**

2.2 Bias-Variance Tradeoff

1. Basic Model: $Y = f(X) + \varepsilon, \quad \varepsilon \sim N\left(0, \sigma_{\varepsilon}^{2}\right)$

2. The expected prediction error of a regression fit $\hat{f}(X)$.

$$\begin{aligned}
\operatorname{Err}\left(x_{0}\right) &= E\left[\left(Y-\hat{f}\left(x_{0}\right)\right)^{2} \mid X=x_{0}\right] \\
&= \sigma_{\varepsilon}^{2}+\left[\mathrm{E} \hat{f}\left(x_{0}\right)-f\left(x_{0}\right)\right]^{2}+E\left[\hat{f}\left(x_{0}\right)-\mathrm{E} \hat{f}\left(x_{0}\right)\right]^{2} \\
&= \sigma_{\varepsilon}^{2}+\operatorname{Bias}^{2}\left(\hat{f}\left(x_{0}\right)\right)+\operatorname{Var}\left(\hat{f}\left(x_{0}\right)\right) \\
&= \text{ Irreducible Error }+\text{ Bias}^{2}+\text{ Variance.}
\end{aligned}$$

3. The more complex the model, the lower the (squared) bias but the higher the variance.

Example (KNN)**.**

$$\hat{f}_{k}\left(x_{0}\right)=\frac{1}{k} \sum_{j=1}^{k} f\left(x_{j}\right), \quad x_{j} \in N\left(x_{0}\right)$$

$$\begin{aligned}
\operatorname{Err}\left(x_{0}\right) &= E\left[\left(Y-\hat{f}_{k}\left(x_{0}\right)\right)^{2} \mid X=x_{0}\right] \\
&= \sigma_{\varepsilon}^{2}+\left[f\left(x_{0}\right)-\frac{1}{k} \sum_{\ell=1}^{k} f\left(x_{(\ell)}\right)\right]^{2}+\frac{\sigma_{\varepsilon}^{2}}{k}.
\end{aligned}$$

Example (linear regression)**.** For the linear model fit $\hat{f}_{d}(x)=\hat{\beta}^{T} x$. Solution is $\hat{\beta}=\left(X^{T} X\right)^{-1} X^{T} y$, $\mathbf{h}\left(x_{0}\right)=\mathbf{X}\left(\mathbf{X}^{T} \mathbf{X}\right)^{-1} x_{0}$

$$\begin{aligned}
\operatorname{Err}\left(x_{0}\right) &= E\left[\left(Y-\hat{f}_{d}\left(x_{0}\right)\right)^{2} \mid X=x_{0}\right] \\
&= \sigma_{\varepsilon}^{2}+\left[f\left(x_{0}\right)-E \hat{f}_{d}\left(x_{0}\right)\right]^{2}+\left\|h\left(x_{0}\right)\right\|^{2} \sigma_{\varepsilon}^{2}
\end{aligned}$$

In-sample Error:

$$\frac{1}{N} \sum_{i=1}^{N} \operatorname{Err}\left(x_{i}\right)=\sigma_{\varepsilon}^{2}+\frac{1}{N} \sum_{i=1}^{N}\left[f\left(x_{i}\right)-\mathrm{E} \hat{f}_{d}\left(x_{i}\right)\right]^{2}+\frac{d}{N} \sigma_{\varepsilon}^{2},$$

3 Regression

3.1 Linear Regression

1. Input x_i Output y_i

2. Regression function: $f(x) = \mathbb{E}(Y|x)$

3. Linear Regression model: $f(x) = \beta_0 + \beta x$

4. Minimize: $\hat{\beta}_0, \hat{\beta} = \text{argmin}_{\beta_0,\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \beta x_i)^2$
   (a) Prediction: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta} x_i$
   (b) Residuals: $\xi_i = y_i - \left(\hat{\beta}_0 + \hat{\beta} x_i\right)$
5. Multiple Linear Regression:
   (a) Model is $f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{p-1} x_{ij}\beta_j$
   (b) Equivalently in matrix notation: $\mathbf{f} = \mathbf{X}\boldsymbol{\beta}$ $\quad \mathbf{x}_i = [1; x_{i,1}; \ldots; x_{i,p-1}]$ $\quad \boldsymbol{\beta} = [\beta_0; \beta_1; \ldots; \beta_{p-1}]$,
   where f is N-dim vector of predicted values $\mathbf{X}$ is $N \times p$ input matrix $\boldsymbol{\beta}$ is a p-dim vector of model parameters

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \sum_i \left(y_i - \beta_0 - \sum_{j=1}^{p-1} x_{ij}\beta_j\right)^2 \\ &= \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\end{aligned}$$

   Solution is $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$
   Prediction is $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$

3.2 Ridge Regression

1. The ridge estimator is defined by

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min(\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

   Equivalently,

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{\text{ridge}} &= \arg\min(\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &\text{subject to } \|\boldsymbol{\beta}\|^2 \leq s\end{aligned}$$

   The parameter $\lambda > 0$ penalizes $\|\beta\|^2$

$$\hat{\boldsymbol{\beta}}_\lambda = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{y}$$

2. Note $\lambda = 0$ gives the least squares estimator; if $\lambda \to \infty$, then $\hat{\beta} \to 0$
3. SVD

SVD

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

D is a diagonal matrix with $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_p \geq 0$.
Note that $U^TU = I, V$ is invertable.

   (a) For Ordinary Regression

$$\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{ls}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{U}\mathbf{U}^T\mathbf{y}$$

(b) For Ridge Regression

$$\begin{aligned}\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{ridge}} &= \mathbf{X}\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}\\ &= \mathbf{UD}(\mathbf{DD}+\lambda\mathbf{I})^{-1}\mathbf{DU}^T\mathbf{y} = \sum_{j=1}^{p}\mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y}\end{aligned}$$

3.3 Lasso Regression

1. The lasso is a shrinkage method like ridge, but acts in a nonlinear manner on the outcome **y**.
2. The lasso is defined by

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{\text{lasso}} &= \arg\min(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\\ &\text{subject to } \|\boldsymbol{\beta}\|_1 \le t\end{aligned}$$

3. This makes the solutions nonlinear in y, and a quadratic programming algorithm is used to compute them.
4. Because of the nature of the constraint, if t is chosen small enough, then the lasso will set some coefficients exactly to zero. Thus, the lasso does a kind of continuous model selection.
5. General Form

   (a) Consider the criterion

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \quad \text{subject to } \|\boldsymbol{\beta}\|_q^q \le s \quad \|\boldsymbol{\beta}\|_q = \left(\sum_i |\beta_i|^q\right)^{\frac{1}{q}}$$

   (b) for $q >= 0$. The contours of $\|\boldsymbol{\beta}\|_q^q = s$ are shown for the case of two inputs.

3.4 Logistic Regression

$$\begin{aligned}\Pr(g=k \mid X=x) &= \frac{\exp\left(\beta_k^T x\right)}{1+\sum_{l=1}^{K-1}\exp\left(\beta_l^T x\right)}, \quad k=1,\cdots,K-1\\ \Pr(g=K \mid X=x) &= \frac{1}{1+\sum_{l=1}^{K-1}\exp\left(\beta_l^T x\right)}\end{aligned}$$

1. Objective function

$$\hat{\beta} = \arg\max_{\beta}\sum_{i=1}^{N}\log\Pr_{\beta}\left(y_i \mid x_i\right)$$

2. Parameters estimation (binary case)

$$p(x,\beta)=\Pr{}_\beta(y=1\mid x)=\frac{\exp\left(\beta^T x\right)}{1+\exp\left(\beta^T x\right)}$$
$$1-p(x,\beta)=\Pr_\beta(y=0\mid x)=\frac{1}{1+\exp\left(\beta^T x\right)}$$
$$l(\beta)=\sum_{i=1}^{N} y_i\log p\left(x_i,\beta\right)+\left(1-y_i\right)\log\left(1-p\left(x_i,\beta\right)\right)$$

$$\begin{aligned}p(x,\beta)=&\ \frac{\exp\left(\beta^T x\right)}{1+\exp\left(\beta^T x\right)}\\ l(\beta)=&\sum_{i=1}^{N} y_i\log p\left(x_i,\beta\right)+\left(1-y_i\right)\log\left(1-p\left(x_i,\beta\right)\right)\\ =&\sum_{i=1}^{N}\left\{y_i\beta^T x_i-\log\left(1+e^{\beta^T x_i}\right)\right\}\end{aligned}$$

Set the first-order derivative of $l(\beta)$ w.r.t β as 0

$$\frac{\partial l(\beta)}{\partial\beta}=\sum_{i=1}^{N} x_i\left(y_i-p\left(x_i,\beta\right)\right)=0$$

3. Use Newton-Raphson algorithm to solve this equation:

$$\frac{\partial l(\beta)}{\partial\beta}=\sum_{i=1}^{N} x_i\left(y_i-p\left(x_i,\beta\right)\right)=0$$

Newton-Raphson algorithm: $f(\beta)=\frac{\partial l(\beta)}{\partial\beta}=0$

$$\begin{aligned}\beta_1&=\beta_0-\frac{f\left(\beta_0\right)}{f'\left(\beta_0\right)}\\ &\vdots\\ \beta_{n+1}&=\beta_n-\frac{f\left(\beta_n\right)}{f'\left(\beta_n\right)}\end{aligned}$$

4 PCA, LDA, Local linear embedding

4.1 PCA

1. Maximum variance direction: when $\mathbf{X}$ is decentralized

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i\right)^2=\frac{1}{n}\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

2. Minimum reconstruction error

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i-\left(\mathbf{v}^T\mathbf{x}_i\right)\mathbf{v}\right\|^2\\
&=\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top-\left(\mathbf{v}^\top\mathbf{x}_i\right)\cdot\mathbf{v}^\top\right)\left(\mathbf{x}_i-\left(\mathbf{v}^\top\mathbf{x}_i\right)\cdot\mathbf{v}\right)\\
&=\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top\mathbf{x}_i+\left(\mathbf{v}^\top\mathbf{x}_i\right)^2\mathbf{v}^\top\mathbf{v}-2\left(\mathbf{v}^\top\mathbf{x}_{\,i}\right)\mathbf{v}^{\,\top}\mathbf{x}^i\right)\\
&=\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top\mathbf{x}_i-\left(\mathbf{v}^\top\mathbf{x}_i\right)^2\right)\\
&\iff-\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{v}^\top\mathbf{x}_i\right)^2
\end{aligned}$$

3. Maximum variance direction: when $\mathbf{X}$ is decentralized

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{v}^T\mathbf{x}_i\right)^2&=\frac{1}{n}\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}\\
&\max_{\mathbf{v}}\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}\\
&\text{s.t. }\mathbf{v}^T\mathbf{v}=1
\end{aligned}$$

Lagrangian form:

$$\begin{aligned}
\mathcal{L}_{\mathbf{v}}&=\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}+\lambda\left(1-\mathbf{v}^T\mathbf{v}\right)\\
\frac{\partial\mathcal{L}_{\mathbf{v}}}{\partial\mathbf{v}}&=\mathbf{X}\mathbf{X}^T\mathbf{v}-\lambda\mathbf{v}=\mathbf{0}
\end{aligned}$$

Obtain $\mathbf{v}$ using eigen decomposition on covariance matrix $\mathbf{X}\mathbf{X}^T$

4. Kernel PCA:

$$\begin{aligned}
&\phi(\mathbf{X})=\left[\phi\left(\mathbf{x}_1\right),\phi\left(\mathbf{x}_2\right),\ldots,\phi\left(\mathbf{x}_n\right)\right]\\
&\mathbf{v}=\phi(\mathbf{X})\boldsymbol{\alpha}\\
\mathbf{X}\mathbf{X}^T\mathbf{v}=&\lambda\mathbf{v}\Rightarrow\phi(\mathbf{X})\phi(\mathbf{X})^T\phi(\mathbf{X})\boldsymbol{\alpha}=\lambda\phi(\mathbf{X})\boldsymbol{\alpha}\\
&\phi(\mathbf{X})\phi(\mathbf{X})^T\phi(\mathbf{X})\boldsymbol{\alpha}=\lambda\phi(\mathbf{X})\boldsymbol{\alpha}\\
\Longrightarrow\quad&\phi(\mathbf{X})^T\phi(\mathbf{X})\phi(\mathbf{X})^T\phi(\mathbf{X})\boldsymbol{\alpha}=\lambda\phi(\mathbf{X})^T\phi(\mathbf{X})\boldsymbol{\alpha}\\
\Longrightarrow\quad&\mathbf{K}\mathbf{K}\boldsymbol{\alpha}=\lambda\mathbf{K}\boldsymbol{\alpha}\\
\Longrightarrow\quad&\mathbf{K}\boldsymbol{\alpha}=\lambda\boldsymbol{\alpha}
\end{aligned}$$

4.2 Linear Discriminant Analysis

1. According to the Bayes optimal classification, the posteriors is needed. post probability : $\Pr(G\mid X)$

2. Assume:

$$f_k(x)\ (\mathbf{p}(\mathbf{x}|\mathbf{k}))\longrightarrow\text{ condition-density of x in class G = k}$$

$$\pi_k\ \ (\mathbf{p}(\mathbf{k}))\longrightarrow\text{ prior probability of class k, with }\sum_{k=1}^{K}\pi_k=1$$

3. Multivariate Gaussian density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}\left|\Sigma_k\right|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

4. Many techniques are based on models for the class densities:
   (a) Linear and quadratic discriminant analysis use Gaussian densities
   (b) more flexible mixtures of Gaussians allow for nonlinear decision boundaries
   (c) general nonparametric density estimates for each class density allow the most flexibility

5. Bayes theorem give us the discriminant:

$$\Pr(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

6. Comparing two classes k and l, assume $\Sigma_k = \Sigma, \forall k$

$$\begin{aligned}\log \frac{\Pr(G = k \mid X = x)}{\Pr(G = l \mid X = x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l} \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \frac{1}{2}\mu_l^T \Sigma^{-1} \mu_l \\ &+ x^T \Sigma^{-1} (\mu_k - \mu_l)\end{aligned}$$

$$\log \frac{\Pr(G = k \mid X = x)}{\Pr(G = l \mid X = x)} = \log \frac{\pi_k}{\pi_l} - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \frac{1}{2}\mu_l^T \Sigma^{-1} \mu_l + x^T \Sigma^{-1} (\mu_k - \mu_l)$$

7. The above function implies that the boundary of class k and l is linear in x.

8. Linear Discriminant Function

$$\delta_k(x) = \log \Pr(G = k \mid X = x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

$$G(x) = \arg\max_k \delta_k(x)$$

   In practice we do not know the parameters of the Gaussian distributions, so we need to estimate $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}$ using our training data.

9. Classification: $\mathbf{w}^T\mathbf{x} = c, \quad \mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

   LDA rule: $G(x) = \arg\max_k \delta_k(x)$
   Decision boundary: $\{\text{x} \mid \delta_k(x) = \delta_l(x)\}$
   In binary case, we assume $\pi_1 = \pi_2$

$$\delta_1(x) = \delta_2(x)$$

$$x^T \Sigma^{-1} \mu_1 - \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 = x^T \Sigma^{-1} \mu_2 - \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2$$

$$x^T \left(\Sigma^{-1} (\mu_1 - \mu_2)\right) = \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 - \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2$$

$$w^T x = c$$

10. Dimensionality Reduction: $\hat{x} = \mathbf{v}^T\mathbf{x}, \quad \mathbf{v} \propto \mathbf{S}_W^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)$

$$\begin{aligned}
J(\mathbf{v}) &= \frac{\left(\mathbf{v}^T\boldsymbol{\mu}_1 - \mathbf{v}^T\boldsymbol{\mu}_2\right)^2}{\boldsymbol{\sigma}_1^2 + \boldsymbol{\sigma}_2^2} \\
&= \frac{\left(\mathbf{v}^T\boldsymbol{\mu}_1 - \mathbf{v}^T\boldsymbol{\mu}_2\right)^2}{\sum_{i=1}^{n_1}\left(\mathbf{v}^T\mathbf{x}_{1,i} - \mathbf{v}^T\boldsymbol{\mu}_1\right)^2 + \sum_{i=1}^{n_2}\left(\mathbf{v}^T\mathbf{x}_{2,i} - \mathbf{v}^T\boldsymbol{\mu}_2\right)^2} \\
&= \frac{\mathbf{v}^T\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^T\mathbf{v}}{\mathbf{v}^T\left(\sum_{i=1}^{n_1}\left(\mathbf{x}_{1,i} - \boldsymbol{\mu}_1\right)\left(\mathbf{x}_{1,i} - \boldsymbol{\mu}_1\right)^T + \sum_{i=1}^{n_2}\left(\mathbf{x}_{2,i} - \boldsymbol{\mu}_2\right)\left(\mathbf{x}_{2,i} - \boldsymbol{\mu}_2\right)^T\right)\mathbf{v}} \\
&= \frac{\mathbf{v}^T\mathbf{S}_B\mathbf{v}}{\mathbf{v}^T\mathbf{S}_W\mathbf{v}}
\end{aligned}$$

Object Function:

$$\max_{\mathbf{v}} \frac{\mathbf{v}^T\mathbf{S}_B\mathbf{v}}{\mathbf{v}^T\mathbf{S}_W\mathbf{v}}$$

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_B\mathbf{v}}{\mathbf{v}^T\mathbf{S}_W\mathbf{v}}, \frac{\partial J(\mathbf{v})}{\partial \mathbf{v}} = 0 \Longrightarrow \left(\mathbf{v}^T\mathbf{S}_B\mathbf{v}\right)\mathbf{S}_W\mathbf{v} = \left(\mathbf{v}^T\mathbf{S}_W\mathbf{v}\right)\mathbf{S}_B\mathbf{v}$$

$$\mathbf{S}_B = \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^T, \text{ so } \mathbf{S}_B\mathbf{v} \text{ is in the direction of } \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)$$

Drop the scale factors $\left(\mathbf{v}^T\mathbf{S}_B\mathbf{v}\right)$ and $\left(\mathbf{v}^T\mathbf{S}_W\mathbf{v}\right)$

$$\mathbf{S}_W\mathbf{v} = \mathbf{S}_B\mathbf{v}$$

$$\mathbf{v} \propto \mathbf{S}_W^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)$$

11. Virtues and Failings of LDA:

- Simple prototype (centroid) classifier: new observation classified into the class with the closest centroid: But uses Mahalonobis distance. Single prototype per class may not be insufficient
- Simple decision rules based on linear decision boundaries: Linear boundaries ? not true in many cases. May estimate quadratic decision boundary(QDA)
- Estimated Bayes classifier for Gaussian class conditionals: But data might not be Gaussian

4.2.1 Quadratic Discriminant Analysis

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T\Sigma_k^{-1}(x - \mu_k) + \log\pi_k$$

When the covariances of class k and l are different, this is the Quadratic Discriminant Function. The decision boundary is described by a quadratic equation

$$\{x : \delta_k(x) = \delta_l(x)\}$$

4.2.2 Regularized Discriminant Analysis

a method between LDA and QDA.

1. Shrink the separate covariances of QDA towards a common covariance as in LDA.
$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}, \quad \alpha \in [0,1]$$

2. or toward the scalar covariance
$$\hat{\Sigma}_k(\gamma) = \gamma\hat{\Sigma}_k + (1-\gamma)\hat{\sigma}^2\mathrm{I}, \quad \gamma \in [0,1]$$

   Example. Test and training errors using regularized discriminant analysis with a series of values of $\alpha \in [0,1]$. The optimum for the test data occurs around 0.9, close to quadratic discriminant analysis

4.2.3 Reduced Rank LDA

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k$$

The eigen-decomposition for each $\hat{\Sigma}_k = U_k D_k U_k^T$ where U_k is $p \times p$ orthonormal, and D_k is a diagonal matrix of positive eigenvalues d_{kl}. So the ingredients for $\delta_k(x)$ are:

$$(x-\hat{\mu}_k)^T\hat{\Sigma}_k^{-1}(x-\hat{\mu}_k) = \left[(x-\hat{\mu}_k)^T U_k\right] D_k^{-1}\left[U_k^T(x-\hat{\mu}_k)\right]$$
$$\log\left|\hat{\Sigma}_k\right| = \sum_l \log d_{kl}$$

For QDA, $\hat{\Sigma}_k = U_k D_k U_k^T$

For LDA, $\hat{\Sigma} = UDU^T$

$$(x-\hat{\mu}_k)^T\hat{\Sigma}^{-1}(x-\hat{\mu}_k) = \left[U^T(x-\hat{\mu}_k)\right]^T D^{-1}\left[U^T(x-\hat{\mu}_k)\right]$$
$$= \left\|D^{-\frac{1}{2}}U^Tx - D^{-\frac{1}{2}}U^T\hat{\mu}_k\right\|^2$$
$$x^* = D^{-\frac{1}{2}}U^Tx \quad \hat{\mu}_k^* = D^{-\frac{1}{2}}U^T\hat{\mu}_k$$
$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) = -\frac{1}{2}\|x^* - \hat{\mu}_k^*\|^2$$

The matrix of samples $X^* \in \mathcal{R}^{p\times N}$
The matrix of centroids $\hat{U}^* \in \mathcal{R}^{p\times K}$
Perform PCA on $\hat{U}^*$ and obtain projection matrix V^*
projected samples $\tilde{X} = V^{*^T}X^*$
projected mean vectors $\tilde{U} = V^{*^T}\hat{U}^*$

4.2.4 Flexible Discriminant Analysis

1. Core: LDA in enlarged space of predictors

2. Suppose $\theta : G \mapsto R^1$ is a function that assigns scores to the classes, such that the transformed class labels are optimally predicted by linear regression on X.

3. FDA amounts to LDA in this enlarged space Training data $\{(x_i, g_i), i = 1, 2, \ldots N\}$

4. Objective Function $\min_{\theta,\beta} \sum_i \left(\theta(g_i) - x_i^T\beta\right)^2$

5. More generally, we define L independent scores for class labelling $\theta_1, \theta_2, \cdots, \theta_L$ and corresponding linear maps $\eta_l(X) = X^T\beta_l, \quad l = 1, 2, \cdots, L$

6. Objective Function

$$ASR = \frac{1}{N}\sum_{l=1}^{L}\sum_{i}\left(\theta_l(g_i) - x_i^T\beta_l\right)^2$$

4.2.5 Penalized Discriminant Analysis

1. Core: Fit LDA model with penalized coefficient

2. PDA is a regularized discriminant analysis on enlarged set of predictors via a basis expansion

$$\text{ASR}\left(\{\theta_l, \beta_l\}_{l=1}^{L}\right) = \frac{1}{N}\sum_{l=1}^{L}\left[\sum_{i=1}^{N}\left(\theta_l(g_i) - h^T(x_i)\beta_l\right)^2 + \lambda\beta_l^T\Omega\beta_l\right]$$

Ω depends on the problem.

3. PDA enlarge the predictors to $h(x)$

4. Use LDA in the enlarged space, with the penalized Mahalanobis distance:

$$D(x, \mu) = (h(x) - h(\mu))^T\left(\Sigma_W + \lambda\Omega\right)^{-1}(h(x) - h(\mu))$$

with Σ_W being within-class covariance matrix of $h(x_i)$

4.2.6 Mixture Discriminant Analysis

1. Core: Model each class by a mixture of Gaussians with different centroids, all sharing same covariance matrix

2. The class conditional densities modeled as mixture of Gaussians

3. Possibly different numbers of components in each class

4. Estimate the centroids and mixing proportions in each subclass by max joint likelihood $p(G, X)$

5. EM algorithm for MLE

6. A Gaussian mixture model for the k-th class
LDA: $p(X \mid G = k) = f_k(X) = \phi(X; \mu_k, \Sigma)$
MDA: $p(X \mid G = k) = \sum_{j=1}\pi'_{kj}\phi(X; \mu_{kj}, \Sigma)$

7. The Posterior
LDA:

$$p(G = k \mid X = x) = \frac{\pi_k\phi(X; \mu_k, \Sigma)}{\sum_l \pi_l\phi(X; \mu_l, \Sigma)}$$

MDA:

$$p(G = k \mid X = x) = \frac{\pi_k\sum_{j=1}\pi'_{kj}\phi(X; \mu_{kj}, \Sigma)}{\sum_l \pi_l\sum_{j=1}\pi'_{lj}\phi(X; \mu_{lj}, \Sigma)}$$

4.3 Local linear embedding

1. Local linear embedding tries to preserve the local affine structure of the high dimensional data.

2. Each data point is approximated by a linear combination of neighboring points. Then a lower dimensional representation is constructed that best preserves these local approximations.

3. On given set, obtain local relationship w

$$\begin{array}{ll} \min_{\mathbf{W}} & \left\|\mathbf{x}_i - \sum_{j \in N(i)} w_{ij} \mathbf{x}_j\right\|^2 \\ \text{s.t.} & \sum_j w_{ij} = 1 \end{array}$$

4. On new set transfer W maintain local relationship

$$\begin{array}{ll} \min_{\mathbf{Y}} & \left\|\mathbf{y}_i - \sum_{j \in N(i)} w_{ij} \mathbf{y}_j\right\|^2 \\ \text{s.t.} & \mathbf{Y}^T \mathbf{Y} = \mathbf{I} \end{array}$$

5 K-means, GMM, EM algorithm

5.1 EM algorithm

$$\left.\begin{array}{l} L(\boldsymbol{\theta}) = \log p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \\ L(\boldsymbol{\theta}) = \log p(\mathbf{x}; \boldsymbol{\theta}) \end{array}\right\} \text{Latent variable } \mathbf{Z}$$

5.1.1 Discriminative model

$$\begin{aligned} L(\boldsymbol{\theta}) &= \log p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \\ &= \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \\ &= \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta})}{p(\mathbf{z} \mid \mathbf{x}, \mathbf{y}; \boldsymbol{\theta})} \\ &= \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log \left[\frac{p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta})}{p(\mathbf{z} \mid \mathbf{x}, \mathbf{y}; \boldsymbol{\theta})} \frac{q(\mathbf{z} \mid \mathbf{x}, \mathbf{y})}{q(\mathbf{z} \mid \mathbf{x}, \mathbf{y})}\right] \\ &= \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta})}{q(\mathbf{z} \mid \mathbf{x}, \mathbf{y})} + \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log \frac{q(\mathbf{z} \mid \mathbf{x}, \mathbf{y})}{p(\mathbf{z} \mid \mathbf{x}, \mathbf{y}; \boldsymbol{\theta})} \\ &= l(\boldsymbol{\theta}, q) + \mathrm{KL}[q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \| p(\mathbf{z} \mid \mathbf{x}, \mathbf{y}; \boldsymbol{\theta})] \end{aligned}$$

E-step: given model parameters θ^t from the t-th iteration

$$q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) = p\left(\mathbf{z} \mid \mathbf{x}, \mathbf{y}; \boldsymbol{\theta}^t\right)$$

M-step: lower bound $l(\boldsymbol{\theta}, q)$ can be written as

$$Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^t\right) = \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \log p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta})$$

The derivative of $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^t\right)$ w.r.t. $\boldsymbol{\theta}$ can be written as

$$\frac{\partial Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^t\right)}{\partial \boldsymbol{\theta}} = \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}, \mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta})$$

5.1.2 Generative model

$$\begin{aligned}
L(\boldsymbol{\theta}) &= \log p(\mathbf{x};\boldsymbol{\theta}) \\
&= \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log p(\mathbf{x};\boldsymbol{\theta}) \\
&= \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log \frac{p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})}{p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})} \\
&= \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log \left[\frac{p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})}{p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})}\frac{q(\mathbf{z}\mid\mathbf{x})}{q(\mathbf{z}\mid\mathbf{x})}\right] \\
&= \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log \frac{p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z}\mid\mathbf{x})} + \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log \frac{q(\mathbf{z}\mid\mathbf{x})}{p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})} \\
&= l(\boldsymbol{\theta}, q) + \mathrm{KL}[q(\mathbf{z}\mid\mathbf{x}) \| p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})] \\
\mathbf{E}-\mathbf{Step}: &\quad q(\mathbf{z}\mid\mathbf{x}) = p\left(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta}^t\right) \quad \hat{\gamma}_i = \frac{\hat{\pi}^2 \phi_{\hat{\theta}_2}(x_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(x_i) + \hat{\pi}\phi_{\hat{\theta}_2}(x_i)} \\
\mathbf{M}-\mathbf{step}: &\quad Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^t\right) = \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})
\end{aligned}$$

5.2 K-means

1. The within-cluster scatter can be written as

$$\begin{aligned}
W(C) &= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} d\left(x_i, x_{i'}\right) \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} \left\|x_i - x_{i'}\right\|^2 \\
&= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\|x_i - \bar{x}_k\right\|^2
\end{aligned}$$

   Where N_k is the sample number of k-th class

2. In each iteration of K-means:

   - Expectation-step: estimate the hidden variable

$$C(i) = k \quad \text{if } k = \arg\min_k \|x_i - \bar{x}_k\| 2$$

     or

$$\gamma_i(k) = \begin{cases} 1, & \text{if } k = \arg\min_k \left(x_i - \mu_k\right)^2 \\ 0, & \text{otherwise} \end{cases}$$

   - Maximization-step: maximum likelihood estimation

$$\bar{x}_k = \frac{\sum_{C(i)=k} x_i}{N_k}$$

     or

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_i(k) x_i}{\sum_{i=1}^{N} \gamma_i(k)}$$

5.3 GMM

1. We assume $\{x_1, x_2, \ldots, x_N\}$ follows Gaussian Mixture Model (GMM)

2. GMM parameters: $\boldsymbol{\theta} = \{\pi_1, \mu_1, \sigma_1; \pi_2, \mu_2, \sigma_2; \ldots; \pi_K, \mu_K, \sigma_K\}$

$$p(x_1, x_2, \ldots, x_N \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(x_i \mid \boldsymbol{\theta}), \text{ in which } p(x_i \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} p(x_i \mid \boldsymbol{\theta}_k)\, p(\boldsymbol{\theta}_k) = \sum_{k=1}^{K} p(x_i \mid \mu_k, \sigma_k)\, \pi_k$$

3. Hidden variable $\gamma_i(k)$: the probability that the i-th sample belongs to the k-th Gaussian model
   Due to hidden variable, we use EM algorithm to estimate $\gamma_i(k)$ and $\boldsymbol{\theta} = \{\pi_1, \mu_1, \sigma_1; \pi_2, \mu_2, \sigma_2; \ldots; \pi_K, \mu_K, \sigma_K\}$

4. GMM is probabilistic clustering or soft clustering. In each iteration:

   - Expectation-step: estimate the hidden variable

$$\gamma_i(k) = \frac{\pi_k\, p(x_i \mid \boldsymbol{\theta}_k)}{\sum_{k'=1}^{K} \pi_{k'}\, p(x_i \mid \boldsymbol{\theta}_{k'})}$$

E-step: $q(\mathbf{z} \mid \mathbf{x}) = p\left(\mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta}^t\right)$

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(x_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(x_i) + \hat{\pi}\phi_{\hat{\theta}_2}(x_i)}$$

   - Maximization-step: maximum likelihood estimation

$$\pi_k = \frac{\sum_{i=1}^{N} \gamma_i(k)}{N} \qquad \mu_k = \frac{\sum_{i=1}^{N} \gamma_i(k) x_i}{\sum_{i=1}^{N} \gamma_i(k)} \qquad \sigma_k = \frac{\sum_{i=1}^{N} \gamma_i(k)\,(x_i - \mu_k)^2}{\sum_{i=1}^{N} \gamma_i(k)}$$

$$\text{M-step: } Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^t\right) = \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}) \log p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$$

$$\frac{\partial Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^t\right)}{\partial \boldsymbol{\theta}} = \sum_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}) \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$$

6 Kernel Smoother

6.1 KNN

$$\hat{f}(x_0) = \operatorname{Ave}(y_i \mid x_i \in N_k(x_0))$$

6.2 Kernel Smoother

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i)\, y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{h_\lambda(x_0)}\right)$$

1. $h_\lambda\left(x_0\right)$: width function that determines the width of the neighborhood at x_0 .
2. $h_\lambda\left(x_0\right) = \lambda$: large λ averages over more observations, which implies lower variance but higher bias

7 SVM

1. Motivation:
   - Geometric: Maximizing Margin
   - Kernel Methods: Making nonlinear decision boundaries linear
2. Maximize Margin?:
   (a) Intuitively this feels safest.
   (b) If we've made a small error in the location of the boundary, this gives us least chance of causing a misclassification.
   (c) There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
   (d) Empirically it works very well.
3. Margin and Plane:
   (a) Plus-plane $\left\{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = +1\right\}$
   (b) Minus-plane $\left\{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = -1\right\}$
   (c) Claim: The vector $\boldsymbol{w}$ is perpendicular to the Plus Plane.

Margin

The vector w is perpendicular to the Plus Plane. Let $\mathbf{x}^{-}$be any point on the minus plane. Let x^{+}be the closest plus-plane-point to x^{-}. Claim: $\mathbf{x}^{+} = \mathbf{x}^{-} + \lambda\mathbf{w}$ for some value of λ. Let M = Margin Width. In sum, we have

$$\begin{aligned}
\mathbf{w}^T\mathbf{x}^{+} + b &= +1\\
\mathbf{w}^T\mathbf{x}^{-} + b &= -1\\
\mathbf{x}^{+} &= \mathbf{x}^{-} + \lambda\mathbf{w}\\
\left\|\mathbf{x}^{+} - \mathbf{x}^{-}\right\| &= M
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbf{w}^T\left(\mathbf{x}^{-} + \lambda\mathbf{w}\right) + b = 1\\
&\mathbf{w}^T\mathbf{x}^{-} + b + \lambda\mathbf{w}^T\mathbf{w} = 1\\
&-1 + \lambda\mathbf{w}^T\mathbf{w} = 1
\end{aligned}$$

$$\lambda = \frac{2}{\mathbf{w}^T\mathbf{w}}$$

$$\begin{aligned}
M &= \left\|\mathbf{x}^{+} - \mathbf{x}^{-}\right\|\\
&= \lambda\|\mathbf{w}\| = \lambda\sqrt{\mathbf{w}^T\mathbf{w}}\\
&= \frac{2\sqrt{\mathbf{w}^T\mathbf{w}}}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}} = \frac{2}{\|\mathbf{w}\|}
\end{aligned}$$

4. Large-margin Decision Boundary

   (a) Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be our data set and let $y_i \in \{1, -1\}$ be the class label of $\mathbf{x}_i$ The decision boundary should classify all points correctly $\Rightarrow$

   $$y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, \quad \forall i.$$

   (b) The decision boundary can be found by solving the following constrained optimization problem

   $$\begin{array}{ll}
   \min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2\\
   \text{s.t.} & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, \quad \forall i.
   \end{array}$$

   (c) This is a constrained optimization problem. Solving it requires some new tools.

   $$\begin{array}{ll}
   \min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2\\
   \text{s.t.} & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, \quad \forall i.
   \end{array}$$

   (d) The Lagrangian is

   $$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N}\alpha_i\left(1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right)\right)$$

(e) Setting the gradient of $\mathcal{L}$ w.r.t. w and $\mathbf{b}$ to zero, we have

$$\mathbf{w}+\sum_{i=1}^{N}\alpha_i(-y_i)\mathbf{x}_i=0\Rightarrow\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i$$

$$\sum_{i=1}^{N}\alpha_i y_i=0$$

(f) The Karush-Kuhn-Tucker (KKT) conditions,

$$\alpha_{\mathrm{i}}\left[y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)-1\right]=0,\quad\forall i.$$

- If $\alpha_i>0$, then $y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)=1$, or in other word, x_i is on the boundary of the slab;
- If $y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)\neq 1,\mathbf{x}_i$ is not on the boundary of the slab, and $\alpha_i=0$.

(g) If we substitute $\mathrm{w}=\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i$ to $\mathcal{L}$, we have

$$\begin{aligned}\mathcal{L}&=\frac{1}{2}\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i^T\sum_{j=1}^{N}\alpha_j y_j\mathbf{x}_j+\sum_{i=1}^{N}\alpha_i\left(1-y_i\left(\sum_{j=1}^{N}\alpha_j y_j\mathbf{x}_j^T\mathbf{x}_i+b\right)\right)\\&=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\mathbf{x}_i^T\mathbf{x}_j+\sum_{i=1}^{N}\alpha_i-\sum_{i=1}^{N}\alpha_i y_i\sum_{j=1}^{N}\alpha_j y_j\mathbf{x}_j^T\mathbf{x}_i-b\sum_{i=1}^{N}\alpha_i y_i\\&=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\mathbf{x}_i^T\mathbf{x}_j+\sum_{i=1}^{N}\alpha_i\end{aligned}$$

This is a function of α_i only

$$\begin{array}{ll}\max_{\alpha_i} & \sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1,j=1}^{N}\alpha_i\alpha_j y_i y_j\mathbf{x}_i^T\mathbf{x}_j,\\ \text{s.t.} & \alpha_i\geq 0,\quad\sum_{i=1}^{N}\alpha_i y_i=0.\end{array}$$

5. Characteristics of the Solution

- $\mathbf{x}_i$ with non-zero α_i are called support vectors (SV)
- Let $t_j(j=1,\ldots,S)$ be the indices of the S support vectors. We can write $\mathbf{w}=\sum_{j=1}^{S}\alpha_{t_j}y_{t_j}\mathbf{x}_{t_j}$
- Note that for support vectors $\mathbf{X}_i,y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)=1$

$$b=\frac{1}{S}\sum_{j=1}^{S}\left(y_{t_j}-\mathbf{w}^T\mathbf{x}_{t_j}\right)$$

- For testing with a new data $\mathbf{z}$

$$\mathbf{w}^T\mathbf{z}+b=\sum_{j=1}^{S}\alpha_{t_j}y_{t_j}\left(\mathbf{x}_{t_j}^T\mathbf{z}\right)+b$$

6. Learning Maximum Margin with Noise

$$\begin{aligned}
&\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\
&\text{s.t. } y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \forall i, \\
&\qquad \xi_i \geq 0, \quad \forall i
\end{aligned}$$

The Lagrange function:

$$\begin{aligned}
L_p(\mathbf{w}, b, \xi_i) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \\
-\sum_{i=1}^{N}\alpha_i\left[y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - (1 - \xi_i)\right]
\end{aligned}$$

Setting the respective derivatives to zero, we get

$$\begin{aligned}
&\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i \\
&\sum_{i=1}^{N}\alpha_i y_i = 0 \\
&\alpha_i = C - \mu_i, \forall i \\
&\alpha_i, \mu_i, \xi_i \geq 0, \forall i
\end{aligned}$$

Dual QP

$$\begin{aligned}
&\max_{\alpha_i}\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j Q_{ij} \text{ where } Q_{ij} = y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\
&\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \forall i. \quad \sum_{i=1}^{N}\alpha_i y_i = 0.
\end{aligned}$$

8 Piecewise Polynomials and Splines

Assume x is 1 -dim feature

8.1 Definition

1. Linear basis expansion

$$f(x) = \sum_{d=1}^{D}\beta_d h_d(x)$$

2. Some basis functions that are widely used

$$\begin{aligned}
&h_d(x) = x \\
&h_d(x) = x^p \\
&h_d(x) = \log(x) \\
&h_d(x) = \sin(d\pi x) \text{ or } \cos(d\pi x) \\
&h_d(x) = I(L \leq x \leq U)
\end{aligned}$$

8.2 Piecewise polynomials

1. An order- M spline with knots $\xi_j, j = 1, \dots, K$ is a piecewise-polynomial of order M, and has continuous derivatives up to order $M-2$.

2. The general form of truncated power basis set is:

$$
\begin{aligned}
h_j(X) &= X^{j-1}, \quad j = 1, \dots, M \\
h_{M+l}(X) &= (X - \xi_l)_+^{M-1}, \quad l = 1, \dots, K
\end{aligned}
$$

8.3 Natural Cubic Splines

1. The behavior of polynomials fit to data tends to be erratic near the boundaries and beyond the boundary knots.

2. Natural cubic spline adds additional constraints, namely that **the function is linear beyond the boundary knots**.

Example. Natural Cubic Spline

$$
\begin{aligned}
N_1(x) = 1, N_2(x) = x, N_{k+2}(x) &= d_k(x) - d_{K-1}(x) \\
d_k(x) &= \frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}
\end{aligned}
$$

When $K = 2, \quad N_3(x) = d_1(x) - d_1(x) = 0$

$$
\begin{aligned}
\text{When } K = 3, N_3(x) = d_1(x) - d_2(x) &= \frac{(x-\xi_1)_+^3 - (x-\xi_3)_+^3}{\xi_3 - \xi_1} - \frac{(x-\xi_2)_+^3 - (x-\xi_3)_+^3}{\xi_3 - \xi_2} \\
N_4(x) = d_2(x) - d_2(x) &= 0
\end{aligned}
$$

$$
f(x) = \beta_1 N_1(x) + \beta_2 N_2(x) + \beta_3 N_3(x) + \beta_4 N_4(x)
$$

linear beyond boundary knots ($x \le \xi_1, x \ge \xi_3$).

8.4 B-splines

1. Let $\xi_0 < \xi_1$ and $\xi_K < \xi_{K+1}$ be two boundary knots

2. The augmented knot sequence τ :

$$
\begin{aligned}
&\tau_1 \le \tau_2 \le \cdots \le \tau_M \le \xi_0 \\
&\tau_{j+M} = \xi_j, \quad j = 1, \cdots, K \\
&\xi_{K+1} \le \tau_{K+M+1} \le \tau_{K+M+2} \le \cdots \le \tau_{K+2M}
\end{aligned}
$$

$$
\tau_1, \tau_2, \dots, \tau_M, \underbrace{\boxed{\tau_{M+1}, \ \dots, \ \tau_{M+K}}}_{\xi_1, \dots, \xi_K}, \tau_{M+K+1}, \dots, \tau_{K+2M}
$$

3. $B_{i,m}(x)$, the i-th B-spline basis function of order m for the knot sequence $\tau, \quad m < M$.

$$B_{i,1}(x) = \begin{cases} 1 & \tau_i \leq x < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \cdots, K + 2M - m$$

$$B_{i,m}(x) = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

   The spline function of order m on a given set of knots can be expressed as a linear combination of B-splines: $\sum_i \alpha_i B_{i,m}(x)$

9 PageRank

1. Motivation:

   - We suppose that we have N web pages and wish to rank them in terms of importance.
   - The PageRank algorithm considers a webpage to be important if many other webpages point to it.
   - The linking webpages that point to a given page are not treated equally: the algorithm also takes into account both the importance (PageRank) of the linking pages and the number of outgoing links that they have.

2. Modeling:

   - Let $L_{ij} = 1$ if page j points to page i, and zero otherwise.
   - Let $c_j = \sum_{i=1} L_{ij}$ equal the number of pages pointed to by page j (number of outlinks).
   - Then the Google PageRank p_i are defined by the recursive relationship

$$p_i = (1 - d) + d \sum_{j=1}^{N} \frac{L_{ij}}{c_j} p_j$$

   - d is a positive constant (say $d = 0.85$). In matrix form

$$\mathbf{p} = (1 - d)\mathbf{1} + d\mathbf{L}\mathbf{D}_c^{-1}\mathbf{p}$$

   - $\mathbf{1}$ is a vector of N ones; $\mathbf{D}_c = \operatorname{diag}(c)$
   - - Introducing the normalization $\mathbf{1}^T\mathbf{p} = N$ (i.e., the average PageRank is 1), rewrite the above equation

$$\mathbf{p} = \left[(1 - d)\mathbf{1}\mathbf{1}^T/N + d\mathbf{L}\mathbf{D}_c^{-1}\right]\mathbf{p} = \mathbf{A}\mathbf{p}$$

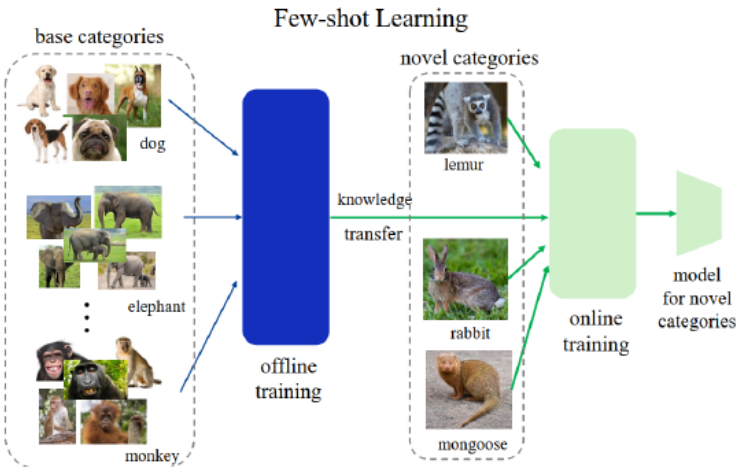
   - PageRank solution:
     Initialize $\mathbf{p}_0$

$$\mathbf{p}_k \leftarrow \mathbf{A}\mathbf{p}_{k-1} ; \quad \mathbf{p}_k \leftarrow N \frac{\mathbf{p}_k}{\mathbf{1}^T\mathbf{p}_k}$$

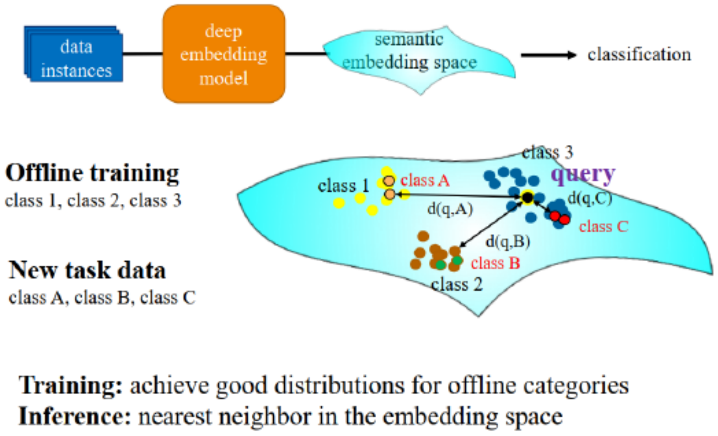
10 Few-shot learning

(metric-learning based)

1. Few-shot learning: We are given a base category set $\mathcal{C}^b$ (each base category has sufficient labeled samples), as well as a novel category set $\mathcal{C}^n$ (each novel category has only a few labeled samples). We need to learn a good classifier for the novel classes by transferring the knowledge from base classes.



2. Metric learning: Learn a semantic embedding space using a distance loss function. Compare each testing (query) data sample with training (support) data sample: training semantic embeddings.



Training: achieve good distributions for offline categories
Inference: nearest neighbor in the embedding space

3. Three models: Relation Network, Matching Network, Prototypical Network

11 Weakly-supervised learning

(Multi-instance learning)

11.1 Basis

1. Definition:

   (a) We do not lack data, but lack well-annotated data.

   (b) The annotation cost of strong annotation is very high, so weakly annotated samples are more accessible.

   (c) Given a specific task, weak annotation means lower-degree (or cheaper) annotation than regular annotation

   (d) When the noisy data is crawled from web, weakly supervised classification is also called webly supervised classification.

2. Multi-instance learning:

   (a) It is a special form of weakly supervised learning.

   (b) Training instances are arranged in sets, called bags.

   (c) A label is provided for entire bags but not for instances.
   Negative bags do not contain positive instances.
   Positive bags contain at least one positive instance. Positive bags may contain negative and positive instances.

$$\mathcal{B}_l = \{\mathbf{x}_{l,1}, \dots, \underbrace{\mathbf{x}_{l,\mathcal{B}_l}}_{y_{l,i}\,|\mathcal{B}_l|}\} \quad \begin{cases} Y_l = \sum_{i\in\{1,\dots,|\mathcal{B}_l|\}} \frac{y_{\,l,i}+1}{2} \geq 1, & \forall Y_l = 1, \\ y_{l,i} = -1, \quad \forall Y_l = -1. \end{cases}$$

3. The standard MIL assumption is too restrictive. MIL can be relaxed as:

   - A bag is positive when it contains a sufficient number of positive instances.
   - A bag is negative when it contains a certain number of negative instances.
   - Positive and negative bags differ by their positive/negative distributions.

$$\begin{cases} \sum_{i\in\{1,\dots,|\mathcal{B}_l|\}} \frac{y_{l,i}+1}{2} \geq \sigma_p\,|\mathcal{B}_l|\,, & \forall Y_l = 1 \\ \sum_{i\in\{1,\dots,|\mathcal{B}_l|\}} \frac{y_{l,i}+1}{2} \leq \sigma_n\,|\mathcal{B}_l|\,, & \forall Y_l = -1 \end{cases}$$

11.2 Methods

1. Bag-level method

   (a) These methods embed the content of bags in a single feature vector, thus transforming the problem into supervised learning.

   (b) Pros:

   - Can model distributions and relation between instances.
   - Deal with unclassifiable instances.
   - Can be faster than instance based methods.
   - Often more accurate for bag classification tasks.

   (c) Cons: Cannot be directly used for instance classification tasks.

(d) sMIL:

$$\begin{aligned}
&\min_{\mathbf{w},b,\xi_{\,i}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{l=1}^{L}\xi_{\,l}\\
&\text{s.t. } Y_l\left(\mathbf{w}^T\phi\left(\mathbf{z}_l\right)+b\right) \geq \rho-\xi_l, \quad \forall l,\\
&\xi_l \geq 0, \forall l.\\
&\rho=\sigma-(1-\sigma)=2\sigma-1
\end{aligned}$$

the margin between pos/neg samples, σ means the ratio. The averaged bag feature: $\mathbf{z}_l=\frac{1}{|\mathcal{B}_l|}\sum_{i=1}^{|\mathcal{B}_l|}\mathbf{x}_{\,l,i}$

2. Instance-level methods:

   (a) These methods try to uncover the true nature of each instance in order to make a decision on bag labels.

   (b) Pros: Can be directly used for instance classification tasks.

   (c) Cons:

   - Do not work when instances have no precise classes.
   - Usually less accurate than bag space methods.

   (d) mi-SVM:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi_{l,i},y_{l,i}} &\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{l=1}^{L}\sum_{i=1}^{|\mathcal{B}_l|}\xi_{l,i}\\
\text{s.t. } & y_{l,i}\left(\mathbf{w}^T\phi\left(\mathbf{x}_{l,i}\right)+b\right) \geq 1-\xi_{l,i}, \quad \forall l,\forall i,\\
&\xi_{l,i} \geq 0, \quad \forall l,\forall i.
\end{aligned}$$

Solve $\{\mathbf{w},b\}$ and $y_{l,i}$ alternatingly:
1. Fix $\{\mathbf{w},b\}$, update $y_{l,i}$ $\tilde{y}_{l,i}=\mathbf{w}^T\phi\left(\mathbf{x}_{l,i}\right)+b$
convert $\tilde{y}_{l,i}$ to $y_{l,i}$ based on the constraint $\begin{cases}\sum_{i\in\{1,\ldots,|\mathcal{B}_l|\}}\frac{y_{l,i}+1}{2}\geq\sigma\left|\mathcal{B}_l\right|, & \forall Y_l=1,\\ y_{l,i}=-1, \quad \forall Y_l=-1.\end{cases}$
2. Fix $y_{l,i}$, update $\{\mathbf{w},b\}$, solve standard SVM