CS4220: Machine Learning

Yuhang Tian[1]

[1]Technology University of Delft

y.tian-13@student.tudelft.nl

1. Bayesian Classifier Theorem

1.1. Decision Boundary

$$p(x|w_1)p(w_1) \gtrless p(x|w_2)p(x|w_2) \tag{1}$$

1.2. Minimizing the Classification Error Probability (Bayes Error: the minimum error for a certain number of features)

$$P_e = p(w_2)\int_{-\infty}^{x_0} p(x|w_2)dx + p(w_1)\int_{x_0}^{\infty} p(x|w_1)dx \tag{2}$$

where $p(x_0|w_1)p(w_1) = p(x_0|w_2)p(x|w_2)$

1.3. Minimizing the Average Risk

Loss Matrix

$$L = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix} \tag{3}$$

Risk Function

$$r = \lambda_{21}p(w_2)\int_{-\infty}^{x_0} p(x|w_2)dx + \lambda_{12}p(w_1)\int_{x_0}^{\infty} p(x|w_1)dx \tag{4}$$

1.4. Gaussian pdf in the l-dimensional space

$$p(x) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{l/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{5}$$

where $\Sigma = E[(x-\mu)(x-\mu)^T]$

1.5. Bayesian Classifier

discriminant functions

$$g_i(x) = ln(p(x|w_i)p(w_i)) = lnp(x|w_i) + lnp(w_i) \tag{6}$$

Normally Distributed Classifier

$$g_i(x) = -\frac{1}{2}x^T\Sigma_i^{-1}x + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x + lnp(w_i) + c_i \tag{7}$$

where $c_i = -(l/2)ln2\pi - (1/2)ln(det(\Sigma_i))$

- if l=2, corelation=0

$$g_i(x) = -\frac{1}{2\sigma_i^2}(x_1^2 + x_2^2) + \frac{1}{\sigma_i^2}(\mu_i 1x_1 + \mu_i 2x2) - \frac{1}{2\sigma_i^2}(\mu_{i1}^2 + \mu_{i2}^2) + lnp(w_i) + c_i \tag{8}$$

$g_i(x) - g_j(x) = 0$ are quadratics (i.e., ellipsoids, parabolas, hyperbolas, pairs of lines)

- if covariance matrix is the same in all classes

$$g_i(x) = w_i^T x + b \tag{9}$$

where $w_i = \Sigma^{-1}\mu_i$ and $b = lnp(w_i) - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i$

- if Diagonal covariance matrix with equal elements ($\Sigma = \sigma^2 I$)

$$g_i(x) = \frac{1}{\sigma^2}\mu_i^T x + b \tag{10}$$

2. ESTIMATION OF UNKNOWN PROBABILITY DENSITY FUNCTIONS

2.1. ML

we considered θ as an unknown parameter.

$$\hat{\theta}_{ML} = \textit{arg}\ max_\theta \prod_{k=1}^{N} p(x_k;\theta) \tag{11}$$

ML estimate of σ^2

$$\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2 \tag{12}$$

ML estimate of μ

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{13}$$

2.2. MAP

we considered θ as a random vector.

$$p(\theta|X) = \frac{p(\theta)p(X|\theta)}{P(X)} \tag{14}$$

then,

$$\hat{\theta}_{MAP} : \frac{\partial}{\partial\theta}p(\theta|X) = 0\ or\ \frac{\partial}{\partial\theta}p(X|\theta)p(\theta) = 0 \tag{15}$$

2.3. Bayesian Inference

Given the set X of the N training vectors and the *a priori information* about the pdf $p(\theta)$, the goal is to compute the conditional pdf $p(x|X)$.

$$p(x|X) = \int p(x|\theta)p(\theta|X)d\theta \tag{16}$$

with

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} \tag{17}$$

$$p(X|\theta) = \prod_{k=1}^{N} p(x_k|\theta) \tag{18}$$

3. Normal-based Classifier: Quadratic Discriminant, Linear Discriminant and Nearest Mean

Let's assume that we have two classes:

3.1. Quadratic Discriminant

by eq.7, the quadratic classifier,

$$f(x) = x^T W x + w^T + w_0 \tag{19}$$

with

$$W = \frac{1}{2}(\Sigma_2^{-1} - \Sigma_1^{-1}) \tag{20}$$

$$w = \mu_1^T \Sigma_1^{-1} - \mu_2^T \Sigma_2^{-1} \tag{21}$$

$$w_0 = -\frac{1}{2} ln(det(\Sigma_1)) - \frac{1}{2}\mu_1^T \Sigma_1^{-1} \mu_1 + lnp(y_1) + \frac{1}{2} ln(det(\Sigma_2)) + \frac{1}{2}\mu_2^T \Sigma_2^{-1} \mu_2 - lnp(y_2) \tag{22}$$

(i.e., ellipsoids, parabolas, hyperbolas, pairs of lines)

3.2. Linear Discriminant

by eq.9, the linear classifier,

$$f(x) = w^T x + w_0 \tag{23}$$

with

$$w = \Sigma^{-1}(\mu_2 - \mu_1) \tag{24}$$

$$w_0 = \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 - \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + ln\frac{p(y_1)}{p(y_2)} \tag{25}$$

Therefore, the Linear Discriminant has a **strong assumption** on that $\Sigma_1 = \Sigma_2$

3.3. Nearest Mean Classifier

by eq.10, the nearest mean classifier,

$$f(x) = w^T x + w_0 = \sigma^2(g_1(x) - g_2(x)) \tag{26}$$

with

$$w = \mu_2 - \mu_1 \tag{27}$$

$$w_0 = \frac{1}{2}\mu_1^T \mu_1 - \frac{1}{2}\mu_2^T \mu_2 + \sigma^2 ln\frac{p(y_1)}{p(y_2)} \tag{28}$$

Therefore, the Nearest Mean has a **strong assumption** on mutually uncorrelated and of the same variance ($\Sigma_1 = \Sigma_2 = \sigma^2 I$)

4. More Parametric Classifiers

4.1. Logistic Classifier

$$\begin{cases} p(y_1|x) = \frac{1}{e^{-(w^T x + w_0)} + 1} \\ p(y_2|x) = \frac{1}{e^{(w^T x + w_0)} + 1} \end{cases} \tag{29}$$

Maximize Log Likelihood

$$lnp(y|x) = \sum_{i=1}^{N} ln(\frac{1}{e^{-y_i(w^T x_i + w_0)} + 1}) \tag{30}$$

4.2. Fisher Classifier

$$y_i = w^T x_i \begin{cases} \geq 0 & \text{if class 1} \\ < 0 & \text{if class 2} \end{cases} \tag{31}$$

Minimize Square Loss

$$L(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2 \tag{32}$$

4.3. The Perception

Forward&backward propagation and update w

$$w \leftarrow w + \eta y x \tag{33}$$

5. Non-parametric Classification

In the above sections, all the classifiers are based on **Parametric Classification** method, more precisely, based on normal distribution and Bayes Theorem. In this section, it will mainly demonstrate two non-parametric classifiers - Parzen Classifier and Nearest Neighbour Classifier (Both methods are sensitive to the scaling of the features).

5.1. Parametric vs. Non-parametric

- Parametric: Assumptions can greatly simplify the learning process, but can also limit what can be learned. Algorithms that simplify the function to a known form are called parametric machine learning algorithms.
- Non-parametric: Algorithms that do not make strong assumptions about the form of the mapping function are called non-parametric machine learning algorithms. By not making assumptions, they are free to learn any functional form from the training data.

5.2. Histogram method

it's easy, so I just skip it.

5.3. Parzen Density Estimation

- Kernel

$$f(x) = \begin{cases} 0 & \text{if |r|>h} \\ \frac{1}{V} & \text{if |r|}\leq \text{ h} \end{cases} \tag{34}$$

- Parzen Classifier

$$p(z|h) = \frac{1}{n}\sum_{i=1}^{n} K(||z - x_i||, h) \tag{35}$$

- Parzen plugs in the Gaussian density:

$$p(x|w_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} N(x|x_j^{(i)}, hI) \tag{36}$$

5.4. Nearest Neighbour Classification

$$p(x) = p(x|w_m)p(w_m) = \frac{k_m}{n_m V_k} \cdot \frac{n_m}{n} \tag{37}$$

where V_k is the volume of the sphere centered at x with radius r (the distance to the k-th nearest neighbor)

6. More Non-parametric Classifiers

GitHub Markdown

6.1. SVM

By putting some constraints on the linear classifier, the VC dimension can be reduced. Why do that? Ans: When h is small, the true error is close to the apparent error

$$\begin{cases} w^T x_i + b \geq +1 & \text{for } y_i = +1 \\ w^T x_i + b \leq -1 & \text{for } y_i = -1 \end{cases} \tag{38}$$

Core Idea of SVM: Find the decision boundary, while maximize the margin

1. The above two equation can be merged into one

$$y_i(w^T x_i + b) - 1 \geq 0 \tag{39}$$

2. The distance between the two boundaries

$$maximize\ \frac{2}{||w||} \rightarrow minimize\ \frac{1}{2}||w||^2 \tag{40}$$

3. by Lagrange Multiplier

$$L = \frac{1}{2}||w||^2 - \sum \alpha_i[y_i(wx_i + b) - 1] \tag{41}$$

4. by $\frac{\partial L}{\partial w} = 0$

$$w = \sum \alpha_i y_i x_i \tag{42}$$

5. by $\frac{\partial L}{\partial b} = 0$

$$\sum \alpha_i y_i = 0 \tag{43}$$

6. put eq.42 back to eq.41

$$L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{44}$$

7. put eq.42 back to decision rule

$$\begin{cases} \sum \alpha_i y_i x_i \cdot u_i + b - 1 \geq 0 & \text{Then, +} \\ \sum \alpha_i y_i x_i \cdot u_i + b - 1 \leq 0 & \text{Then, -} \end{cases} \tag{45}$$

8. kernelize

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{46}$$

6.2. Agglomerative Hierarchical Clustering

1. Determine distances between all clusters
   - Two nearest objects in the clusters: single linkage
   - Two most remote objects in the clusters: complete linkage
   - Cluster centers: average linkage
2. Merge clusters that are closes
3. IF #clusters>1 THEN GOTO 1

- Dendrogram: Cut at "largest jump"$\rightarrow$ Clustering
- Fusion Graph: Cut at "largest drop"$\rightarrow$ Clustering

7. Regression

7.1. Intuitively Understanding

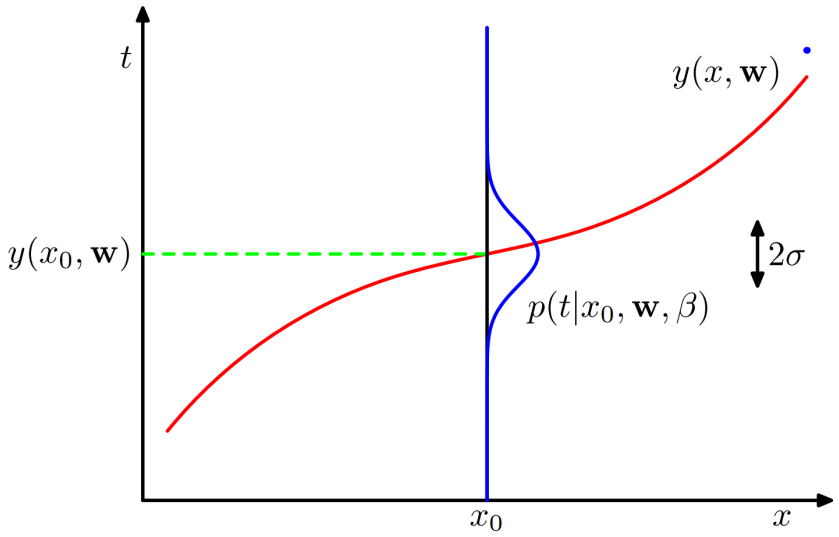


Figura 1. Regression Curve

Suppose that there is a fixed x_0 and a bunch of choices of θ (here are μ and $\sigma^2(\beta^{-1})$). If we expect the predicted value $\hat{y}(x_0, w)$ to locate on the true value t_0, the probability of this occurrence $p(t_0|x_0, w, \beta)$ should be maximized.

7.2. Maximum Likelihood Regression

$$p(t|x,w,\beta) = \prod_{n=1}^{N} N(t_n|y(x_n,w),\beta^{-1}) \tag{47}$$

Log Likelihood function

$$lnp(t|x,w,\beta) = -\frac{\beta}{2}\sum_{n=1}^{N} y(x_n,w) - {t_n}^2 + \frac{N}{2}ln\beta - \frac{N}{2}ln(2\pi) \tag{48}$$

To solve

$$\frac{\partial}{\partial w} lnp(t|x_0,w,\beta) = 0 \tag{49}$$

and

$$\frac{\partial}{\partial \beta^{-1}} lnp(t|x_0,w,\beta) = 0 \tag{50}$$

If we use linear regression, the solutions of eq.49 and eq.50

$$w_{ML} = (X^T X)^{-1} X^T Y \ \beta_{ML}^{-1} = \frac{1}{N}(w_{ML}^T X - Y) \tag{51}$$

7.3. Max a Posterior Regression

Suppose that we have some knowledge about w $N(0,\alpha I)$

$$w_{MAP} : (\prod_{i=1}^{N} p(y_i|w^T x_i,\sigma^2))p(w|0,\alpha I) \tag{52}$$

$$\frac{\partial}{\partial w}(\prod_{i=1}^{N} p(y_i|w^T x_i,\sigma^2))p(w|0,\alpha I) = 0 \rightarrow w_{MAP} = (X^T X + \frac{\sigma^2}{\alpha} I)^{-1} X^T Y \tag{53}$$

8. Regularization

8.1. Keep Eigenvalues Away From Zero

Add identity to XX^T

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T Y \tag{54}$$

8.2. LASSO, L1 Norm

$$min_\beta \frac{1}{N}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda||w|| \tag{55}$$

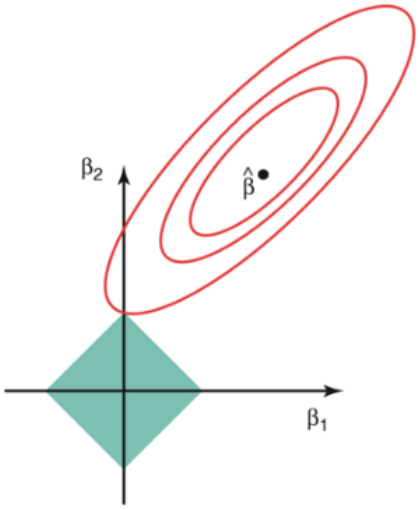


Figura 2. LASSO/L1 Regularization

$\lambda \propto \frac{1}{\tau}$

8.3. Ridge, L2 Norm

$$min_{\beta} \frac{1}{N} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda ||w||^2 \tag{56}$$

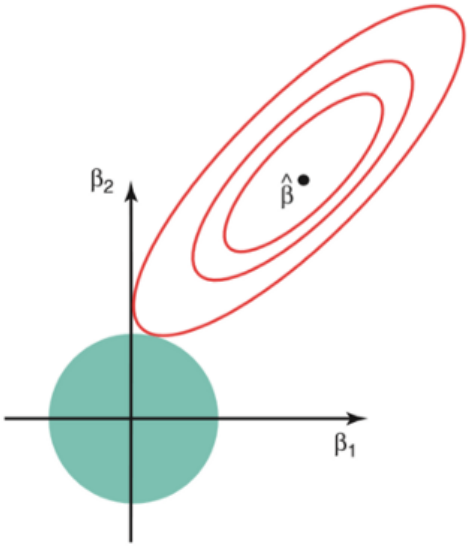


Figura 3. Ridge/L2 Regularization

$\lambda \propto \frac{1}{\tau}$

8.4. L1 vs. L2

- L1 is for feature selection

- L2 is for avoiding overfitting

Read More

9. Data Pre-processing

9.1. LDA vs. PCA

9.2. Scatter Matrix

- $m, S_T = \Sigma$: mean and covariance of all samples
- m_i, Σ_i: mean and covariance of class i

- Total scatter equals sum of within and between Within-scatter and Between-scatter
- Within-scatter:

$$S_w = \sum_{i=1}^{C} \frac{n_i}{n} \Sigma_i \tag{57}$$

- Between-scatter:

$$S_B = \sum_{i=1}^{C} \frac{n_i}{n} (m_i - m)(m_i - m)^T \tag{58}$$

10. The Number of Parameter For Estimation

Assume we have a training set consisting of n objects and d features with 2 classes.

10.1. Bayesian Classifiers

- for mean: $2 \times d$
- for variance:
  - if correlation$= 0$: $2 \times d$
  - if correlation$\neq 0$: $2 \times \frac{d \times (d+1)}{2}$
- for class prior: 1
- totally: $1 + 3d + d^2$ or $1 + 4d$

10.2. SVC

If it is a linear one

- constrain C: 1
- Lagrange multipliers: n
- totally: $1 + n$

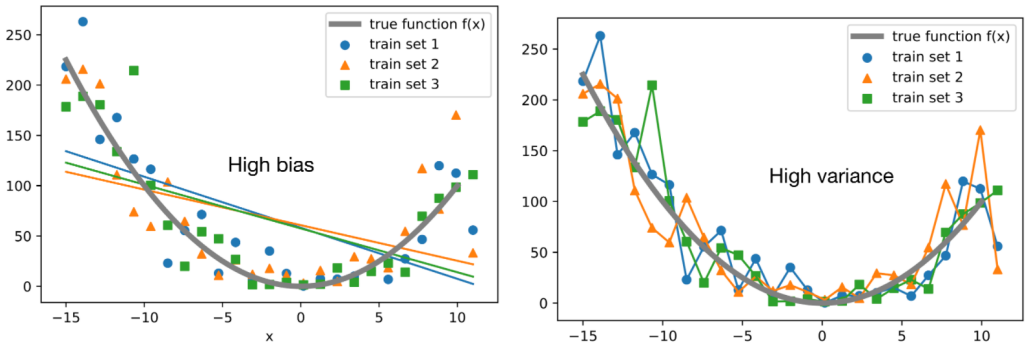
10.3. Parzen

If the kernel is fixed

- kernel width h: 1
- totally: 1

11. Curves

11.1. Bias-Variance Decomposition



(a) High Bias (b) High Variance

Figura 4. High Bias vs. High Variance

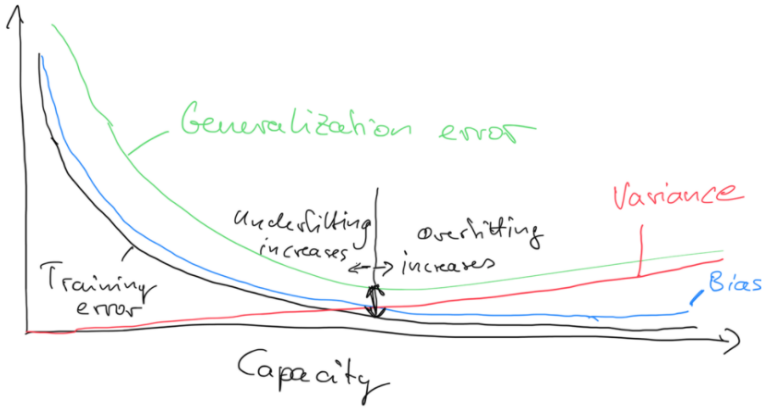


Figura 5. Decomposition Of Loss

11.2. Cross Validation Curve

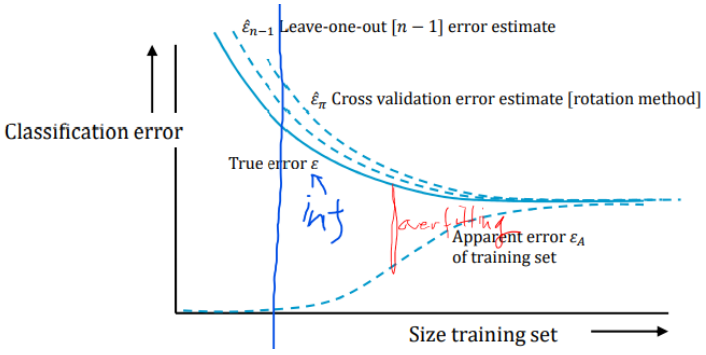


Figura 6. Cross Validation Curve

11.3. Learning Curve

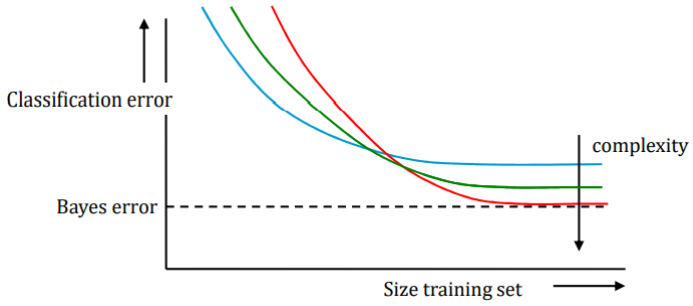


Figura 7. Learning Curve

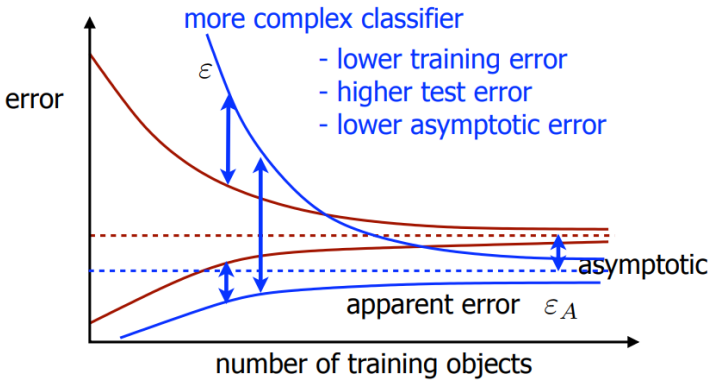


Figura 8. Learning Curve 2

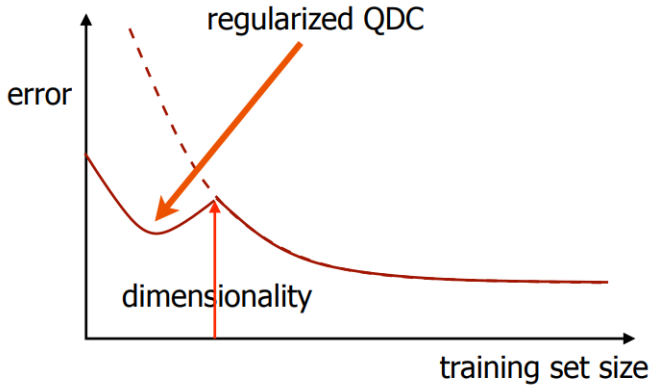


Figura 9. Regularized QDC

11.4. Feature Curve

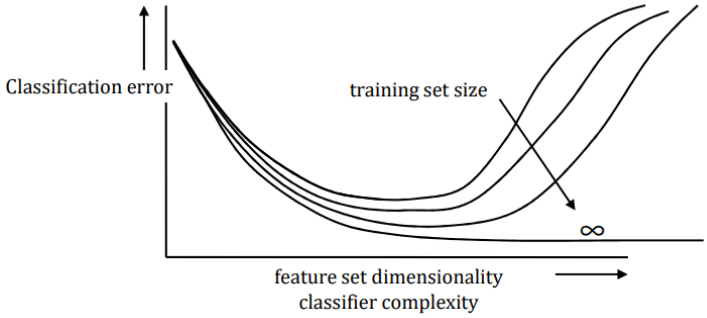


Figura 10. Feature Curve

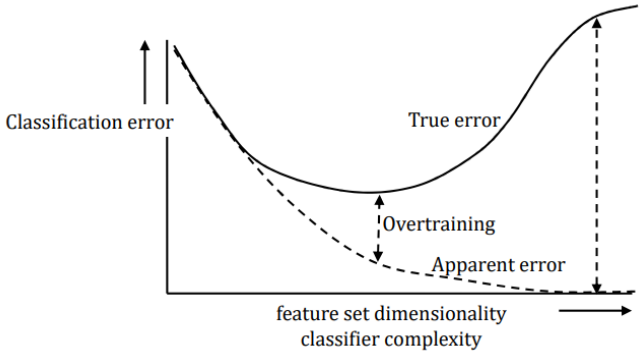


Figura 11. Curse of Dimensionality

11.5. ROC: receiver operating characteristic curve

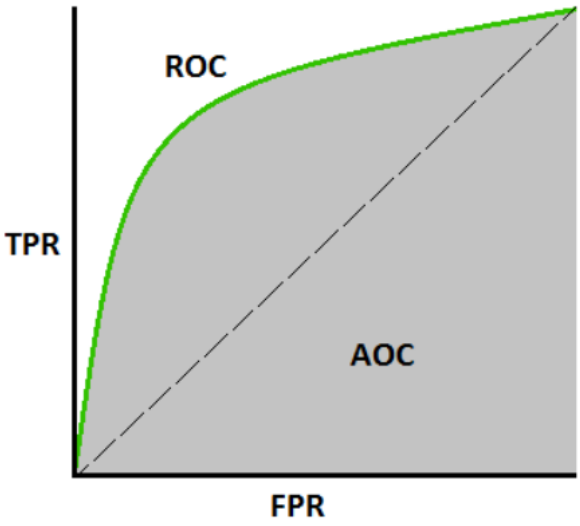
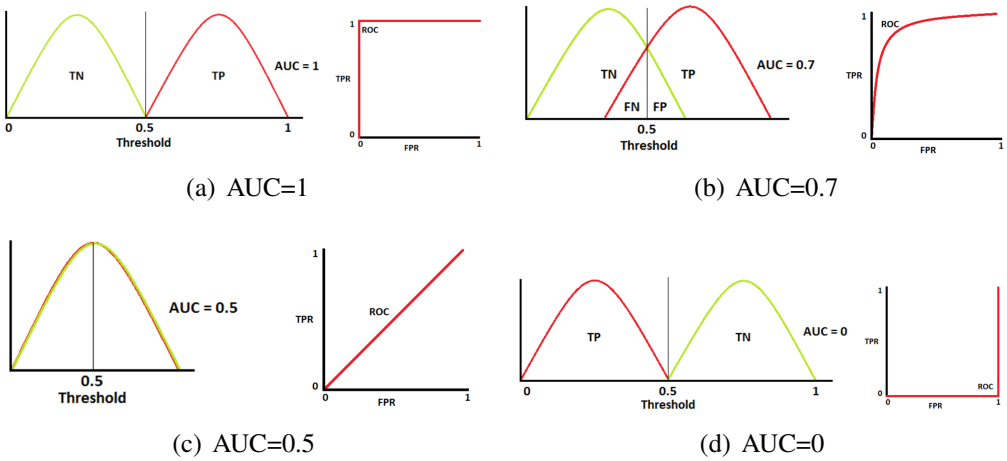


Figura 12. ROC Curve

$$TPR/Recall/Senstivity = \frac{TP}{TP + FN} \tag{59}$$

$$FPR = \frac{FP}{TN + FP} \tag{60}$$



(a) AUC=1 (b) AUC=0.7

(c) AUC=0.5 (d) AUC=0

Figura 13. Different ROCs

Area under the Curve of ROC (AUC ROC)

12. DAG

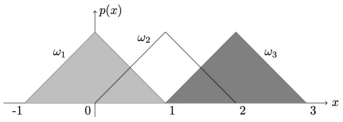
Directional-separation (D-separation): If every path between A and B is blocked then $A \perp B|C$ conditionally independent: Given that I have observed C, observing B will not give me additional information about A

13. Training Questions

13.1. Bayes Classifier & Bayes Error

For the Bayes classifier, one has to know the true class conditional probabilities. **False**
The Bayes error is always smaller or equal to the estimated true error. **False**
The Bayes error is always larger or equal to the apparent error. **False**
The Quadratic Discriminant classifier is a generative classifier. **True**
Even if the Bayes classifier is nonlinear, the nearest mean may still outperform the k-nearest neighbor classifier. **True**
In three dimensions, the nearest mean classifier can typically perfectly separate any configuration of four points from two classes. **False**
You want to know if you are one of the few people that knows all there is to know about artificial intelligence. To find out, you decide to go through a host of exams and other tests. The overall score you receive indicates that you are indeed one of the happy few. Knowing so much about AI, however, you do not jump to conclusions and you check with your local expert how accurate these scores are. "Well," the person in the lab coat tells you, "of course they are not perfect, but these test will correctly identify 99% of the people that know all of ML and will be incorrect in a mere 0.1% when people do not

belong to this select group." True or false: the probability of you knowing all of ML will be greater than 0.99. **False**
By reducing the dimensionality one can decrease the Bayes error. **False**

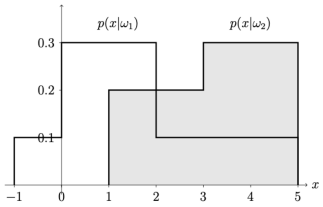


- Assume that all classes are equiprobable. Where is the optimal decision boundary (Bayes' classifier) is located? **At x=0.5 and x=1.5**
- What is the expected confusion matrix for the Bayes' classifier when the classes are equiprobable, and we classify 1000 points (rounded to the nearest integer).

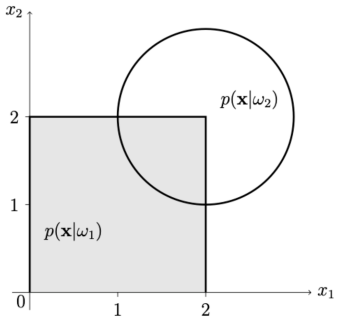
$$\begin{pmatrix} 292 & 42 & 0 \\ 42 & 250 & 42 \\ 0 & 42 & 292 \end{pmatrix}$$

- Assume that the class priors are changed such that class 1 is now twice as likely as class 2 and 3: $p(w_1) = 2p(w_2) = 2p(w_3)$. How does the optimal decision boundary change? How will the expected confusion matrix look like.

$$\begin{pmatrix} 389 & 111 & 0 \\ 56 & 163 & 31 \\ 0 & 31 & 219 \end{pmatrix}$$



- Compute the Bayes error when $p(w_1) = p(w_2)$. **0.25**
- Compute the Bayes error when $p(w_1) = 0.2$ and $p(w_2) = 0.8$. **0.12**
- Suppose, we again have $p(w_1) = p(w_2) = 0.5$, and suppose we are fitting a nearest mean classifier on the given training data. Compute the true error made by the nearest mean classifier.
  w1:x1=0.6,x2=0.9,x3=1.6,x4=2.4,x5=4.5
  and
  w2:x6=2.5,x7=3.5,x8=4.0,x9=5.0,x10=5.0
  0.3
- What is the apparent error of the nearest mean classifier derived from the previous question? **0.2**
- Where is the decision boundary of the 3-nearest neighbour classifier? **Between x_6 and x_7.**



- Assume that both classes are equally likely. What is the Bayes error for this problem? **0.098**
- Now assume that the prior of class 1 is changed to 0.8. What will be the Bayes error now? **0.05**
- Assume we fit a logistic classifier:

$$p(w_1|x) = \frac{1}{1 + e^{-(w^T x + w_0)}}$$

  on a very large training set. In which direction will w point towards? $w = [-1, -1]^T$
- Now we have three classifiers available: (1) the nearest mean classifier, (2) the quadratic classifier and (3) the 1-nearest neighbour classifier. What classifier should you choose for (a) very small training set sizes, and for (b) very large training set sizes? **Small training size: nearest mean, large training size: 1-nearest neighbour**

13.2. Regularization & MAP

Consider the standard probabilistic form of (unregularized) linear regression based on a Gaussian noise model. Denote the likelihood of this model by p(X|w) with X the observed data. Let q(w) be a pdf that is uniform on all values of w for which the L1 norm is smaller than some constant a. The MAP solution, using this prior, is equivalent to L1-regularized least squares regression. **True**
Consider the standard probabilistic form of (unregularized) linear regression based on a Gaussian noise model. Denote the likelihood of this model by p(X|w) with X the observed data. Let q(w) be a pdf that is uniform on all values of w for which the L2 norm is smaller than some constant a. The MAP solution, using this prior, is equivalent to... **L2 regularized least squares regression**
The unregularized empirical loss for a particular least-squares regression problem in 2D is as follows:

$$w_1^2 + w_2^2 - 4w_1 - 8w_2 + 32$$

The optimal solution is w=(2,4).

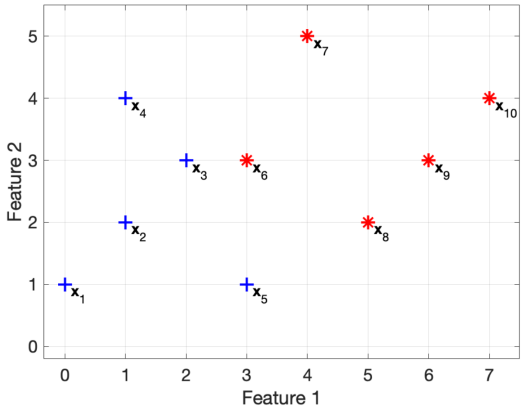
- The empirical loss achieved by this optimal solution equals: **12**
- Say, we want to have a L1 regularized least-squares solution en we consider weight vectors for which $||w||_1 \leq 1$. What is the regularized solution going to be? **(0,1)**
- Let us now consider all unregularized least-squares loss function with a minimum at w=(2,4). With different underlying loss functions, the iso-loss ellipses change,

and so one may find different L1 regularized solutions. Assume again that $||w||_1 \leq 1$. Identify all the values our regularized solution can take on. **All values on the line connecting the points (0,1) and (1,0); All values on the line connecting the points (0,1) and (-1,0)**

- Let t be the regularization parameter that we need to set, making us only consider solutions for which $||w||_1 \leq t$. (And which was set to 1 in the foregoing.) What is the smallest t for which we will find the optimal regularized solution to be the same as the original unregularized solution? **6**

13.3. SVC

The VC dimension is only defined for support vector classifiers. **False**
To find the Support Vector classifier on a training set, you need to know the class prior probabilities $p(y_i)$. **False**
What is the leave-one-out-crossvalidation error of a linear support vector classifier (with no slack)?



The support vectors are x_3, x_5 and x_6. They run the risk of being misclassified when being left out in the leave-one-out-crossvalidation.

If we leave out x_3, then x_4 will become support vector, but x_3 will still be correctly classified.

If we leave out x_5, then only x_3 and x_6 will become support vector, and x_5 will be misclassified.

If we leave out x_6, then x_7 and x_8 will become support vector, and x_6 will be misclassified.

The error then becomes $\frac{2}{10}$

13.4. Gradient Descend

$$J(w, w_0) = \sum_{miscalssify\ x_i} \sqrt{-y_i(x^T w + w_0)}$$

We start with initialisation $w = [1, 0]^T$, $w_0 = 0.01$, and we use a learning rate of $\eta = 0.1$. Given dataset ($x_1 = [0, -1]^T$,$y_1 = -1$),($x_2 = [1.5, 0]^T$,$y_2 = +1$),($x_3 = [0, +1]^T$,$y_3 =$

$+1$) what are the parameters values after one update step? **w=[1.0,0.5], w_0=-0.49**

13.5. Logistic Classifier/Regression

The logistic classifier assumes that the decision boundary can be modeled by a logistic function. **False**
Linear regression is a special form of logistic regression. **False**
Given are 5 one-dimensional input data points $X = (-1, -1, 0, 1, 1)^T$ and their 5 corresponding outputs $Y = (0, 0, 1, 0, 1)^T$. We are going to have a look at linear regression using polynomial basis functions.

- Fit a linear function (including the bias term) to this date under the standard least-squares loss. What value does the bias term take on? $\frac{2}{5}$
- Fit a linear function (including the bias term) to this date under the standard least-squares loss. What value does the slope take on (i.e. what is the coefficient for the linear term)? $\frac{1}{4}$
- Let us now fit a parabola, a second-order polynomial, to this data. Again, we use the standard squared loss as our optimality criterion. What total loss (i.e., the loss added over all training data point) does the optimal second-order polynomial attain? (Rather than doing the computations, you may want to have a look at a sketch of the situation.) $\frac{1}{2}$
- Again determine the total loss over the training data, but now assume we optimally fitted a third-order polynomial. $\frac{1}{2}$
- Rather than just fitting a least-squares model, we consider a maximum likelihood solution under an assumed Gaussian noise model. That is, we assume that outputs are obtained as a function f from x plus some fixed-variance, independent Gaussian noise. If our fit to the 5 data point equals the constant zero function, i.e., f(x)=0, what then is the maximum likelihood estimate for the variance of the Gaussian noise? $\frac{2}{5}$

We consider linear regression in 10 dimensions, where the input data always takes on the form of a standard basis vector (i.e., all entries are 0, except for one with value 1). We are given three training points. The first one is standard basis vector (1,0,0,...,0,0), the second is (0,1,0,...,0,0), and the third is (0,0,1,...,0,0). Their corresponding outputs are a,b,c, respectively.

- We consider standard L2 regularized linear regression with regularization parameter t. Its solution without intercept is given by:

$$X^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$XY = \begin{pmatrix} a & b & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

  $w = (XX^T + tI)^{-1}XY = [a, b, c, 0, 0, 0, 0, 0, 0, 0]/(t+1)$ (the inverse of diagonal matrix is the reciprocal of the elements of the matrix)
- Let's now consider the unregularized minimum norm solution, i.e., the solution one would also obtain when using the pseudo-inverse. The solution we find is the following: **(a,b,c,0,0,0,0,0,0,0)**

- Let us now also take into account an intercept (also called bias term or offset). The intercept is not regularized, but the other parameters are. Assume that a, b, and c are nonnegative. Which of the following statements are correct?
  - The intercept is always nonnegative in this setting. **This is true**($XX^T + tI$ is positive, $XY = [a, b, c, 0, 0, 0, 0, 0, 0, 0, a + b + c]$ is non-negative)
  - With intercept, no matter how much training data we get, we will always be able to achieve zero squared error. **This is false**

13.6. Cluster

k-Means clustering results reproduce exactly when the initialization of the cluster means is non-random. **True**
In hierarchical clustering, one way to determine the number of clusters is to cut the dendrogram at its largest jump. **True**
Average linkage clustering typically results in more compact clusters than single linkage clustering. **True**
The Davies-Bouldin index looks at the worst-case for each cluster in terms of overlap with other clusters. **True**
Hierarchical clustering results depend on the initialization. **False**

X	1	2	3	4	5
1	0	4	14	40	50
2	4	0	3	19	31
3	14	3	0	7	22
4	40	19	7	0	17
5	50	31	22	17	0

- Assume we perform a 'single linkage' hierarchical clustering on this data. At what height are the (clusters containing) objects 1 and 3 being merged? **4**
- Assume we perform a 'single linkage' hierarchical clustering on this data. According to the largest jump in the fusion graph, what is the number of clusters in this data? **2**
- Assume we perform a 'complete linkage' hierarchical clustering on this data. According to the largest jump in the fusion graph, what is the number of clusters in this data? **2**

13.7. Feature/PCA

The Parzen classifier, i.e., the generative classifier in which the underlying classes are modeled by Parzen estimators, is insensitive to feature scaling. **False**
Principal component analysis reduces the dimensionality such that the class separability is optimized. **False**
We trained a classifier on some dataset, and we obtained the following feature curve. What happens when we increase the number of features further? (We assume that each of the features is informative for the classification problem) **The curve will go up eventually**

We trained a classifier on some dataset, and we obtained the following feature curve: What happens when we increase the number of training objects? **The curve will move down overall**

Given mean-centered data in 3D for which the covariance matrix is given by C. Also given is a data transformation matrix R.

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix}$$

- What is the first principal component of the original data for which we have the covariance matrix C? $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
- Assume we transform all the data by the transformation matrix R, what does the covariance of the transformed data become? $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{7}{2} & -\frac{\sqrt{3}}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{5}{2} \end{pmatrix}$
- What is the first principal component for the transformed data? $\begin{pmatrix} 0 \\ \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{pmatrix}$

Assume that we have a two-class classification problem. Each of the classes has a Gaussian distribution in k dimensions: $p(x|w_i) = N(x; \mu i, I)$, where I is the $k \times k$ identity matrix. The means of the two classes are $\mu_1 = [0, 0, ..., 0]^T$ and $\mu_2 = [2, 2, ..., 2]^T$. Per class, we have n objects per class. On this data a nearest mean classifier is trained.

- When the number of features increases, **the Bayes error decreases.**
- When the number of features increases, **the true error first decreases, then increases again.**
- Before we train a classifier, we also perform a forward feature selection to reduce the number of features to $m = [k/2]$. When the number of features increases, **the true error first decreases, then increases again.**

You are dealing with a 10-dimensional classification problem with two classes and 1000 samples per class. You decide to study the nearest mean classifier (NMC) in combination with a feature selection to 3 (three) features in order to solve this classification problem. As feature subset evaluation criterion you take the classification performance the respective classifier has on the training set.

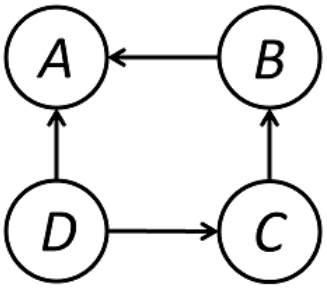
- How many criterion evaluations are necessary to come to your choice of 3 features (out of the 10) when you use sequential feature forward selection? **10+9+8=27**
- How many criterion evaluations are necessary to come to your choice of 3 features (out of the 10) when you use sequential feature backward selection? **10+9+8+7+6+5+4=49**
- How many criterion evaluations do you need to make sure you find the overall optimal combination of 3 features? C_{10}^3**=120**

- Recall that the NMC is not scale-independent (NMC is scale-dependent). Say it is allowed for features to be chosen more than once. How many criterion evaluations do you need to make sure you find the overall optimal combination of 3 features (so they don't have to be unique)? $C_{10}^3 + C_{10}^2 A_2^2 + C_{10}^1$**=220**

13.8. Bias-Variance

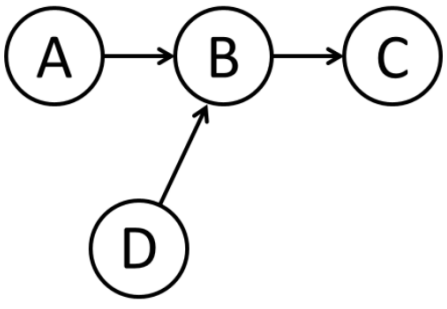
Consider 30 draws of 10 training samples from a 2-class classification problem p(x,y) in 3 dimensions. Say we train the same classifier on all of these data sets and measure their performances on the same, large and independent test set. It turns out that the trained classifiers misclassify the same part of the test set. Is this a result of classifier bias or classifier variance? **Bias**

13.9. DAG



When are D and B independent of each other? **Conditional upon observing C**

We are given the following DAG.



In addition, say that the variables, A, B, C, and D can only take on the values 0 and 1. Finally, we are given the following:

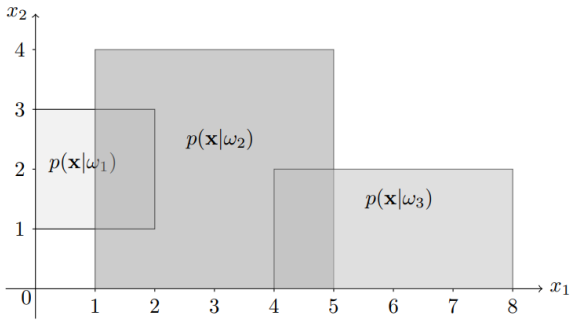
$P(A = 0) = 1/4$
$P(D = 0) = 1/2$
$P(B = 0|A = 0, D = 0) = 1/4$
$P(B = 0|A = 0, D = 1) = 3/4$
$P(B = 0|A = 1, D = 0) = 1/2$
$P(B = 0|A = 1, D = 1) = 1/2$
$P(C = 0|B = 0) = 1/2$
$P(C = 0|B = 1) = 1/4$

- Consider all pairs of different nodes from the given DAG and indicate which pairs are d-separated by the empty set. **AD**

- Given that B=0, what is the value of the covariance between the variables A and C? **0**
- Determine the value of P(B=0). $\frac{1}{2}$
- Given the same definition of the various prior and conditional probabilities as under the previous item, what is the probability P(A=0,D=0|B=0). $\frac{1}{16}$

14. Testing Questions

14.1. Question 1:



Let us consider the above two-dimensional, three-class classification problem, with two class-conditional distributions $p(\mathbf{x}|\omega_1)$, $p(\mathbf{x}|\omega_2)$ and $p(\mathbf{x}|\omega_3)$. Class ω_1 has a uniform distribution for $0 < x_1 < 2$, $1 < x_2 < 3$, class ω_2 has a uniform distribution for $1 < x_1 < 5$, $0 < x_2 < 4$ and class ω_3 has a uniform distribution for $4 < x_1 < 8$, $0 < x_2 < 2$.

Assume that all classes are equally likely.

1. Determine the decision boundary of the Bayes classifier.
2. What are the posterior probabilities for x = [4.5, 1]?
3. Determine the classification error that the Bayes classifier is making, i.e. the Bayes error.
4. Determine the decision boundary of the nearest mean classifier when it is trained on a extremely large dataset drawn from this distribution. Give an explicit formula that shows which x are part of this boundary.
5. Assume now that the class priors are changed to: P(w1) = P(w3) = 1/5 and P(w2) = 3/5. How does the Bayes classifier change? How large does its error become?

14.2. Solution 1:

1. The boundary between 1 and 2 is (1,1)-(2,1)-(2,3)-(1,3) where the overlapping area belongs to w1. The boundary between 2 and 3 is (4,0)-(4,2)-(5,2) where the overlapping area belongs to w3.
2. $p(w_3|x) = \frac{p(x|w_3)p(w_3)}{p(x|w_1)p(w_1)+p(x|w_2)p(w_2)+p(x|w_3)p(w_3)} = \frac{1/8\times1/3}{0+1/16\times1/3+1/8\times1/3} = \frac{2}{3}$
3. $P_e = 1/3 \times 1/16 \times (3-1)(2-1) + 1/3 \times 1/16 \times (2-0)(5-4) = \frac{1}{12}$
4. $\mu_1 = (1,2)$, $\mu_2 = (3,2)$, $\mu_3 = (6,1)$
   the boundary between 1 and 2 is $y = x_1 - 2$ where 1 is negative and 2 is positive
   the boundary between 2 and 3 is $y = (6-3, 1-2)(x_1, x_2)^T + b$, put x = (4.5,1.5) and y = 0 in $\rightarrow$ $b = -12$ (see eq.27 if you don't understand how to get ω). Therefore, the boundary is $y = (3,-1)(x_1, x_2)^T - 12$ where 2 is negative and 3 is positive.

5. The boundary between 1 and 2 is (1,1)-(2,1)-(2,3)-(1,3) where the overlapping area belongs to w1. The boundary between 2 and 3 is (4,0)-(4,2)-(5,2) where the overlapping area belongs to w2.
   $P_e = 3/5 \times 1/16 \times (3-1)(2-1) + 1/5 \times 1/8 \times (2-0)(5-4) = \frac{1}{8}$

14.3. Question 2:

a. Consider 1000 samples from a 3D Gaussian distribution. Assume this sample has zero mean and a 3 × 3-covariance matrix C given by

$$C = \begin{pmatrix} 99 & 0 & 0 \\ 0 & 99 & 0 \\ 0 & 0 & 124 \end{pmatrix}$$

1. Which direction gives the first principle component for this data set?

b. It turns out, that these 1000 samples are in fact the class means of 1000 different classes of bird species described by three different features. Assume that all class priors are $\frac{1}{1000}$ and that all these classes are normally distributed with covariance matrix equal to the identity matrix.

1. Give the total covariance matrix (also called the mixture scatter matrix) for this data set.
2. The Fisher mapping determines the Eigenvectors with the largest Eigenvalues of the matrix $\Sigma_W^{-1}\Sigma_B$. Which direction gives the optimal Fisher mapping if we want to reduce the feature space to one dimension? (So, we look for a single 3D vector here.)

c. In a next step, a linear feature transformation T is applied to the original 3D space. The 3 × 3- matrix describing this transformation is given by

$$T = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

1. Recall your answer under b. and show that the total covariance matrix of the data after the transformation equals?
2. Are the first two features correlated and, if so, are they positively or negatively correlated?
3. Determine the first principal component for this new feature space.

14.4. Solution 2:

1. This should be a vector pointing in the direction of the third dimension.

1. ST = SB + SW where SB=C and SW=I. The total covariance is given by the sum of the 3D covariance of the Gaussian plus the identity matrix, which gives diag(100,100,125) as the solution.
2. One can do the explicit calculations and find the Eigenvector with the largest Eigenvalue, but one can also see that, as the mean within class is spherical, we will find the same component as the one under a.

1.
$$T \cdot C \cdot T^{transpose} = \begin{pmatrix} 125 & 100 & 0 \\ 100 & 125 & 0 \\ 0 & 0 & 125 \end{pmatrix}$$
2. Considering the off diagonals, they are positively correlated.
3. The first dimension owns the highest Eigenvalues and the second dimension is correlated to the first dimension, so (1,1,0) has the largest variance and therefore is the first PC.

14.5. Question 3:

Assume we have a one-dimensional dataset with five objects: x1 = -0.5, x2 = +0.5, x3 = +3.0, x4 = +3.5 and x5 = 5.5. First, we are clustering this dataset with a Mixture of Gaussians with k = 2 clusters. A mixture of 2 Gaussians models the data with the following probability density function:

$$p(x) = \sum_{i}^{2} P_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

with $\sum_i P_i = 1$.

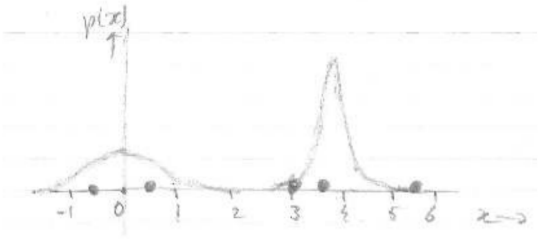
1. Compute the loglikelihood of the data, given the cluster parameters: μ_1 = 0, μ_2 = 3.5, $\sigma_1 = \sigma_2$ = 1 and $P_1 = P_2$ = 0.5.
2. Now change the standard deviation of class two to $\sigma_2 = 0.1$ (the standard deviation of class one stays the same, i.e. 1). Sketch this probability density and indicate where the 5 objects are located. Argue if the loglikelihood of this mixture is smaller, equal, or larger than the loglikelihood from question (a).

Consider now the situation that we are clustering using hierarchical clustering, with average linkage.

1. Perform the hierarchical clustering, and draw the fusion graph (also called the threshold dendrogram).
2. Compute the cophenetic correlation coefficient (also known as the Pearson correlation) between the true distances and the dendrogram distances of objects x3, x4, and x5.

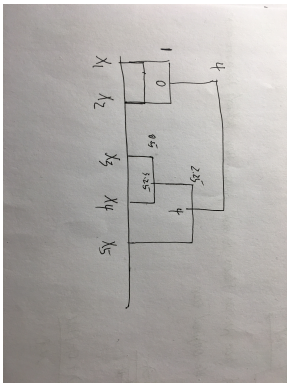
14.6. Solution 3:

1. sum += np.log(1/(2*np.sqrt(2*np.pi)) * (np.exp(-1/2*x**2)+np.exp(-1/2*(x-3.5)**2))). I get -10.408, but the answer provided is -3.9595.



2. I think the log-likelihood becomes smaller, but the provided answer is higher.

1. Fusion levels are 0.5, 1, 2.25 and 4 for average linking.



2. It is not necessary to know for this time.

Solution: The true distances are

$$D = \begin{pmatrix} 0 & 1 & 3.5 & 4 & 6 \\ 1 & 0 & 2.5 & 3 & 5 \\ 3.5 & 2.5 & 0 & 0.5 & 2.5 \\ 4 & 3 & 0.5 & 0 & 2 \\ 6 & 5 & 2.5 & 2 & 0 \end{pmatrix}$$

The distances over the dendrogram are:

$$D = \begin{pmatrix} 0 & 1 & 4 & 4 & 4 \\ 1 & 0 & 4 & 4 & 4 \\ 4 & 4 & 0 & 0.5 & 2.25 \\ 4 & 4 & 0.5 & 0 & 2.25 \\ 4 & 4 & 2.25 & 2.25 & 0 \end{pmatrix}$$

So for the full correlation, we need to compute the correlation coefficient between $d_1 = [1, 3.5, 4, 6, 2.5, 3, 5, 0.5, 2.5, 2]$ and $d_2 = [1, 4, 4, 4, 4, 4, 4, 0.5, 2.25, 2.25]$. It appears $\rho = 0.8175$.

For the subset of the three objects we only have coefficient between $d_1 = [0.5, 2.5, 2]$ and $d_2 = [0.5, 2.25, 2.25]$. Then $\rho = 0.97$.

The correlation in itself is not so important, but the fact that you compute it over the distances *is*.

14.7. Question 4:

1. The Bayes error for a two-class classification problem is smaller than the Bayes error of a three-class problem.
2. To train the quadratic classifier, you need to estimate the class prior probabilities.
3. For a k-nearest neighbor classifier you need to estimate the class prior probabilities.
4. To train the logistic classifier, you need to estimate the class conditional probabilities.
5. Rescaling the features of a classification problem may improve the classification performance of the nearest mean classifier.
6. If no features are available, the best classification in a two-class problem is realized by assigning all objects to the class with the largest prior probability.
7. The classification error found for an infinite training set is for the Bayes classifier a nonincreasing function of the number of features.
8. Due to their immense flexibility, deep nets (i.e., an artificial neural network with a lot of hidden layers) typically outperform all other classifiers.
9. If one knows the AUC (the area under the ROC) associated with a classifier, then one can also determine the classification error rate that classifier gives.
10. Using the reject option typically leads to improved error rates

14.8. Solution 4:

1. False
2. True
3. False
4. False
5. True
6. True
7. True
8. False
9. False
10. True