1 Univariate Statistics

- A discrete random variable x can assume any of a finite number of m different values in $X = v_1, v_2, \ldots, v_m$. The probability of the value p_i is defined as: $Pr[x = v_i]$ with $p_i \geq 0$ and $\sum_{i=1}^{m} p_i = 1$. A random real variable has a probability function $p(x)$.
- Expected value ($\equiv$ mean $\equiv$ average) of a random variable x that can assume m different values in X:

$$\varepsilon[x] = \mu = \sum_{x \in X} x \cdot p(x) = \sum_{i=1}^{m} v_i \cdot p_i. \tag{1.1}$$

  For a random real variable the expected value is:

$$\varepsilon[f(x)] = \sum_{x \in X} f(c) \cdot p(x). \tag{1.2}$$

- Linearity of the mean:

$$\varepsilon[\alpha_1 \cdot f_1(x) + \alpha_2 \cdot f_2(x)] = \alpha_1 \cdot \varepsilon[f_1(x)] + \alpha_2 \cdot \varepsilon[f_2(x)] \tag{1.3}$$

- Second moment (skewness):

$$\varepsilon[x^2] = \sum_{x \in X} x^2 \cdot p(x). \tag{1.4}$$

- Variance:

$$\begin{aligned} \text{var}[x] &= \sigma^2 \\ &= \varepsilon[(x - \mu)^2] \\ &= \varepsilon[x^2] - (\varepsilon[x])^2 \end{aligned} \tag{1.5}$$

- Unlike the mean, the variance is not linear.

2 Joint Probability Statistics

- For two random variables $x \in X$ ($X = v_1, v_2, \ldots, v_m$) and $y \in Y$ ($Y = w_1, w_2, \ldots, w_m$) the joint probability $p_{ij} = Pr[x = v_i, y = w_i]$, with $p(x, y) \geq 0$ and $\sum_{x \in X} \sum_{y \in Y} p(x, y) = 1$.
- Marginal probability of variable x from a joint probability distribution:

$$p_x(x) = \sum_{y = Y} p(x, y). \tag{2.1}$$

- Two variables are statistically independent iff $p(x, y) = p_x(x) \cdot p_y(y)$.
- Expected value of two variables:

$$\varepsilon[f(x, y)] = \sum_{x \in X} \sum_{yinY} (f(x, y) \cdot p(x, y)) \tag{2.2}$$

- Mean of a variable x from a joint probability distribution:

$$\mu_x = \varepsilon[x] = \sum_{x \in X} \sum_{y \in Y} (x \cdot p(x, y)) \tag{2.3}$$

- Variance of a variable x from a joint probability distribution:

$$\begin{aligned} \text{var}[x] &= (\sigma_x)^2 \\ &= \varepsilon[(x - \mu_x)^2] \\ &= \sum_{x \in X} \sum_{y \in Y} \left((x - \mu_x)^2 \cdot p(x, y)\right) \end{aligned} \tag{2.4}$$

- Covariance (= cross-moment):
$$\begin{aligned}\sigma_{xy} &= \varepsilon[(x-\mu_x)\cdot(y-\mu_y)] \\ &= \sum_{x\in X}\sum_{y\in Y}((x-\mu_x)\cdot(y-\mu_y)\cdot p(x,y))\end{aligned} \tag{2.5}$$
- Correlation coefficient:
$$\rho = \frac{\sigma_{xy}}{\sigma_x\cdot\sigma_y} \tag{2.6}$$
- Correlation coefficient interpretation:
  $\rho = -1 \quad \rightarrow$ events are maximally negatively correlated.
  $\rho = 0 \quad \rightarrow$ events are uncorrelated.
  $\rho = 1 \quad \rightarrow$ events are maximally positively correlated.
- Conditional probabilty:
$$P(x|y) = \frac{P(x,y)}{P(y)} \tag{2.7}$$
- Covariance matrix using vectors x and μ:
$$\Sigma = \varepsilon[(x-\mu)(x-\mu)^T] \tag{2.8}$$
- Properties of the covariance matrix: symmetric, positive-semi-definitive, eigenvalues ≥ 0.
- The covariance matrix is diagonal if the variables are statistically independent.

3 Probability Distribution Function

z-score The mahalanobis distance (Equation 3.4) in one dimension.

- The probability distribution function of the sum of two independent random variables is the convolution between the concerned pdf's.
- Normal distribution:
  - Pdf:
$$p(x) = \frac{1}{\sqrt{2\cdot\pi}\cdot\sigma}\cdot e^{-\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2}} \tag{3.1}$$
  - Mean:
$$\mu = \varepsilon[x] = \int_{-\infty}^{\infty} x\cdot p(x)dx \tag{3.2}$$
  - Variance:
$$\sigma^2 = \varepsilon[(x-\mu)^2] = \int_{-\infty}^{\infty}(x-\mu)^2\cdot p(x)dx \tag{3.3}$$
- Mahalanobis distance from x to μ:
$$r = \frac{|x-\mu|}{\sigma} \tag{3.4}$$
- Probability density function of a d-dimensional normal distribution:
$$p(x) = \frac{1}{(2\cdot\pi)^{\frac{d}{2}}\cdot|\Sigma|^{\frac{1}{2}}}\cdot e^{-\frac{1}{2}\cdot(x-\mu)^T\cdot\Sigma^{-1}\cdot(x-\mu)} \tag{3.5}$$
- Squarred muldidimensional Mahalanobis distance:
$$r^2 = (x-\mu)^T\cdot\Sigma^{-1}\cdot(x-\mu) \tag{3.6}$$

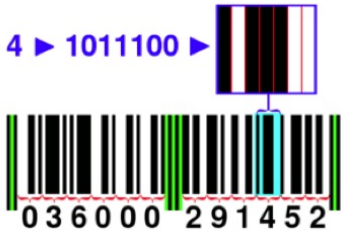


Figure 4.1: *Universal Product Code with the representation of four highlighted.*

- Entropy of a distribution:

$$H(p(x)) = -\int_{-\infty}^{\infty} p(x) \ln p(x) dx \tag{3.7}$$

4 Universal Product Code (= Bar Code)

Feature vector An n-dimensional vector of numerical features that represent some object.
Pattern (Relevant part of) acquired or sensed data.

- Each UPC
  - has thirty bars that code for 95 bits.
  - starts and ends with an 101.
  - has a 01010 in the middle.
- A group of seven bits code a decimal digit, see Figure 4.1.
- Automatic UPC recognition:
  - Segment the UPC
  - Divide the UPC in 95 segments of equal width, using the start bars to determine bar width.
  - Assign 1 or 0 to dark or light segments.

5 Recognizing Natural Patterns

Decision Theory Devise a decision rule or set a decision boundary as to minimize some error.
Generalization Suggest actions when presented with novel patterns.
Error rate How many patterns are assigned to the wrong category.
Risk total expected cost.
Model Typically mathematical description of the populations to be classified.
Learning The process of providing an automatic pattern recognition system with feature vectors that can be used to classify patterns.
Supervised learning The type of learning in which a feature vector is provided with a label that specifies the class of the object.

- The values of the features extracted from natural objects exhibit a certain spread, which is dealt with by statistical methods.

- Stages of pattern recognition:
  1. Sensing: acquiring input data.
  2. Pre-processing/segmentation: select relevant information from input data.
  3. Feature extraction: extract numerical or symbolic values that are distinguishing and invariant to irrelevant transformations of the input.
  4. Classification (or recognition or identification or authentication): assign a category to an object.
  5. Post-processing: take some action taking into account error/risk/cost.
- Ocam's razor in pattern recognition: we might be satisfied with slightly poorer performance on the training samples if it means that our classifier will have better performance on novel patterns.
- Design cycle of a pattern recognition system:
  1. Data collection:
  2. Feature choice: select distinguishing features.
  3. Model choice: which features are used in the models and which models are used.
  4. Training: the process of using the data to determine the classifier.
  5. Evaluation: measure the performance of the system and optionally identify the need for improvements in its components.
- In statistical pattern recognition made inferences are stated in terms of probabilities and the quality of classification is measured in error rates.
- A pattern (object) that is represented by a feature vector needs to be classified. This can be achieved by comparing this feature vector with previously stored feature vectors for which it is known what types of objects they represent.

6 Recognition based on dissimilarity to a prototype

- Hamming distance: The number of positions at which the corresponding symbols are different. Hamming distance between I_t and I_p:
  $$HD(I_t, I_p) = |I_t \,\mathrm{XOR}\, I_p|. \tag{6.1}$$
  The normalized hamming distance between I_t and I_p:
  $$HD(I_t, I_p) = \frac{|I_t \,\mathrm{XOR}\, I_p|}{|I_t|}. \tag{6.2}$$
- The difference in hamming distance of a binary representation is $^1/_2$.
- The central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined expected value and well-defined variance, will be approximately normally distributed.
- From the central limit theorem follows that the envelope of the hamming distance of binary codes can be fitted very well by a normal distribution.
- Upside of using a mathematical model instead of a hard boundary value: we can use it to compute the chance of measuring a certain value of the Hamming distance also in domains in which the histogram contains no data

7 Iris Recognition

Affine transformation Transformation that preserves the relation of parallelism between lines.

- Advantages of iris recognition for visual recognition of persons:
  - Its inter-person variability is enormous.
  - It is well protected from the environment.
  - It is stable over time.
  - It is relatively insensitive to the angle of illumination.
  - Changes in viewing angle cause affine transformations.
  - Non-affine changes (pupillary dilation) are easily reversible.
  - Eyes are easy to locate.
- The key to iris recognition is the failure of a test of statistical independence which involves so many degrees of freedom this test is virtually guaranteed to be passed whenever the binary representation for two different eyes but to be uniquely failed when an eye's binary representation is compared with another version of itself.
- Normalized Hamming distance between two binary representations (*codeA*, *codeB*) with masks (*maskA*, *maskB*):
$$HD = \frac{\|codeA \oplus codeB \cap maskA \cap maksB\|}{\|maskA \cap maksB\|} \tag{7.1}$$
The masks prevent non iris-artefacts from influencing iris comparisons.
- The expected *HD* for two different irises is 0.5 since any given bit is equally likely to be one or zero and different irises are uncorrelated.
- Only small bits of subsets of bits of the binary representation of the iris are mutually independent due to internal correlation in the iris.
- In practice number of degrees of freedom is 244 for iris recognition.
- It is extremely probably that two different irises disagree by $< {}^1/_3$ of the bits, since the minimum HD of the irises that are the same is $\approx {}^1/_3$.
- If p_1 is the false match probability for single one-to-one verification trials than p_n:
$$p_n = 1 - (1 - p_1)^n \tag{7.2}$$
is the probability of making at least one false match when searching a database of n unrelated patterns.
- Identification is vastly more demanding than one-to-one verification.
- A dual distribution representation of the decision problem may be called the decision environment, because it reveals the extent to which the two cases (same versus different iris) are separable and thus how reliably decisions can be made, since the overlap between the two distributions determines the error rate.

8 Hypothesis Testing

Type I error The null hypothesis is true, but is rejected. It is asserting that something is absent, a false hit, see Figure 8.1.

Type II error The null hypothesis is false, but erroneously fails to be rejected. It is failing to assert what is present, a miss, see Figure 8.1.

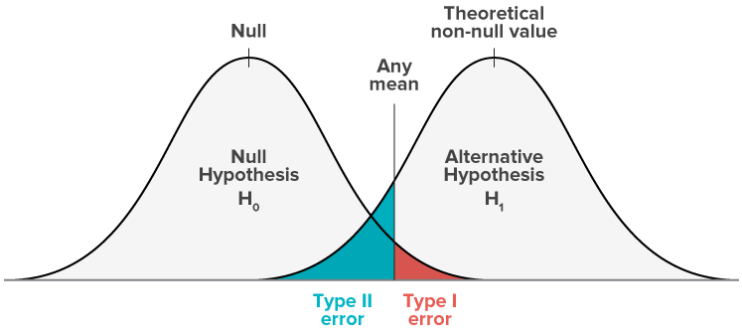


Figure 8.1: *Type I and Type II errors.*

- Hypothesis testing applied to iris scanning: Given the probability density function(HD) that two irises are different formulate two hypotheses:
  - H_0 (null hypothesis): the two irises are different.
  - H_a (alternative hypothesis): the two irises are identical.

  Select a decision criterion (HD value) C and a rejection region according to the desired significance level (odds of false decision or false rejection of H_0). Compare the two iris codes and obtain their HD. Test H_0 hypothesis:
  - Do not reject H_0 if $HD > C$
  - Reject H_0 and accept H_a if $HD \leq C$
- For two choice tasks the decidability index is one measure of how well separated the two distributions are:

$$d' = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{{\sigma_1}^2 + {\sigma_2}^2}{2}}} \tag{8.1}$$

9 Missing Binary Features

Degrees of freedom The number of values used in the computation of a statistic that can vary independently.

Binary degrees of freedom The values used in the computation of a statistic are binary.

- Origins of missing data with iris recognition:
  - Occlusion by eyelids or eyelashes.
  - Reflections from eye-glasses
  - Ring shadow by hard contact lenses.
  - Local signal-to-noise-ration
  - No good local iris texture.
- An iris code mask takes value 1 when the data are considered good and zero when they are bad.
- A tail appears with probability p in a coin tossing experiment, the coin is tossed n times. The numbers of tails d is normally distributed:

$$d \sim \mathcal{N}\left(p \cdot n,\, p \cdot (1-p) \cdot n\right) \tag{9.1}$$

- The higher the number of degrees of freedom, n, the larger the number, 2^n, of unique objects that can be represented and discriminated.
- Comparing the discriminative power of methods that use binary representation and methods that don't: Assume that the dissimilarity D is normally distributed: $D \sim \mathcal{N}(\mu, \sigma^2)$. Transform D to $D' = \frac{D}{2 \cdot \mu} \rightarrow D \sim \mathcal{N}\left({}^1\!/_2, \left(\frac{\sigma}{2 \cdot \mu}\right)^2\right)$. We can think of D' as the Hamming distance of two binary vectors of n statistically independent bits. Using that $D' \sim \mathcal{N}\left(p, \frac{p \cdot (1-p)}{n}\right)$ we get that $n = ({}^{\mu}\!/_{\sigma})^2$. The higher n the higher the descriptive power, this n can be used to compare methods.

10 Introduction

Prior probability Unconditional probability: the prior of a proposition a is the degree of belief accorded to it in the absence of any other information.

Probability mass function A function that gives the probability that a discrete random variable is exactly equal to some value.

Probability density function A function that describes the relative likelihood for a random variable to take on a given value.

Class conditional density function ($p(x|\omega)$) State-conditional pdf: the probability density function for x given that the state of nature ω.

- Using only prior probabilities we get to the decision rule: $\begin{cases} \omega_1 & P(\omega_1) > P(\omega_2) \\ \omega_2 & \text{otherwise} \end{cases}$
- Bayes formula:

$$p(\omega_j, x) = P(\omega_j|x)p(x) = p(x|\omega_j)P(\omega_j) \tag{10.1}$$

$$P(\omega_j|x) = \frac{p(x|\omega_j) \cdot P(\omega_j)}{p(x)} \tag{10.2}$$

  where $p(x)$ is called the evidence factor, $p(x|\omega_j)$ the likelihood and $P(\omega_j)$ the prior probability.
- $p(x|\omega_j)$ is the likelihood of ω_j with respect to x, i.e. other things being equal the category ω_j for which $p(x|\omega_j)$ is large more likely to be the true category.
- The evidence factor can be seen as a scaling factor that guarantees that posterior probabilities sum to one.
- When we observe a particular x the probability of error is:

$$P(error|x) = \begin{cases} P(\omega_1|x) & \text{if we decide } \omega_2 \\ P(\omega_2|x) & \text{if we decide } \omega_1 \end{cases} \tag{10.3}$$

- Bayes decision rule for minimizing the probability of error:

$$\text{Decide } \omega_1 \text{if } P(\omega_1|x) > P(\omega_2|x); \text{ otherwise decide } \omega_2. \tag{10.4}$$

  Under this rule the new (old = 10.3) probability of error:

$$P(error|x) = \min[P(\omega_1|x), P(\omega_2, x)]. \tag{10.5}$$

11 Generalizations of Bayesian Decision Theory

Decision rule A function $\alpha(\mathbf{x})$ that tells us which action to take for every possible observation, i.e. for every $\mathbf{x}$ the decision function $\alpha(\mathbf{x})$ assumes one of the α values.

Overall risk The expected loss associated with a given decision rule. Because $R(\alpha_i|\mathbf{x})$ is the condition risk associated with action α_i and the decision rule specifies the action the overall risk R is given by:

$$R = \int R(\alpha(\mathbf{x})|\mathbf{x}) \cdot p(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{11.1}$$

- To generalize the Bayesian decision theory we replace the scalar x with the feature vector $x \in \mathcal{R}^d$. And we introduce a cost (or a loss) function λ which states how costly each classification is. Let ω_i be the classes and α_i the possible actions with $i \in \mathcal{N}$. The posterior probability $P(\omega_j)|\mathbf{x})$:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j) \cdot P(\omega_j)}{p(\mathbf{x})} \tag{11.2}$$

$$p(x) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j)P(\omega_j) \tag{11.3}$$

- The loss function $\lambda(\alpha_i|\omega_j)$ describes the loss incurred for taking action α_i when the category is ω_j.
- When taking action α_i upon observing $\mathbf{x}$ the conditional risk is:

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j) \cdot P(\omega_j|\mathbf{x}) \tag{11.4}$$

- We can minimize the expected loss by selecting the action which minimizes the conditional risk.
- To minimize overall risk compute the conditional risk (11.4) for all actions and then select the action α_i for which $R(\alpha_i|\mathbf{x})$ is minimum. The resulting minimum risk is the Bayes Risk (R^*).

12 Missing features

Joint distribution The joint probability distribution for at least two random variables $X, Y, \ldots$ is a probability distribution that gives the probability that each of $X, Y, \ldots$ falls in any particular range or discrete set of values specified for that variable.

Marginal distribution The marginal distribution of a subset of a collection of random variables is the probability distribution of the variables contained in the subset. It gives the probabilities of various values of the variables in the subset without reference to the values of the other variables. This contrasts with a conditional distribution, which gives the probabilities contingent upon the values of the other variables.

- To classify a feature vector $\mathbf{x} = [\mathbf{x_g}, \mathbf{x_b}]$ where $\mathbf{x_g}$ are the known/good features and $\mathbf{x_b}$ are the bad features we marginalized the posterior prob-

abilities over the bad features:

$$\begin{aligned} P(\omega_i|\mathbf{x_g}) &= \frac{p(\omega_i, \mathbf{x_g})}{p(\mathbf{x_g})} \\ &= \frac{\int p(\omega_i, \mathbf{x_g}, \mathbf{x_b}) d\mathbf{x_b}}{p(\mathbf{x_g})} \\ &= \frac{\int P(\omega_i|\mathbf{x_g}, \mathbf{x_b}) p(\mathbf{x_g}, \mathbf{x_b}) d\mathbf{x_b}}{p(\mathbf{x_g})} \\ &= \frac{\int g_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x_b}}{\int p(\mathbf{x}) d\mathbf{x_b}} \end{aligned}$$

where g_i is defined as:

$$\begin{aligned} g_i(\mathbf{x}) &= g_i(\mathbf{x_g}, \mathbf{x_b}) \\ &= P(\omega_i|\mathbf{x_g}, \mathbf{x_b}) \end{aligned}$$

We use the following Bayes decision rule: choose ω_i if for all i, j the following holds:

$$P(\omega_i|\mathbf{x_g}) > P(\omega_j|\mathbf{x}_g)$$

- $g_i(\mathbf{i})$ is one form of the discriminant function.
- The distribution $\int p(\omega_i, \mathbf{x_g}, \mathbf{x_b}) d\mathbf{x_b}$ is a marginal distribution, which is the distribution where the full joint distribution is marginalized (integrated) over the variable $\mathbf{x_b}$.

13 Naïve Bayes Probability Estimation

- The central problem in the Bayesian approach to classification: How to estimate class conditional probabilities.
- Bayes rule:

$$P(\omega_j|x_1, x_2, \ldots, x_n) = \frac{p(x_1, x_2, \ldots, x_n|\omega_j) P(\omega_j)}{p(x_1, x_2, \ldots, x_n)} \tag{13.1}$$

- The naïve Bayes rule assumes statistical independence:

$$P(\omega_j|x_1, x_2, \ldots, x_n) = \frac{p(x_1|\omega_j) p(x_2|\omega_j)(x_n|\omega_j) P(\omega_j)}{p(x_1, x_2, \ldots, x_n)} \tag{13.2}$$

- Advantages of the naïve Bayes rule:
  - Each distribution can be independently estimated as an one dimensional distribution.
  - No need for large data sets that scale exponentially with the number of features, curse of dimensionality.
- The naive Bayes Classifier provides a correct classification as long as the correct class is more probable than any other class, hence class probabilities do not have to be estimated very well.
- The naive Bayes classifier fails if the marginal probability density functions overlap.

14 Discriminant Functions

Classifier Something that computes discriminant functions and selects the category corresponding to the largest discriminant.

Decision boundary Surface in feature space where ties occur among the largest discriminant functions.

Dichotomizer Classifier that places a pattern in one of only two categories.

- A classifier assigns a feature vector $\mathbf{x}$ to class ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})\ \forall_{i,j}\ j \neq i$.
- If every discriminant function $g_i(\mathbf{x})$ is replaced by a monotonically increasing function $f(g_i(\mathbf{x}))$, the classification result does not change.
- The effect of any decision rule is to divide the feature space into c decision regions $\mathcal{R}_1, \ldots, \mathcal{R}_c$. if $g_i(\mathbf{x}) > g_j(\mathbf{x})\ \forall_{i,j}\ i \neq j$ then $\mathbf{x} \in \mathcal{R}_i$ and the decision rule calls for us to assign $\mathbf{x}$ to ω_i. The regions are separated by decision boundaries.
- Dichotomizers often have only one discriminant function: $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$ and use the decision rule: ω_1 if $g(\mathbf{x}) > 0$, otherwise decide ω_2.
- The discriminant function of a dichotomizer can be rewritten to:

$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) \tag{14.1}$$
$$= P(\omega_1|\mathbf{x}) - P(\omega_2|\mathbf{x}) \tag{14.2}$$
$$= \ln\left(\frac{p(\mathbf{x}|\omega_1)}{p(x|\omega_2)}\right) + \ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right) \tag{14.3}$$

- Minimum-error-rate classification can be achieved using the discriminant functions.

15 Normal Distribution

- Normal distribution with mean μ, standard deviation σ and variance σ^2:
$$p(x) = \frac{\sigma}{\sqrt{2 \cdot \pi}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$
- Mean of a normal distribution with mean μ and standard deviation σ:
$$\varepsilon[x] = \int_{-\infty}^{\infty} x \cdot p(x) dx = \mu$$
- Variance of a normal distribution with mean μ and standard deviation σ:
$$\varepsilon[(x-\mu)^2] = \int_{-\infty}^{\infty} (x - mu)^2 \cdot p(x) dx = \sigma^2$$
- Multivariate normal distribution with d-dimensional mean μ and covariance matrix Σ:
$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$$
- Element of a covariance matrix of a multivariate normal distribution with mean μ:
$$\varepsilon[(x_i - \mu_i)(x_j - \mu_j)] \int (x_i - \mu_i)(x_j - \mu_j) \cdot p(x) dx = \sigma_{i,j}$$
- The covariance matrix of multivariate normal distribution is always symmetric and positive semi-definite ($|\Sigma| \geq 0$).
- The principal axes of the hyper-ellipsoids defined by the multivariate normal distribution are given by the eigenvectors of Σ, the eigenvalues determine the length of these axes.
- The squared Mahalanobis distance from a point $\mathbf{x}$ to a class $\mathcal{N}(\mu, \Sigma)$:
$$r^2 = (\mathbf{x} - \mu)^T\Sigma^{-1}(\mathbf{x} - \mu)$$
- The contours of constant density are hyper-ellipsoids of constant Mahalanobis distance to μ.
- Linear combinations of normally distributed random variables (independent or not) are normally distributed.

16 Discriminant Functions for the Normal Density I

Linear machine Classifier that uses a linear discriminant function.

Minimum-distance-classifier To classify a feature vector $\mathbf{x}$ measure the Euclidean distance $\|\mathbf{x} - \mu_\mathbf{i}\|$ from each $\mathbf{x}$ to to each of the c mean vectors, and assign $\mathbf{x}$ to the category of the nearest mean.

- Discriminant functions for the normal density if $\Sigma_i = \sigma^2 I$: the features are statistically independent and each features has variance σ^2. Samples fall into equal-size hyper-spherical clusters, the cluster of the ith class is centred around the mean $\mu_\mathbf{i}$. Filling $\mu_\mathbf{i}$ and Σ into Equation 14.3 we get the discriminant functions:
$$g_i(\mathbf{x}) = -\frac{\|\mathbf{x} - \mu_\mathbf{i}\|^2}{2\sigma^2} + \ln P(\omega_i) \tag{16.1}$$
where $\|\mathbf{x} - \mu_\mathbf{i}\| = (\mathbf{x} - \mu_\mathbf{i})^T(\mathbf{x} - \mu_\mathbf{i})$.
- If $\mathbf{x}$ is equally near two different mean vectors it follows from Equation 16.1 that the optimal decision will favour the a priori more likely category.
- The equivalent linear discriminant function based on Equation 16.1 is:
$$g_i(\mathbf{x}) = \mathbf{w_i}^T\mathbf{x} + \omega_{io}$$
where
$$\mathbf{w_i} = \frac{1}{\sigma^2}\mu_\mathbf{i}$$
$$\omega_{io} = -\frac{1}{2\sigma^2}\mu_\mathbf{i}{}^T\mu_\mathbf{i} + \ln P(\omega_i).$$
ω_{io} is called the threshold or bias of the ith category.
- The decision surfaces for a linear machine are pieces of hyperplane defined by linear equations $g_i(\mathbf{x}) = g_j(\mathbf{x})$ for the two categories with the highest posterior probabilities.
- If the prior probabilities for each category are the same you get a minimum-distance-classifier.
- The hyperplane separating two classes orthogonal to the line between their means.

17 Discriminant Functions for the Normal Density II

- Discriminant functions for the normal density if $\Sigma_i = \Sigma$: each class has the same, but otherwise arbitrary, covariance matrix. Samples fall into equal-size hyper-ellipsoidal clusters of equal size and shape, the cluster of the ith class is centred around the mean $\mu_\mathbf{i}$. Filling $\mu_\mathbf{i}$ and Σ into Equation 14.3 we get the discriminant functions:
$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_\mathbf{i})^T\Sigma^{-1}(\mathbf{x} - \mu_\mathbf{i}) + \ln P(\omega_i) \tag{17.1}$$
- If the prior probabilities are the same for each class the optimal decision rule is a minimum-distance-classifier that uses the Mahalanobis distance.
- The equivalent linear discriminant function based on Equation 17.1 is:
$$g_i(\mathbf{x}) = \mathbf{w_i}^T\mathbf{x} + \omega_{io}$$
where
$$\mathbf{w_i} = \Sigma^{-1}\mu_\mathbf{i}$$
$$\omega_{io} = -\frac{1}{2}\mu_\mathbf{i}{}^T\Sigma^{-1}\mu_\mathbf{i} + \ln P(\omega_i).$$

18 Discriminant Functions for the Normal Density II

- Discriminant functions for the normal density if Σ_i is arbitrary are inherently quadratic and we can only drop the $^d/_2 \ln 2\pi$ term from Equation 14.3:

$$g_i(\mathbf{x}) = \mathbf{x}^T \mathbf{W_i} \mathbf{x} + \mathbf{w_i}^T \mathbf{x} + \omega_{io} \tag{18.1}$$

where

$$\mathbf{W_i} = -\frac{1}{2} {\Sigma_i}^{-1}$$

$$\mathbf{w_i} = {\Sigma_i}^{-1} \mu_\mathbf{i}$$

$$\omega_{io} = -\frac{1}{2} {\mu_\mathbf{i}}^T {\Sigma_i}^{-1} \mu_\mathbf{i} - \frac{1}{2} \ln|\Sigma_i| + \ln P(\omega_i).$$

- In the two-category case the decision surfaces are hyper-quadrics.

19 Classification Error

- If a dichotomizer has divided the space into two regions $\mathcal{R}_1$ and $\mathcal{R}_2$ the probability of error is the sum of the grey and pink areas in Figure 19.1:

$$\begin{aligned} P(\text{error}) &= P(x \in \mathcal{R}_2, \omega_1) + P(x \in \mathcal{R}_1, \omega_2) \\ &\quad + P(x \in \mathcal{R}_2 | \omega_1) P(\omega_1) + P(x \in \mathcal{R}_1 | \omega_2) P(\omega_2) \\ &= \int_{\mathcal{R}_2} p(x|\omega_1) P(\omega_1) dx + \int_{\mathcal{R}_1} p(x|\omega_2) P(\omega_2) dx \end{aligned}$$

- With a polychotomizer of c classes it is simpler to compute the probability of being correct:

$$\begin{aligned} P(\text{correct}) &= \sum_{i=1}^{c} P(\mathbf{x} \in \mathcal{R}_i, \omega_i) \\ &= \sum_{i=1}^{c} P(\mathbf{x} \in \mathcal{R}_i | \omega_i) P(\omega_i) \\ &= \sum_{i=1}^{c} \int_{\mathcal{R}_i} p(\mathbf{x}|\omega_i) P(\omega_i) dx \end{aligned}$$

20 Maximum Likelihood Estimation

- Maximum likelihood estimation of the parameters of class conditional probabilities tries to estimate the parameters of a function of a known type e.g. μ_i and Σ_i of a normal density to compute the priors and class conditional probabilities.
- Use a data set $\mathcal{D} = \{\mathbf{x}_1 \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n feature vectors drawn independently from the probability density $p(\mathbf{x}|\theta)$ to estimate the unknown parameter θ. Because the samples were drawn independently we have:

$$p(\mathcal{D}|\theta) = \prod_{k=1}^{n} p(\mathbf{x}_k|\theta).$$

$p(\mathcal{D}|\theta)$ is the likelihood of θ with respect to $\mathcal{D}$. The maximum likelihood of θ is the value $\hat{\theta}$ that maximizes $p(\mathcal{D}|\theta)$. If we have to estimate p parameters we denote θ the p-component vector and we let ∇_θ be the gradient

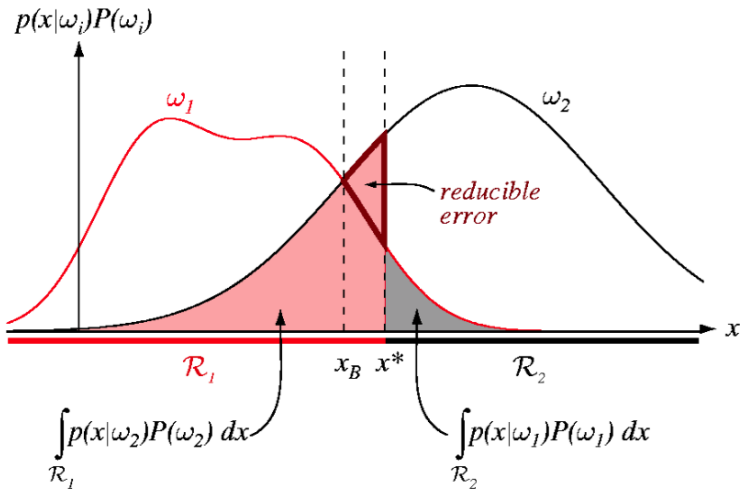


Figure 19.1: *Classification error in one dimension*

operator:

$$\nabla_\theta = \begin{bmatrix} \frac{\partial}{\partial\theta_1} \\ \vdots \\ \frac{\partial}{\partial\theta_p} \end{bmatrix}$$

The log-likelihood function, customarily with base e:

$$l(\theta) = \ln p(\mathcal{D}|\theta) \tag{20.1}$$
$$= \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\theta).$$

We can then write our solution as the argument θ that maximizes (20.1):

$$\hat{\theta} = \arg\max_\theta l(\theta)$$

Thus a set of necessary conditions of the maximum-likelihood estimate for θ can be obtained from the set of p equations:

$$\nabla_\theta l = 0, \tag{20.2}$$

where $\nabla_\theta l$ is defined as:

$$\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta \ln p(\mathbf{x}_k|\theta).$$

- A solution $\hat{\theta}$ to (20.2) could represent a true global maximum, but also a local maximum or minimum.

21 Decision Boundary for Normal Distributions

Absolutely unbiased An estimator that is unbiased for all distributions is absolutely unbiased.

Asymptotically unbiased An estimator that tends to become unbiased as the number of samples becomes very large is asymptotically unbiased.

- One can use the log-likelihood instead of the likelihood because the logarithm is monotonically increasing the $\hat{\theta}$ that maximizes the log-likelihood

also maximizes the likelihood.
- Only if we have infinite large number of training points we find the true value of the generating function.
- The maximum-likelihood estimate for the variance σ^2 is biased, that is the expected value over all data sets of size n of the sample variance is not equal to the true variance. Consider the extreme case with non-zero variance σ^2 and $n = 1$, the mle of the variance will be zero.
- Elementary unbiased estimator for the covariance matrix:
$$C = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\mu})(\mathbf{x}_k - \hat{\mu})^T. \tag{21.1}$$
- Optimal decision boundary can be computed by assuming equal probabilities and then solving the equation:
$$p(\omega_1) = p(w_2)$$
$$\mathcal{N}(\mu_1, \Sigma_1) = \mathcal{N}(\mu_2, \Sigma_2)$$
-

22 Non-parametric Classification Techniques

Parametric classifier The form of the underlying density is known or assumed to be of a certain type which has certain parameters.

Non-parametric classifier No (explicit) assumption is made about the underlying probability density.

k-nn classification rule Select the class that is most frequently among the k nearest neighbours.

- Two possible approaches to non-parametric classification:
  - Estimate the density functions $p(\mathbf{x}|\omega_j)$ from the training data.
  - Estimate the posterior probabilities $P(\omega_j|\mathbf{x})$ directly.
- A good density estimation for n samples from a normal distribution is:
$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n} \tag{22.1}$$
where V_n is a volume around $\mathbf{x}$ and k_n is the number of samples observed in that volume.
- Equation 22.1 converges i.e. $p_n(\mathbf{x}) \to p(\mathbf{x})$ iff:
  - $\lim_{n\to\infty} V_n = 0$
  - $\lim_{n\to\infty} k_n = \infty$
  - $\lim_{n\to\infty} k_n/n = 0$
- Two approaches to density estimation:

  Parzen windows Choose V_n in (22.1) fixed e.g. $V_n = \sqrt{n}$.

  k_n **nearest neighbours** Grow V_n in (22.1) till it includes a fixed numbers of neighbours k_n, e.g. $k_n = \sqrt{n}$.
- Large V_n with Parzen windows leads to low resolution, small V_n leads to statistical variability.
- Suppose that a cell V around $\mathbf{x}$ contains k samples of which k_i from class ω_i an estimate for $p(\mathbf{x}|\omega_i)P(\omega_i)$ is given by:
$$p_n(\mathbf{x}|\omega_i)P_n(\omega_i) = \frac{k_i/n_i}{V}\frac{n_i}{n} = \frac{k_i/n}{V}$$

Thus

$$P_n(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x},\omega_i)P(\omega_i)}{\sum_{j=1}^{c} p_n(\mathbf{x},\omega_j)P(\omega_j)} = \frac{k_i}{k}$$

- The estimate of the posterior probability that a new vector $\mathbf{x}$ belongs to a given class ω_i is the fraction k_i/k of training vectors from that class within the concerned cell.
- Advantages of 1-NN rule: simple, no training time, works well for almost separable classes, useful to shape non-linear boundaries.
- Disadvantages of 1-NN rule: long execution time, all data should be stored, scaling dependent (depends on distance measure), the classifier is over-trained.

23 Non-parametric Classification Techniques II

Discriminative classifier Classifier that either estimates posterior probabilities directly or determines a decision function.

Nearest neighbour classification rule Assign a new feature vector $\mathbf{x}$ to the class ω_i if the closest prototype belongs to ω_i.

Condensing Reduce the number of training samples by retaining only samples that are needed to define the decision boundary.

Editing Remove points that do not agree with the majority of their k nearest neighbours.

- Comparison of k-NN with Bayesian classification:

k-NN	Bayesian classification
+ No assumptions about the distributions.	- Assumptions about the distributions.
- To classify a new point all distance have to be recomputed.	+ Once the parameters of the distribution are estimated we only need to evaluate the discriminant function.

- Comparison of k-NN with LVQ:

k-NN	LVQ
+ Uses all training data	
+ Good theoretical foundation	- Based on experience and heuristics.
- To classify a new point all distance have to be recomputed.	+ Only compute distances to prototypes.

- Reducing the complexity of k-nn:

  Compute the partial distances Compute distances in a selected number of dimensions r, only compute further distances if the partial distance is greater than the full distance to the current nearest prototype.

  Pre-structuring Create an additional data structure, e.g. search tree, in which the prototypes are linked according to some criteria.

  Editing stored prototypes Reduce the number of prototypes by eliminating the unnecessary ones (pruning or condensing).

- Minkowski distance:
$$L_k(a,b) = \sqrt[k]{\sum_{i=1}^{d} |a_i - b_i|^k}$$
- Special cases of the Minkowski difference:
  - $k = 1 \rightarrow$ Manhattan distance: $L_1(a,b) = \sum_{i=1}^{d} \|a_i - b_i\|$
  - $k = 2 \rightarrow$ Euclidean distance: $L_2(a,b) = \sqrt{\sum_{i=1}^{d} |a_i - b_i|^2}$
  - $k = \infty$: $L_\infty(a,b) = \max_{i=1}^{d} \|a_i - b_i\|$
- Non-parametric classification needs a lot of samples for good results.

24 Receiver Operating Characteristic Curve

ROC curve Plot of p(hit) versus p(false alarm), see Figure 24.1.
OC curve Plot of p(false alarm) versus p(hit).

- Discriminability describes the inherent and unchangeable properties due to noise and the strength of the external signal, but not the decision strategy:
$$d' = \frac{|\mu_2 - \mu_1|}{\sigma} \tag{24.1}$$
We assume that that the distributions are normal with different means but the same variance, i.e. $p(x|\omega_i) \sim \mathcal{N}(\mu_i, \sigma^2)$.
- Classifying the classifications of a system:

  Hit $p(x > x^*|x \in \omega_2)$ the probability that the internal signal is above x^* given that the external signal is present.

  False alarm $p(x > x^*|x \in \omega_1)$ the probability that the internal signal is above x^* given that the external signal is absent.

  Miss $p(x < x^*|x \in \omega_2)$ the probability that the internal signal is below x^* given that the external signal is present.

  Correct rejection $p(x < x^*|x \in \omega_1)$ the probability that the internal signal is below x^* given that the external signal is absent.
- For an optimal Bayes classifier the values of hit rate and false alarm rate should add up to one, that points lies at the crossing of the ROC curve and the line that connects the points (0,1) and (1,0), the blue line in Figure 24.1.
- OC curves are determined experimentally by changing some parameter of a classifier and measuring the hit and false alarm.
- OC curves are used to compare classifiers.

25 Learning Vector Quantization

- LVQ identifies prototype vectors from labelled training data and uses distance based classification to classify new data.
- LVQ formally: Given a set of prototypes $w_1, w_2, \ldots w_k$ with $k \in \mathcal{R}$ representing class $\omega_1, \omega_2, \ldots \omega_k$ with $\omega_k \in \{1, 2, \ldots C\}$ where C is the number of classes. Given feature vector ξ determine the winner $w_{i^*} = \arg\min_j \{d[w_j, \xi]\}$ and assign ξ to the class ω_{i^*}. $d[w, \xi]$ is the distance between the prototype w and the feature vector ξ.

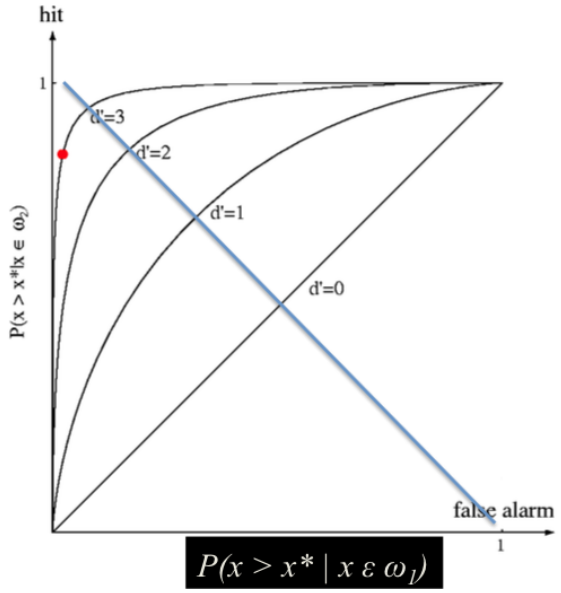
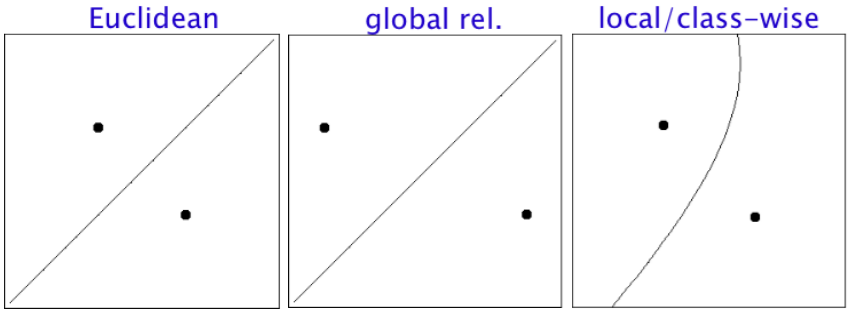


Figure 24.1: *Example of an ROC curve, the red dot denotes a pair of experimentally determined values of false alarm and hit.*

- LVQ1 training: Start with randomized w_k for example close to the class-conditional means. Present the labelled examples (ξ_t, σ_t) sequentially. For each labelled example: determine the nearest prototype w_{i^*}:
  $$w_{i^*} = \arg\min_j \{d[w_j, \xi_t]\} \tag{25.1}$$
  where $d[\cdot]$ is a distance measure between w_j and ξ_t. Update the winning prototype according to:
  $$w_{i^*} = w_{i^*} + \eta_w \cdot \psi(\omega_{i^*}, \sigma_t)(\xi_t - w_{i^*}) \tag{25.2}$$
  Where η_w is the learning rate and $\psi(\cdot)$ is defined as:
  $$\psi(\omega_{i^*}, \sigma_t) = \begin{cases} 1 & \omega_{i^*} = \sigma_t \\ -1 & \text{otherwise} \end{cases}$$
- LVQ algorithms are often purely heuristic arguments or cost functions with unclear relation to classification error.
- RLVQ adapts the distance measures during learning. The winning prototype is determined according to (25.1), w_{i^*} is updated according to (25.2). Each global relevance is updated according to:
  $$\lambda_j = \lambda_j - \eta_\lambda \cdot \psi(\omega_{i^*}, \sigma_t)|\xi_t - w_{i^*}|.$$
  The global relevances need to be a convex combination.
- The relevances are used in the computation of the distances, e.g. Euclidean distance becomes:
  $$d_\lambda[w_i, \xi] = \sum_{j=1}^{n} \lambda_j (w_{ji} = \xi_j)^2$$
- Different relevances and their requirements:



(a) *No relevances* **(b)** *Global relevances* **(c)** *Local relevances*

Figure 25.1: *Learning without relevances and with different relevances.*

global	$\lambda_j \geq 0$	$\sum_{j=1}^{n} \lambda_j = 1$	
local	$\lambda_{ji} \geq 0$	$\sum_{j=1}^{n} \lambda_{ji} = 1$	$i \in [1, K]$
class-wise	$\lambda_{jc} \geq 0$	$\sum_{j=1}^{n} \lambda_{jc} = 1$	$c \in [1, C]$

where C is the number of classes and K is the number of prototypes. See

26 Cross validation

Test set method Take a subset of your data as test set and train on the rest. This method is very simple but wastes a part of your data and the results depend in part on luck.

Leave One Out Cross Validation (LOOCV) If you have a dataset of n points train n times on a dataset of $n-1$ point and test with the left out point, report the mean of the error of the different iterations. This method is expensive and has some weird behavior, but does not waste data.

k-fold cross validation Randomly break the dataset in k partitions, for each partition t train on the others and test on t and report the mean error of the partitions.

- k-fold cross validation for $k \neq n$ wastes less more data than LOOCV but is also not as expensive.
- When using cross validation for classification compute the total number of misclassifications on the test set.

27 Clustering

Receptive field of x The cluster associated with x.

k-centres selects existing objects as prototypes contrary to k-means that constructs prototypes not existing in the training set.

- Goal of clustering: division of a data set X into k disjoint subset $C_1, \ldots, C_k$ such that the objects within each subset are similar and objects in different subset are dissimilar.
- k-means clustering: Given element $\mathbf{x_j} \in \mathcal{R}^n$ and a number of clusters, k find k prototypes $\mu_\mathbf{1}, \ldots, \mu_\mathbf{k} \in \mathcal{R}^n$ that minimize the quantization error:
$$J_e = \frac{1}{2} \sum_{\mu_\mathbf{i}} \sum_{\mathbf{x_j} \in C(\mu_\mathbf{i})} \|\mathbf{x_j} - \mu_\mathbf{i}\|^2 \tag{27.1}$$
where $C(\mu_\mathbf{i})$ denotes the receptive field of μ_i and is defined as:
$$C(\mu_\mathbf{i}) = \{\mathbf{x}_j | \|\mathbf{x}_j - \mu_\mathbf{i}\| < \|\mathbf{x}_j - \mu_\mathbf{m}\| \forall_m m \neq i\} \tag{27.2}$$
- Lloyd's algorithm for k-means clustering:
  1. Initialize $\mu_\mathbf{1}, \ldots, \mu_\mathbf{k}$ with k random samples from the data set.
  2. do
     (a) Assign data points to the nearest cluster centre $\mu_\mathbf{i}$.
     (b) Compute C_i
     (c) Recompute $\mu_\mathbf{i}$ as the mean of the points in C_i.
     until no change in $\mu_\mathbf{1}, \ldots, \mu_\mathbf{k}$.
- Lloyd's algorithm for k-means clustering converges in a finite number of steps because a non-negative cost function, the quantization error, decreases or remains constant with each step. However there is no guarantee that a global minimum is reached.
- Choosing k: Compute the quantization error (27.1) $J(k)$ of the data as a function of k. Also compute the quantization error $R(k)$ for a uniformly distributed reference data set. Define $D(k)$ as:
$$D(k) = \frac{R(k)}{J(k)}$$
The optimal k, k_opt, is the maximum of the ratio $D(k)$:
$$k_{opt} = \arg\max_k D(k)$$
- Problems with k-means clustering:

  Dead prototypes If some prototypes are initialized far away from the data set no data points are assigned to them and they are never used.

  Non spherical clusters

  Local optima

28 Fuzzy k-means clustering

- Fuzzy k-means allows points to belong to different clusters.
- Fuzzy k-means normalizes cluster membership values so that they form an affine combination.
- Fuzzy k-means tries to minimize the heuristic global cost function:
$$J_{fuzzy} = \sum_{i=1}^{k} \sum_{j=1}^{n} [\hat{P}(\omega_i | x_j)]^m \|x_j - \mu_i\|^2 \quad 1 \leq m \leq \infty,\ m \in \mathcal{R}.$$
$x_1, \ldots x_n$ are that data points and $\omega_1, \ldots, \omega_k$ are the clusters and μ_i is the cluster centre of cluster ω_i. And m is the fuzziness argument. $[\hat{P}(\omega_i|x_j)]$ denotes the membership of vector x_j to cluster ω_i.
- Larger values of the fuzziness argument lead to fuzzier clusters.

- Fuzzy k-means clustering defines centres as:
$$\mu_i = \frac{\sum_{j=1}^{n}[\hat{P}(\omega_i|x_j)]^m \cdot x_j}{\sum_{j=1}^{n}[\hat{P}(\omega_i|x_j)]^m}$$
- Fuzzy k-means computes cluster membership according to:
$$\hat{P}(\omega_i|x_j) = \frac{1}{\sum_{r=1}^{k}(^{d_{ij}}/_{d_{rj}})^{2/(m-1)}}$$
where $d_ij = \|x_j - \mu_i\|$
- Fuzzy k-means algorithm:
  1. Initialize n, k, $\mu_1, \ldots, \mu_k$, $\hat{P}(\omega_i|x_j)$.
  2. do
     (a) normalize $\hat{P}(\omega_i|x_j)$
     (b) re-compute μ_i
     (c) re-compute $\hat{P}(\omega_i|x_j)$

     until there are only small changes in μ_i and $\hat{P}(\omega_i|x_j)$.
- Fuzzy k-means, with $m \neq 1$, has better convergence properties than k-means.
- When using fuzzy k-means the membership depends implicitly on the number of clusters, thus wrong $k \rightarrow$ wrong clustering.

29 Optimization by Gradient Methods

- Online k-means = vector quantization:
  1. Initialize $\mu_1, \ldots \mu_k$
  2. repeat
     (a) Choose $\mathbf{x}_j$
     (b) Determine closest μ_i, the winner.
     (c) Update the winner according to: $\mu_i = \mu_i + \eta \cdot (\mathbf{x}_j - \mu_i)$
- Contrary to normal k-means online k-means can be used for streaming data, it has however the same drawbacks as 'normal' k-means.
- The rank of data point $\mathbf{x}_j$ with respect to prototype ω_i:
$$k_{ij} = k_i(\mathbf{x}_j, \omega) = |\{\omega_l | d(\mathbf{x}_j, \omega_l) < d(\mathbf{x}_j, \omega_i)\}| \tag{29.1}$$
- Update rule of neural gas:
$$\omega_i = \frac{\sum_{j=1}^{n} h_\lambda(k_{ij})\mathbf{x}_j}{\sum_{j=1}^{n} h_\lambda(k_{ij})} \tag{29.2}$$
where $h_\lambda(t) = e^{t/\lambda}$
- Neural Gas epoch:
  1. for each prototype ω_i
     (a) for each datapoint $\mathbf{x}_j$
        i. determine the rank k_{ij}, (29.1)
     (b) update the prototype ω_i, (29.2)
- Neural gas updates all prototypes instead of only the winner as is done with vector quantization.
- k-means tries to minimize the quantization error (27.1), neural gas tries

to find prototypes such that they minimize the cost term:

$$J_e = \frac{1}{2} \sum_{\mu_i} \sum_{\mathbf{x}_j} \exp\left(-\frac{r(\mu_i, \mathbf{x}_j)}{\lambda^2}\right) \cdot \|\mu_i - \mathbf{x}_j\|^2$$

where $r(\cdot)$ is the rank function from (29.1), λ is a scaling parameter that is decreased every epoch.

30 Whitening Transform and Gaussianity

- Linear combinations of normally distributed random variables, independent or not, are normally distributed.
- The whitening transform of the covariance matrix Σ

  $$A_w = \Phi \Lambda^{-1/2}$$

  leads to a covariance matrix that is equal to the identity matrix. Where Φ is the matrix whose columns are the orthonormal eigenvectors of Σ and Λ a diagonal matrix is with the corresponding eigenvalues.
- Kurtosis:

  $$\text{kurt}(y) = \frac{E[(y-\mu)^4]}{\sigma^4} - 3$$

  is a measure for the gaussianity of a signal, if $\text{kurt}(y) = 0$ y is Gaussian.
- Kurtosis is very sensitive to outliers and is thus not a robust measure of non-Gaussianity.
- Negentropy:

  $$H(Y) = -\sum_j P(Y) \log P(y) \tag{30.1}$$

  can be used to measure gaussianity of a variable.
- Out of all distributions with a given mean and variance, the normal or Gaussian distribution is the one with the highest entropy.

31 Hierarchical Clustering

Spanning tree A weighted graph connecting all vertices without loops.
Minimal spanning tree A spanning tree with minimum total weight.
Generalized distance function $d(x, y)$ The lowest dissimilarity value for which x and y are in the same cluster.

- Possible ways of defining cluster dissimilarity:

  Single linkage by the two nearest object in the cluster:

  $$d_{min}(D_i, D_j) = \min_{i \in D_i, j \in D_j} \|i - j\|$$

  Complete linkage by the two farthest objects in the cluster:

  $$d_{max}(D_i, D_j) = \max_{i \in D_i, j \in D_j} \|i - j\|$$

  Average linkage by the average distance:

  $$d_{avg}(D_i, D_j) = \frac{1}{n_i \cdot n_j} \sum_{i \in D_i} \sum j \in D_j \|i - j\|$$

- If the weights of a MST are derived from a dissimilarity matrix the MST is identical to the hierarchy defined by single linkage.
- Clustering using a MST:

  - Split the tree by removing the edge with the greatest weight until the required number of clusters is achieved.
  - Remove inconsistent edge, e.g. edges whose length exceeds twice the average length of the other edges incident to a node.

32 Independent Component Analysis

- Independent Component Analysis (ICA) finds linear combinations of the original features, such that the new features are statistically independent, if the source signals are statistically independent and non-Gaussian.
- Assume observations of n variables $x_1, \ldots, x_n$ that are linear mixtures of $m \leq n$ unknown independent variables $s_1, \ldots, s_m$ called the independent components. Than for each variables x_j

  $$x_i = a_{j1}s_j + \ldots + a_{jm}s_m$$
- The ICA model:

  $$\mathbf{x} = A\mathbf{s} = \sum_{i=1}^{m} \mathbf{a}_i \mathbf{s}_i \tag{32.1}$$
- Independent components are obtained by:

  $$s = W\mathbf{x} = \sum_{i=1}^{n} \mathbf{w}_i \mathbf{x}_i$$
- Two scalar-valued random variables s_1 and s_2 are said to be independent if information about the value of s_1 does not give any information about the value of s_2 and vice versa.
- Covariance matrix is a diagonal matrix $\rightarrow$ features are uncorrelated.
- PCA vs ICA:
  - Both PCA and ICA are linear transformations of the original features to new features.
  - In PCA the axes of the new coordinate system are orthogonal, in ICA they need not be.
  - The goal of PCA is to find a new coordinate system such that the features are uncorrelated, goal of ICA is to find a new coordinate system such that the features are independent.
  - In PCA the first PC gives a new feature for which variance is maximal, but this feature needs not have any relation to an independent source variable.
- In pre-processing Principle Component Analysis (Whitening) can be used to determine the number of independent components if the noise level is low.
- Limitations of ICA:
  - The variances of the independent components can not be determined. Because both $\mathbf{s}$ and A, in (32.1), unknown.
  - The order of the independent components can not be determined.
  - Not all independent components can be derived if the amount of sources is larger than the number of observed mixtures.

33 FastICA

Uncorrelated Two random variables X and Y are uncorrelated when their correlation coefficient is zero.

Independent Two random variables are independent when their joint probability distribution is the product of their marginal probability distributions for all x and y.

Central limit theorem Sums of non-Gaussian random variables are closer to Gaussian then the original ones.

- FastICA to derive one independent component:
  1. Choose initial random weight vector $\mathbf{w}$.
  2. Update the weights according to:
     $$w = E\mathbf{x} \cdot g(w^T\mathbf{x}) - Eg'(w^T\mathbf{x})w$$
  3. Normalize the weights.
  4. If the system has not converged go back to two.

  The function (g) is the derivative of the contrast function.
- To obtain multiple independent components, the one-unit FastICA algorithm can be used using multiple weight vectors. To prevent different vectors from converging to the same maxima, the outputs ${w_1}^T\mathbf{x}, \ldots, {w_n}^T\mathbf{x}$ have to be decorrelated after every iteration.
- FastICA compared to classical ICA methods:
  - FastICA converges fast.
  - FastICA finds directly independent component without a known PDF.
  - Performance of FastICA can be optimized by changing the function $g(\cdot)$.
  - Independent components can be estimated one by one.

34 Support Vector Machines

Maximum margin linear classifier LSVM: the linear classifier with the maximum margin.

Support Vectors Data points that the maximum margin pushes up against.

- SVMs pre-process data to represent patterns in a high dimension, typically much higher than the original feature space. Using the fact that with an appropriate non-linear mapping to a sufficiently high dimension data from two categories can always be separated by a hyperplane.
- The margin of a linear classifier is the width that the boundary could be increased before hitting a data point.
- Reasons to use the maximum margin:
  - If we have made a small error in the location of the boundary this gives us least chance of causing a misclassification.
  - The model is immune to removal of any non-support-vector data points.
- The maximum margin $\mathbf{w}\mathbf{x} + b = 0$ where $\arg\max_{\mathbf{w},b} \text{margin}(\mathbf{w}, b, D)$ the margin is defined as the minimum distance between a point $\mathbf{x}$ and the line $\mathbf{w}\mathbf{x} + b = 0 : \min_{\mathbf{x}_i \in D} d(\mathbf{x}_i)$. Where $d(\mathbf{x}_i)$ is the distance from $\mathbf{x}_i$ to the

margin. Actually computing this distance results in:

$$\arg_{\mathbf{w},b} \min_{\mathbf{x}_i \in D} \frac{|b + \mathbf{x}_i \cdot \mathbf{w}|}{\sqrt{\sum_{i=1}^{d} {w_i}^2}}$$

Subject to $\forall \mathbf{x}_i \in D : y_i(\mathbf{x}_i \cdot \mathbf{w} + b) > 0$ where y_i is the class of $\mathbf{x}_i$.

- The previous expression can be reformulated to: find $\mathbf{w}$ such that

$$\arg_{\mathbf{w},b} \min \sum_{i=1}^{d} {w_i}^2$$

Subject to $\forall \mathbf{x}_i \in D : y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$ where y_i is the class of $\mathbf{x}_i$.

- If data are not linearly separable you can slack some variables to allow misclassification of difficult or noisy data points.