ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

Αναγνώριση Προτύπων

Baysian Θεωρία Αποφάσεων
ΕΠΙΣΚΟΠΗΣΗ-ΑΣΚΗΣΕΙΣ

Χριστόδουλος Χαμζάς

Τα περιεχόμενο της παρουσίασης βασίζεται στο βιβλίο: "Introduction to Pattern Recognition A Matlab Approach", S. Theodoridis, A. Pikrakis, K. Koutroubas, D. Caboyras, Academic Press, 2010

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Μπευζιανή θεωρία αποφάσεων

Στα προηγούμενα μάθηματα συζητήσαμε τις τεχνικές βασιζόμενοι στη θεωρία αποφάσεων Bayes και παρουσιάστηκαν οι θεωρητικές αποδείξεις των σχετικών αλγορίθμων. Οι περισσότεροι από τους αλγόριθμους είναι απλοί τόσο στη δομή όσο και στην φυσική αιτιολόγηση τους.

Σε μια εργασία ταξινόμησης (αναγνώρισης προτύτου), μας δίνεται ένα ΠΡΟΤΥΠΟ και θέλουμε να το κατατάξουμε σε μία από C κλάσεις (κατηγορίες) ω_i. Ο αριθμός των κλάσεων, C, θεωρείται ότι είναι γνωστός εκ των προτέρων.

Κάθε πρότυπο αντιπροσωπεύεται από ένα σύνολο χαρακτηριστικών, x (i), i = 1, 2,. . ., d, το οποίο κάνει τον d-διάστατο χαρακτηριστικό διάνυσμα $x = [x(1), x(2), . . . , x(d)]^T \in R^d$
Υποθέτουμε ότι κάθε πρότυπο είναι μοναδικό και εκπροσωπείται από ένα μόνο χαρακτηριστικό διάνυσμα και ότι μπορεί να ανήκει σε μία μόνο κατηγορία ω_i.
Δεδομένου $x \in R^d$ και ένα σύνολο κατηγοριών C, ω_i , i = 1, 2,. . . , C, η θεωρία Bayes δηλώνει ότι

$$P(\omega_i \mid x) = \frac{p(x \mid \omega_i) P(\omega_i)}{\sum_{i=1}^{C} p(x \mid \omega_i) P(\omega_i)} = \frac{p(x \mid \omega_i) P(\omega_i)}{p(x)}$$

Όπου $P(\omega_i)$ είναι η εκ των προτέρων (a priori) πιθανότητα της κλάσης ω_i : i = 1, 2,. . . C,
$P(\omega_i \mid x)$ είναι η εκ των υστέρων πιθανότητας (aposteriori) της κλάσης ω_i δεδομένης του x
p (x) είναι η συνάρτηση πυκνότητας πιθανότητας (pdf) του x
και $p(x|\omega_i)$, i = 1 = 2,. . . , C, είναι η υπο συνθήκη pdf του x δεδομένου ω_i (μερικές φορές καλείται η δεσμευμένη πιθανότητα του x ως προς ω_i ή η πιθανοφάνεια (likelihood) του ω_i σε σχέση με το x.

Σημείωση: Με **P(.)** συμβολίζουμε πιθανότητες ενώ με **p(.)** συναρτήσεις πυκνότητας πιθανότητας (pdf)

Κανόνας απόφασης Bayes

Το πρότυπο x, το ταξινομούμε στην κλάση που έχει τη μεγαλύτερη εκ των υστέρων πιθανότητα !!!

Δηλαδή έαν $\mathbf{x}=\{x_1, x_1, \ldots, x_d\}$ *τότε το* **x** αναγνωρίζεται ότι ανήκει στην κλάση ω_i εάν

$P(\omega_i\,|\mathbf{x}) \geq P(\omega_j\,|\mathbf{x})$ για όλα τα $j=1,2,\ldots,C$

και

$P_e = P(error) = 1 - \max\,[P(\omega_1\,|\mathbf{x}), P(\omega_2\,|\mathbf{x}), \ldots, P(\omega_c\,|\mathbf{x})]$

Η επιλογή αυτή είναι βέλτιστη υπο την έννοια ότι ελαχιστοποιεί την ολική πιθανότητα λάθους

Κανονική Κατανομή (Gauss)

Η Gaussian pdf, χρησιμοποιείται ευρέως στην αναγνώριση προτύπων, λόγω της μαθηματικής ευκολίας χειρισμού της καθώς και λόγω του κεντρικού οριακού θεωρήματος, το οποίο δηλώνει ότι το pdf του αθροίσματος μιας σειράς στατιστικά ανεξάρτητων τυχαίων μεταβλητών τείνει στην κατανομή Gauss καθώς ο αριθμός των τυχαίων μεταβλητών τείνει στο άπειρο. Στην πράξη, αυτό περίπου ισχύει και για έναν αρκετά μεγάλο αριθμό των τυχαίων μεταβλητών.

Η πολυδιάστατη Gaussian pdf έχει τη μορφή

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu) \right)$$

Όπου **μ=E(x)** είναι η μέση τιμή του διανύσματος **x**

Και **Σ** ο πίνακας συνδιασποράς (covariance)

$$\sum = E\left[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T \right] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1D} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{1D} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{D1} & \sigma_{D2} & \cdots & \sigma_{DD} \end{bmatrix}$$

Συχνά αναφερόμαστε στην Gaussian κατανομή σαν **κανονική** κατανομή και την συμβολίζουμε με το **N(μ, Σ).**

Για την περίπτωση της μιάς διάστασης, δηλαδή το x ϵ R, έχουμε **N(μ, σ)** ή **N(μ, σ^2)** με **σ^2** την διασπορά του **x.**
Εάν δεν είναι προφανές από τον συμβολισμό τότε θεωρούμε ότι χρησιμοποιείται το N(μ, σ^2)

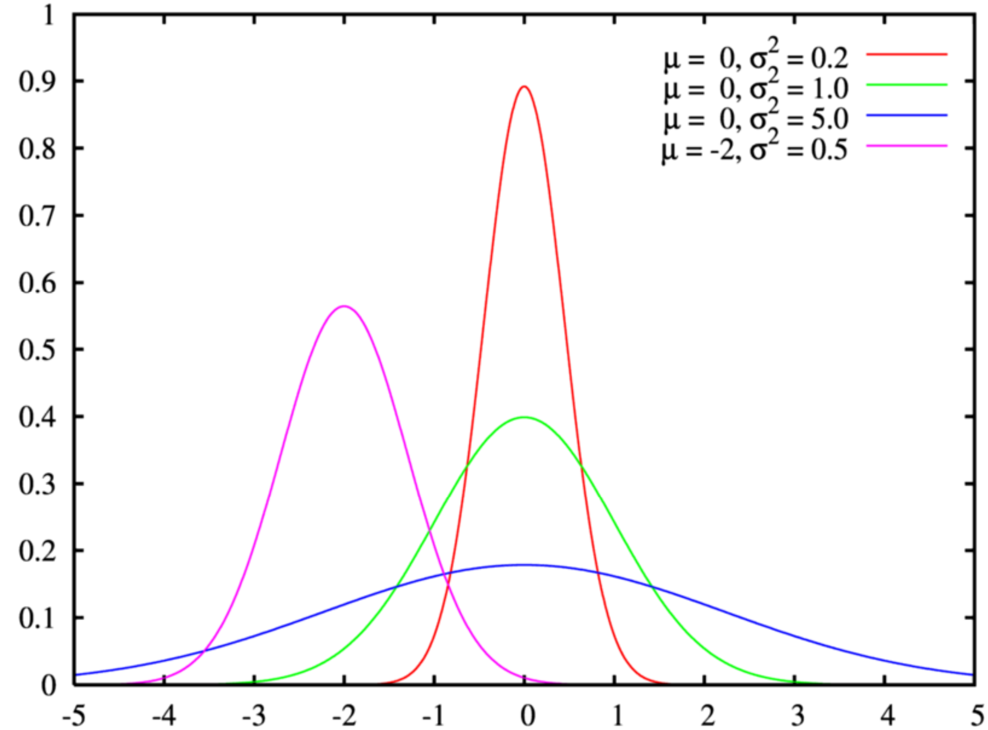
1 D Κανονική Κατανομή (Gauss)

$$N(\mu,\sigma) = p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$E\{1\} = \int_{-\infty}^{\infty} p(x)dx = 1$$

$$E\{x\} = \int_{-\infty}^{\infty} xp(x)dx = \mu$$

$$E\{(x-\mu)^2\} = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx = \sigma^2$$



$$\Pr(|x-\mu| \le \sigma) \cong 0.68,\ \Pr(|x-\mu| \le 2\sigma) \cong 0.95,\ \Pr(|x-\mu| \le 3\sigma) \cong 0.997,$$

Example 1.3.1. Compute the value of a Gaussian pdf, $\mathcal{N}(m, S)$, at $x_1 = [0.2, 1.3]^T$ and $x_2 = [2.2, -1.3]^T$, where

$$m = [0,\ 1]^T, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Solution. Use the function *comp_gauss_dens_val* to compute the value of the Gaussian pdf. Specifically, type

```
m=[0 1]'; S=eye(2);
x1=[0.2 1.3]'; x2=[2.2 -1.3]';
pg1=comp_gauss_dens_val(m,S,x1);
pg2=comp_gauss_dens_val(m,S,x2);
```

The resulting values for $pg1$ and $pg2$ are 0.1491 and 0.001, respectively. ∎

Example 1.3.2. Consider a 2-class classification task in the 2-dimensional space, where the data in both classes, ω_1, ω_2, are distributed according to the Gaussian distributions $\mathcal{N}(m_1, S_1)$ and $\mathcal{N}(m_2, S_2)$, respectively. Let

$$m_1 = [1,\ 1]^T, \quad m_2 = [3,\ 3]^T, \quad S_1 = S_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Assuming that $P(\omega_1) = P(\omega_2) = 1/2$, classify $x = [1.8,\ 1.8]^T$ into ω_1 or ω_2.

Solution. Utilize the function *comp_ gauss_dens_val* by typing

```
P1=0.5;
P2=0.5;
m1=[1 1]'; m2=[3 3]'; S=eye(2); x=[1.8 1.8]';
p1=P1*comp_gauss_dens_val(m1,S,x);
p2=P2*comp_gauss_dens_val(m2,S,x);
```

The resulting values for $p1$ and $p2$ are 0.042 and 0.0189, respectively, and x is classified to ω_1 according to the Bayesian classifier. ■

Example 1.3.3. Generate $N = 500$ 2-dimensional data points that are distributed according to the Gaussian distribution $\mathcal{N}(m, S)$, with mean $m = [0,\ 0]^T$ and covariance matrix $S = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$, for the following cases:

$\sigma_1^2 = \sigma_2^2 = 1,\ \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 0.2,\ \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 2,\ \sigma_{12} = 0$

$\sigma_1^2 = 0.2,\ \sigma_2^2 = 2,\ \sigma_{12} = 0$

$\sigma_1^2 = 2,\ \sigma_2^2 = 0.2,\ \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 1,\ \sigma_{12} = 0.5$

$\sigma_1^2 = 0.3,\ \sigma_2^2 = 2,\ \sigma_{12} = 0.5$

$\sigma_1^2 = 0.3,\ \sigma_2^2 = 2,\ \sigma_{12} = -0.5$

Plot each data set and comment on the shape of the clusters formed by the data points.

Solution. To generate the first data set, use the built-in MATLAB function *mvnrnd* by typing

```
randn('seed',0) %Initialization of the randn function
m=[0 0]';
S=[1 0;0 1];
N=500;
X = mvnrnd(m,S,N)';
```

where X is the matrix that contains the data vectors in its columns.

To ensure reproducibility of the results, the *randn* MATLAB function, which generates random numbers following the Gaussian distribution, with zero mean and unit variance, is initialized to a specific number via the first command (in the previous code *randn* is called by the *mvnrnd* MATLAB function).

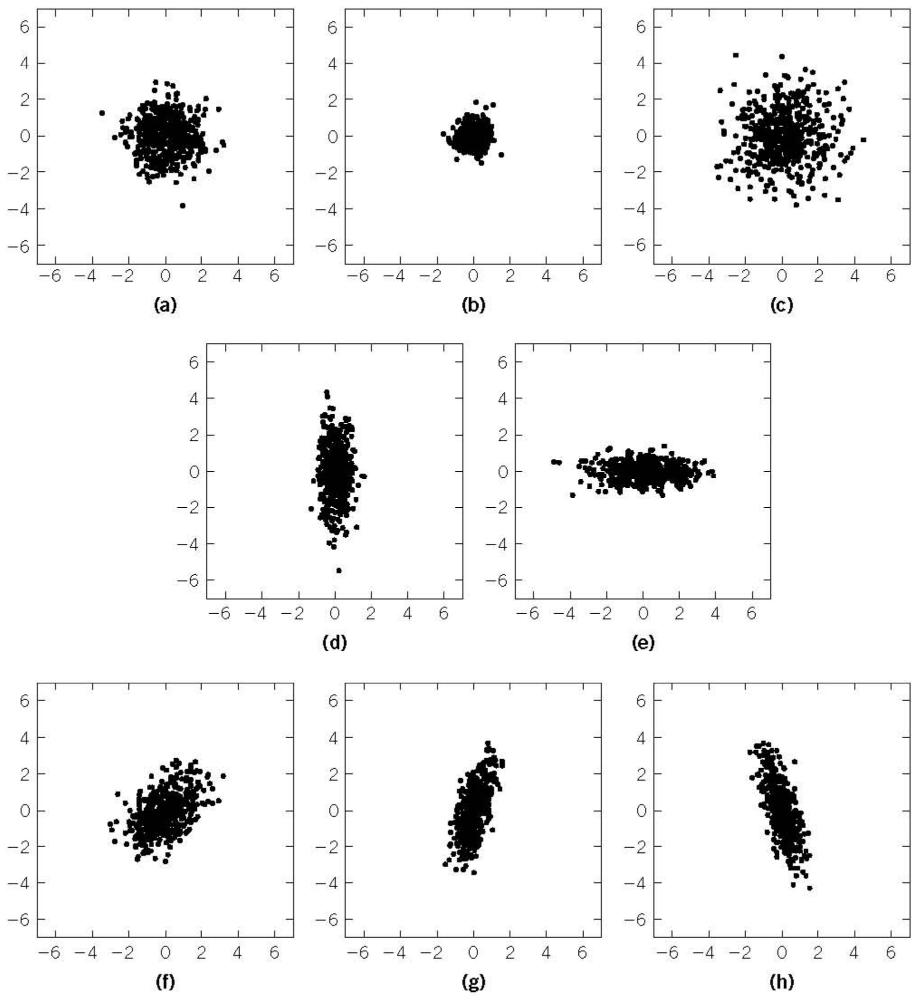
To plot the data set, type

```
figure(1), plot(X(1,:),X(2,:),'.');
figure(1), axis equal
figure(1), axis([-7 7 -7 7])
```

Working similarly for the second data set, type

```
m=[0 0]';
S=[0.2 0;0 0.2];
N=500;
X = mvnrnd(m,S,N)';
figure(2), plot(X(1,:),X(2,:),'.');
figure(2), axis equal
figure(2), axis([-7 7 -7 7])
```

The rest of the data sets are obtained similarly. All of them are depicted in Figure 1.1, from which one can observe the following:


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

- When the two coordinates of x are uncorrelated ($\sigma_{12} = 0$) and their variances are equal, the data vectors form "spherically shaped" clusters (Figure 1.1(a–c)).
- When the two coordinates of x are uncorrelated ($\sigma_{12} = 0$) and their variances are unequal, the data vectors form "ellipsoidally shaped" clusters. The coordinate with the highest variance corresponds to the "major axis" of the ellipsoidally shaped cluster, while the coordinate with the lowest variance corresponds to its "minor axis." In addition, the major and minor axes of the cluster are parallel to the axes (Figure 1.1(d, e)).

- When the two coordinates of x are correlated ($\sigma_{12} \neq 0$), the major and minor axes of the ellipsoidally shaped cluster are no longer parallel to the axes. The degree of rotation with respect to the axes depends on the value of σ_{12} (Figure 1.1(f–h)). The effect of the value of σ_{12}, whether positive or negative, is demonstrated in Figure 1.1(g, h). Finally, as can be seen by comparing Figure 1.1(a, f), when $\sigma_{12} \neq 0$, the data form ellipsoidally shaped clusters despite the fact that the variances of each coordinate are the same.

1.4 MINIMUM DISTANCE CLASSIFIERS

1.4.1 The Euclidean Distance Classifier

The optimal Bayesian classifier is significantly simplified under the following assumptions:

- The classes are equiprobable.
- The data in *all* classes follow Gaussian distributions.
- The covariance matrix is the *same* for all classes.
- The covariance matrix is diagonal and *all* elements across the diagonal are *equal*. That is, $S = \sigma^2 I$, where I is the identity matrix.

Under these assumptions, it turns out that the optimal Bayesian classifier is equivalent to the minimum Euclidean distance classifier. That is, given an unknown x, assign it to class ω_i if

$$||x - m_i|| \equiv \sqrt{(x - m_i)^T (x - m_i)} < ||x - m_j||, \quad \forall i \neq j$$

It must be stated that the Euclidean classifier is often used, even if we know that the previously stated assumptions are not valid, because of its simplicity. It assigns a pattern to the class whose mean is closest to it with respect to the Euclidean norm.

1.4.2 The Mahalanobis Distance Classifier

If one relaxes the assumptions required by the Euclidean classifier and removes the last one, the one requiring the covariance matrix to be diagonal and with equal elements, the optimal Bayesian classifier becomes equivalent to the minimum Mahalanobis distance classifier. That is, given an unknown x, it is assigned to class ω_i if

$$\sqrt{(x - m_i)^T S^{-1} (x - m_i)} < \sqrt{(x - m_j)^T S^{-1} (x - m_j)}, \quad \forall j \neq i$$

where S is the common covariance matrix. The presence of the covariance matrix accounts for the shape of the Gaussians [Theo 09, Section 2.4.2].

Example 1.4.1. Consider a 2-class classification task in the 3-dimensional space, where the two classes, ω_1 and ω_2, are modeled by Gaussian distributions with means $m_1 = [0, 0, 0]^T$ and $m_2 = [0.5, 0.5, 0.5]^T$, respectively. Assume the two classes to be equiprobable. The covariance matrix for both distributions is

$$S = \begin{bmatrix} 0.8 & 0.01 & 0.01 \\ 0.01 & 0.2 & 0.01 \\ 0.01 & 0.01 & 0.2 \end{bmatrix}$$

Given the point $x = [0.1, 0.5, 0.1]^T$, classify x (1) according to the Euclidean distance classifier and (2) according to the Mahalanobis distance classifier. Comment on the results.

Solution. Take the following steps:

Step 1. Use the function *euclidean_classifier* by typing

```
x=[0.1 0.5 0.1]';
m1=[0 0 0]'; m2=[0.5 0.5 0.5]';
m=[m1 m2];
z=euclidean_classifier(m,x)
```

The answer is $z = 1$; that is, the point is classified to the ω_1 class.

Step 2. Use the function *mahalanobis_classifier* by typing

```
x=[0.1 0.5 0.1]';
m1=[0 0 0]'; m2=[0.5 0.5 0.5]';
m=[m1 m2];
S=[0.8 0.01 0.01;0.01 0.2 0.01; 0.01 0.01 0.2];
z=mahalanobis_classifier(m,S,x);
```

This time, the answer is $z = 2$, meaning the point is classified to the second class. For this case, the optimal Bayesian classifier is realized by the Mahalanobis distance classifier. The point is assigned to class ω_2 in spite of the fact that it lies closer to m_1 according to the Euclidean norm. ■

Appendix

This appendix lists the functions (m-files) developed by the authors and used in the examples in this book. Functions used that are part of MATLAB's commercial distribution have been omitted; the reader is referred to the respective MATLAB manuals.

In the following list, functions are ordered alphabetically by chapter. For further function details, including descriptions of input and output arguments, refer to MATLAB's help utility. Also see the complete source code of the listed m-files, provided as part of the software on the companion website.

Chapter 1

bayes_classifier Bayesian classification rule for c classes, modeled by Gaussian distributions (also used in Chapter 2).

comp_gauss_dens_val Computes the value of a Gaussian distribution at a specific point (also used in Chapter 2).

compute_error Computes the error of a classifier based on a data set (also used in Chapter 4).

em_alg_function EM algorithm for estimating the parameters of a mixture of normal distributions, with diagonal covariance matrices.

EM_pdf_est EM estimation of the pdfs of c classes. It is assumed that the pdf of each class is a mixture of Gaussians and that the respective covariance matrices are diagonal.

euclidean_classifier Euclidean classifier for the case of c classes.

Gaussian_ML_estimate Maximum Likelihood parameters estimation of a multivariate Gaussian distribution.

generate_gauss_classes Generates a set of points that stem from c classes, given the corresponding a priori class probabilities and assuming that each class is modeled by a Gaussian distribution (also used in Chapter 2).

k_nn_classifier k-nearest neighbor classifier for c classes (also used in Chapter 4).

knn_density_estimate k-nn-based approximation of a pdf at a given point.

mahalanobis_classifier Mahalanobis classifier for c classes.

mixt_model Generates a set of data vectors that stem from a mixture of normal distributions (also used in Chapter 2).

mixt_value Computes the value of a pdf that is given as a mixture of normal distributions, at a given point.

mixture_Bayes Bayesian classification rule for c classes, whose pdf's are mixtures of normal distributions.

Parzen_gauss_kernel Parzen approximation of a pdf using a Gaussian kernel.

plot_data Plotting utility, capable of visualizing 2-dimensional data sets that consist of, at most, 7 classes.

Auxiliary functions gauss.