

PENGLASIFIKASI BERDASARKAN TEORI KEPUTUSAN BAYES

Ari Fadli, 06827-TE
Jurusan Teknik Elektro FT UGM,
Yogyakarta

1.1. PENDAHULUAN

Dalam bab ini, akan dijelaskan beberapa teknik yang diilhami oleh teori keputusan Bayes, yaitu sebuah teori baru dalam pengenalan pola, perkembangan teori ini dibahas dalam [Theo 09, Chapter 2]. Pembahasan dalam bab ini meliputi teori dasar disertai dengan beberapa latihan soal yang dapat dijadikan sebagai dasar pengetahuan serta ide-ide yang berkenaan dengan klasifikasi.

Dalam sebuah klasifikasi, kita akan diberikan sebuah pola, kemudian berdasarkan pola tersebut, kita dapat mengklasifikasikan sesuatu ke dalam sejumlah kelas. Jumlah kelas disebut sebagai *priori* (probabilitas awal suatu kelas). Setiap pola dinyatakan sebagai himpunan ciri yang bernilai $x(i)$, $i = 1, 2, 3, \dots, l$, yang membentuk vektor ciri berdimensi- l . $x = [x(1), x(2), x(3)]^T \in \mathbb{R}^l$. Diasumsikan bahwa setiap pola bersifat unik yang dinyatakan oleh sebuah vektor ciri tunggal dan hanya dimiliki oleh sebuah *class*.

Diberikan $x \in \mathbb{R}^l$ dan himpunan *class*, ω_i , $i = 1, 2, 3, \dots, c$, dinyatakan dalam teori Bayes, sebagai berikut

$$p(\omega_i | x) p(x) = p(x | \omega_i) p(\omega_i) \dots\dots\dots 1.1$$

Dimana

$$p(x) = \sum_{i=1}^c p(x | \omega_i) p(\omega_i)$$

Ket.

- $p(\omega_i)$: Probabilitas awal *class* ω_i ; $i = 1, 2, \dots, c$.
- $p(\omega_i | x)$: Probabilitas *class* ω_i yang dinyatakan sebagai kelas dari data x
- $p(x)$: *Probability Density Function* (PDF) atau Fungsi Kerapatan Peluang
- $p(x | \omega_i)$: Probabilitas data x termasuk di dalam suatu kelas ω_i

1.2. TEORI KEPUTUSAN BAYES

Sebuah pola dengan label *class* tidak diketahui dan himpunan vektor ciri $x = [x(1), x(2), x(3)]^T \in \mathbb{R}^l$, serta *class* dinyatakan dalam $\omega_1, \dots, \omega_c$

Berdasarkan teori keputusan Bayes, x termasuk ke dalam *class* ω_i , jika

$$p(\omega_i | x) > p(\omega_j | x), \forall j \neq i \dots\dots\dots 1.2$$

Berdasarkan persamaan 1.1, $p(x)$ adalah positif dan sama untuk semua *class*, jika

$$p(x | \omega_i) p(\omega_i) > p(x | \omega_j) p(\omega_j), \forall j \neq i \dots\dots\dots 1.3$$

Catatan

Bayesian classifier optimal dalam meminimalkan kesalahan probabilitas [Theo 09, Chapter 2]

1.3. GAUSSIAN PROBABILITY DENSITY FUNCTION (PDF)

Gaussian PDF [Theo 09, Section 2.4.1] telah secara luas telah digunakan dalam pengenalan pola karena kemam Terakhir PDF adalah jumlah variable statistik random yang independent yang cenderung memiliki nilai tak terbatas.

Bentuk multidimensi gaussian

$$p(x) = \frac{1}{\pi^{n/2} |S|^{1/2}} \exp\left(-\frac{1}{2} (x - m)^T S^{-1} (x - m)\right) \dots\dots\dots (1.4)$$

Dimana

$m = E(x)$ adalah vektor rata-rata (mean vector)

S = Covariance Matrix yang didefinisikan sebagai $S = E[(x - m)(x - m)^T]$

$|S|$ = Determinan dari S

Gaussian PDF sebagai fungsi normal PDF dinotasikan dengan $\mathcal{N}(m, S)$, dalam ruang berdimensi 1, sehingga persamaan 1.4 menjadi

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - m)^2}{2\sigma^2}\right) \dots\dots\dots (1.5)$$

Dimana σ adalah matrik kovarians (covariance matrix)

Contoh 1.3.1

Hitung nilai PDF gaussian, $\mathcal{N}(m, s)$ pada $x_1 = [0.2, 1.3]$ dan $x_2 = [2.2, -1.3]$, dimana $m = [0, 1]^T, S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Solusi

Gunakan fungsi comp_gauss_dens_val untuk menghitung PDF Gaussian :

```
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);
```

memberikan hasil nilai probabilitas pg1 dan pg2 0.1491 dan 0.001,

Contoh 1.3.2.

Klasifikasi vektor data dalam ruang dimensi 2, dengan vektor data pada class ω_1 dan ω_2 diperoleh dari distribusi Gaussian $\mathcal{N}(m_1, S_1)$ dan $\mathcal{N}(m_2, S_2)$ dimana

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

Dengan asumsi bahwa $P(\omega_1) = P(\omega_2) = 1/2$, maka tentukan apakah $x = [1.8, 1.8]^T$ masuk kedalam class ω_1 atau ω_2

Solusi.

Gunakan built-in MATLAB `comp_gauss_dens_val` untuk menghitung Gaussian PDF

```
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);
```

Memberikan hasil nilai probabilitas p_1 dan p_2 sama dengan 0.042 and 0.0189, Sehingga x di klasifikasikan kedalam ω_1 berdasarkan *Bayesian classifier*.

Latihan 1.3.1

Ulangi contoh 1.3.2 untuk nilai $P(\omega_1) = 1/6$ dan $P(\omega_2) = 5/6$, dan $P(\omega_1) = 5/6$ dan $P(\omega_2) = 1/6$. Amati ketergantungan klasifikasi berdasarkan *priori probabilities* [Theo 09, Section 2.4.2]

Contoh 1.3.3.

Dihasilkan sejumlah data ($N=500$) dalam ruang dimensi 2 dari sebuah distribusi gaussian $\mathcal{N}(m, S)$ dengan $m=[0,0]^T$, $S = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$ yang memiliki nilai

$\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0$
 $\sigma_1^2 = \sigma_2^2 = 0.2, \sigma_{12} = 0$
 $\sigma_1^2 = \sigma_2^2 = 2, \sigma_{12} = 0$
 $\sigma_1^2 = 0.2, \sigma_2^2 = 2, \sigma_{12} = 0$
 $\sigma_1^2 = 2, \sigma_2^2 = 0.2, \sigma_{12} = 0$
 $\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0$
 $\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = 0$
 $\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = 0$

Plot setiap himpunan data tersebut seperti sebuah cluster

Solusi

Gunakan fungsi built-in MATLAB `mvnrnd` untuk menghasilkan himpunan data

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

Dimana X adalah matriks dalam bentuk vektor kolom yang berisi data, Fungsi `mvnrnd` merupakan fungsi yang akan menghasilkan bilangan secara random berdasarkan pada distribusi Gaussian, dengan *zero mean* (matriks rerata = 0) dan *unit variance*

Plot Data menggunakan perintah

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

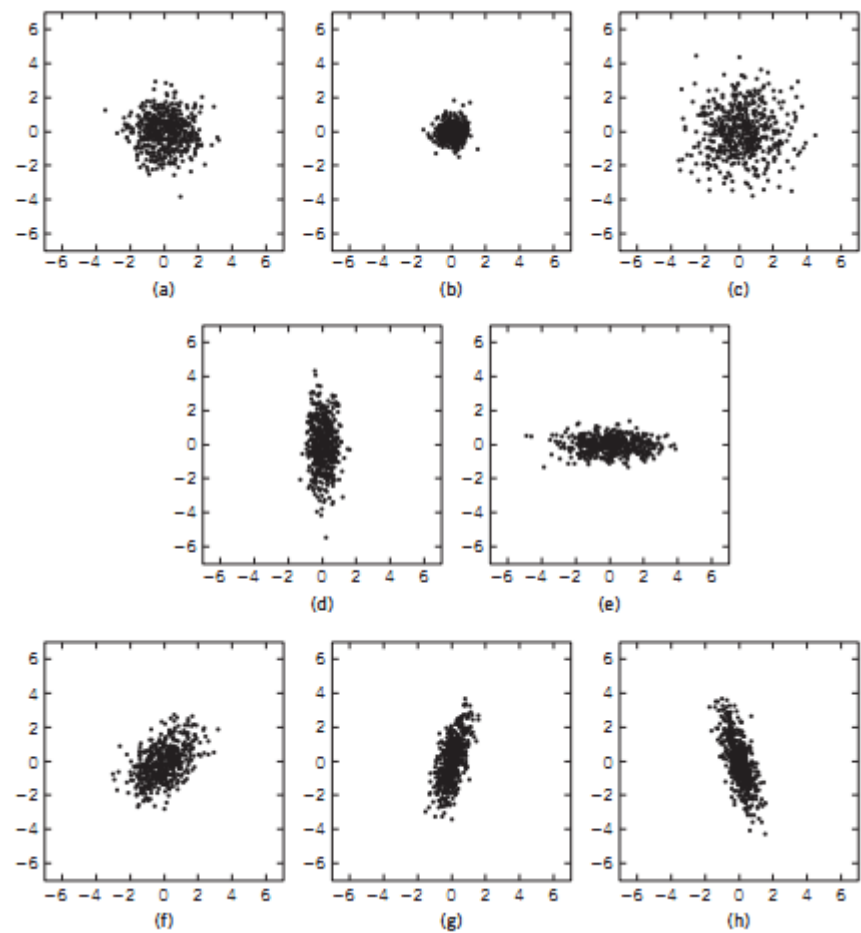
Demikian juga untuk data ke-2

```
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])
```

Berdasarkan gambar 1.1 dapat diamati

- 1. Ketika dua koordinat x adalah *uncorrelated* ($\sigma_{12}=0$) dengan nilai variances sama, maka vektor data membentuk cluster “*spherically shaped*” (gambar 1.1 (a–c)).
- 2. Ketika dua koordinat x adalah *uncorrelated* ($\sigma_{12}=0$) dengan nilai varians tidak sama, maka vektor data membentuk cluster “*ellipsoidally shaped*” (gambar 1.1 (d–e)).
- 3. Ketika dua koordinat x adalah *correlated* ($\sigma_{12}\neq 0$) sumbu mayor dan minor cluster berbentuk *ellipsoidally* tidak lagi sejajar. Derajat perputaran ditentukan oleh nilai sehubungan dengan sumbu tergantung pada nilai σ_{12}

Plot dari himpunan data yang diperoleh, tampak seperti dalam gambar 1.1 berikut



gambar 1.1 Plot 8 buah himpunan data

1.4. PENGKLASIFIKASI BERDASAR JARAK TERPENDEK (MINIMUM DISTANCE CLASSIFIER)

1.4.1. Pengklasifikasi Jarak Euclidean (*The Euclidean Distance Classifier*)

Optimal Bayessian Classifier secara sederhana dapat di asumsikan

- 1. Semua *class* memiliki nilai probabilitas sama
- 2. Data dalam setiap *class* berdasar pada distribusi Gaussian
- 3. Nilai *covariance matrix* sama untuk setiap *class*
- 4. *covariance matrix* adalah diagonal dan semua elementnya sama sehingaa $S = \sigma^2 I$, dimana I adalah matriks identitas

Berdasarkan beberapa asumsi tersebut *Optimal Bayessian Classifier* disebut sebagai *minimum eucludiance distance classifiers*, yang akan menentukan nilaix dalam sebuah himpunan data kedalam sebuah class ω_1 jika

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

1.4.2. The Mahalanobis D istance Classifier

Optimal Bayessian Classifier disebut sebagai *minimum Mahalobis distance classifier* jika asumsi diatas menjadi :

- 1. Semua *class* memiliki nilai probabilitas sama
- 2. Data dalam setiap *class* berdasar pada distribusi Gaussian
- 3. Nilai *covariance matrix* sama untuk setiap *class*

Akan menentukan nilai x dalam sebuah himpunan data kedalam sebuah class ω_1 jika

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

Dimana S sama dengan *covariance matrix* dalam bentuk Gaussian
[[Theo 09, Section 2.4.2](#)]

Contoh 1.4.1.

Klasifikasi dua class (ω_1 dan ω_2) dalam ruang berdimensi 3, yang dimodelkan oleh distribusi Gaussian dengan *mean* $m_1 = [0, 0, 0]^T$ dan $m_2 = [0.5, 0.5, 0.5]^T$ dengan *covariance matrix* untuk kedua distribusi tersebut

$$S_1 = \begin{bmatrix} 0.8 & 0.01 & 0.01 \\ 0.01 & 0.2 & 0.01 \\ 0.01 & 0.01 & 0.2 \end{bmatrix}$$

Jika diberikan data $x[0.1, 0.5, 0.1]^T$ klasifikasikan $x(1)$ menggunakan *minimum eucludiance distance classifiers* dan *minimum Mahalobis distance classifier*

Langkah 1. Gunakan built-in MATLAB `euclidean_classifier`

```
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)
```

sehingga diperoleh hasil $z = 1$, dan data di klasifikasikan ke dalam $class \omega_1$.

Langkah 2. Gunakan built-in MATLAB `mahalanobis_classifier`

```
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; 0.0 1 0.2 0.01; 0.01 0.01 0.2];
z=mahalanobis_classifier(m,S,x);
```

sehingga diperoleh hasil $z = 2$, dan data di klasifikasikan ke dalam $class \omega_2$.

1.4.3. Maximum Likelihood Parameter Estimation of Gaussian PDFs

Maximum Likelihood (ML) merupakan teknik yang digunakan untuk memperkirakan nilai parameter dari PDF *Gaussian* yang tidak diketahui, dengan asumsi terdapat sejumlah data N , $x_i \in \mathcal{R}^1$, maka ML dapat memperkirakan nilai *mean* dan *covariance matrix*

$$m_{ML} = \frac{1}{N} \sum_{i=1}^N x_i \text{ dan } S_{ML} = \frac{1}{N} \sum_{i=1}^N (x_i - m_{ML})(x_i - m_{ML})^T$$

Teknik ini memiliki nilai perkiraan *unbiased*, jika *covariance matrix* dibagi dengan $(N-1)$, [Theo09, Section 2.5.1].

Contoh Soal 1.4.2.

Dihasilkan sejumlah data ($N = 50$) dalam vektor ruang 2 dimensi yang diperoleh dari distribusi Gaussian $\mathcal{N}(m, S)$, dengan *mean* dan *covariance matrix*

$$m = [2, -2]^T \text{ dan } S = \begin{bmatrix} 0.9 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

Hitung nilai estimasi ML yang berupa nilai *mean* (m), dan *covariance matrix* (S) $\mathcal{N}(m, S)$ serta beri komentar pada hasil yang diperoleh

Solusi

Gunakan fungsi built-in MATLAB `rnd` untuk menghasilkan himpunan data X

```
randn('seed',0)
m = [2 -2];
S = [0.9 0.2; 0.2 .3];
X = mvnrnd(m,S,50)';
```

Gunakan fungsi built-in MATLAB `Gaussian_ML_estimate` Untuk memperkirakan nilai m dan S

```
[m_hat,S_hat]=Gaussian_ML_estimate(X);
```

Hasil yang diperoleh adalah

$$m_hat=[2.0495,-1.9418]^T, S_hat=\begin{bmatrix} 0.8082 & 0.0885 \\ 0.0885 & 0.2298 \end{bmatrix}$$

Dengan melihat hasil tersebut, dapat disimpulkan bahwa penggunaan data random dengan jumlah 50 data masih kurang tepat dalam memperkirakan nilai distribusi

Latihan 1.4.1

Ulangi *contoh soal 1.4.2* untuk nilai $N = 500$ points dan $N = 5000$ points. Berikan komentar pada hasil yang diperoleh

Contoh Soal 1.4.3.

Dihasilkan dua himpunan data yaitu, X (data pelatihan) and $X1$ (data pengujian), yang terdiri dari $N = 1000$, dalam vektor ruang 3dimensi, dengantiga buahclass yaitu ω_1 , ω_2 , dan ω_3 setiap class dimodelkan menggunakan distribusi gaussian, $m_1=[0,0,0]^T$, $m_2=[1,2,2]^T$, $m_3=[3,3,4]^T$ dan *covariance matrix*

$$S_1=S_2=S_3=\begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}=\sigma^2 I$$

- Gunakan X , untuk memperkirakan estimasi menggunakan teknik ML untuk nilai mean dan *covariance matrix* dari distribusi ketiga buah class.
- Dengan menggunakan *minimum Euclidean distance classifier* klasifikasikan titik $X1$ berdasarkan nilai ML yang diperoleh pada langkah sebelumnya.
- Dengan menggunakan *minimum Mahalanobis distance classifier* klasifikasikan titik $X1$ berdasarkan nilai ML yang diperoleh pada langkah sebelumnya.
- Dengan menggunakan *Bayesian classifier* klasifikasikan titik $X1$ berdasarkan nilai ML yang diperoleh pada langkah sebelumnya.
- Untuk setiap kasus tersebut hitung nilai kesalahan probabilitas (*error probability*) dan bandingkan hasilnya (mengapa bisa terjadi seperti itu).

Solusi

Gunakan fungsi built-in MATLAB `generate_gauss_classes` menghasilkan himpunan data X

```
m=[0 0 0;122;33 4]';
S1=0.8*eye(3);
S(:, :, 1)= S1;
S(:, :, 2)= S1;
S(:, :, 3)=S1;
P=[1/3 1/3 1/3]';
N=1000;
randn('seed',0)
[X,y]=generate_gauss_classes(m,S,P,N);
```

dimana

$X = 3 \times N$ matrix yang berisi data dalam bentuk vektor kolom

$y = N$ – dimensi vektor yang berisi label class yang berhubungan dengan vektor data

P = Vektor yang menunjukkan Probabilitas masing-masing class

Gunakan fungsi built-in MATLAB `generate_gauss_classes` menghasilkan himpunan data $X1$

```
randn('seed',100);
[X1,y1]=generate_gauss_classes(m,S,P,N);
```

Dimana fungsi randn di inialisasi menggunakan seed = 100.

Langkah 1. Hitung nilai perkiraan ML yaitu nilai mean dan covariance matrix, dengan menggunakan fungsi built-in MATLAB *Gaussian_ML_estimate*

```
class1_data =X(:,find(y==1));
[m1_hat, S1_hat]=Gaussian_ML_estimate(class1_data);
class2_data = X(:,find(y==2));
[m2_hat, S2_hat]=Gaussian_ML_estimate(class2_data);
class3_data =X(:,find(y==3));
[m3_hat,S3_hat]=Gaussian_ML_estimate(class3_data);
S_hat=(1/3)*(S1_hat+S2_hat+S3_hat);
m_hat=[m1_hat m2_hat m3_hat];
```

Langkah 2. Hitung nilai perkiraan ML yaitu nilai mean dan covariance matrix, dengan menggunakan minimum eucludiance distance classifiers menggunakan fungsi built-in MATLAB *euclidean_classifier*

```
z_euclidean =euclidean_classifier(m_hat,X1);
```

dimana z_euclidean adalah vektor berdimensi N

Langkah 3. Hitung nilai perkiraan ML yaitu nilai mean dan covariance matrix, dengan menggunakan minimum *Mahalanobis distance classifier* menggunakan fungsi built-in MATLAB *mahalanobis_classifier*

```
z_mahalanobis= mahalanobis_classifier(m_hat,S_hat,X1);
```

Langkah 4. Hitung nilai perkiraan ML yaitu nilai mean dan covariance matrix, dengan menggunakan minimum *Bayessian classifier* menggunakan fungsi built-in MATLAB *bayes_classifier*

```
z_bayesian=bayes_classifier(m,S,P,X1);

err_euclidean = (1-length(find(y1==z_euclidean))/length(y1));
err_mahalanobis = (1-length(find(y1==z_mahalanobis))/length(y1));
err_bayesian = (1-length(find(y1==z_bayesian))/length(y1));
```

error probality masing-masing pengklasifikasi yaitu Euclidean, Mahal anobis, and Bayesian classifiers adalah 7.61%, 7.71%, and 7.61%.

Latihan 1. 4. 2

Ulangi **contoh soal 1.4.3** menggunakan

$$S_1=S_2 = S_3=\begin{bmatrix}0.8 & 0.2 & 0.1\\0.2 & 0.8 & 0.2\\0.1 & 0.2 & 0.8\end{bmatrix}\neq\sigma^2I$$

Komentari hasilnya

Latihan 1. 4. 3

Ulangi **contoh soal 1.4.3** dengan menggunakan $P_1=1/2,P_2=P_3=1/4$ untuk menghasilkan X dan X1. Untuk kasus dengan *priori probabilities* tidak sama, apakah *Bayesian Classifier*, tetap memberikan performa yang baik, mengapa

Latihan 1. 4. 4

Ulangi **contoh soal 1.4.3** dengan menggunakan $P_1=P_2=P_3= 1/3$ dan

$$S_1=\begin{bmatrix}0.8 & 0.2 & 0.1\\0.2 & 0.8 & 0.2\\0.1 & 0.2 & 0.8\end{bmatrix}, S_2=\begin{bmatrix}0.6 & 0.01 & 0.01\\0.01 & 0.8 & 0.01\\0.01 & 0.01 & 0.6\end{bmatrix}, S_3=\begin{bmatrix}0.6 & 0.1 & 0.1\\0.1 & 0.6 & 0.1\\0.1 & 0.1 & 0.6\end{bmatrix}$$

Cobakan nilai *mean* dan *priori probabilities*

1.5. MIXTURE MODEL

Mixture Model merupakan sebuah metode untuk memodelkan PDF yang tidak diketahui [**Theo 09, Section 2.5.5**]

$$p(x)=\sum_{j=1}^J P_j p(x | j) \dots\dots\dots (1.6)$$

Dimana

$$\sum_{j=1}^J P_j =1, \int p(x | j) dx=1$$

Untuk nilai J yang cukup besarp (x|j)yang digunakan adalah Gaussi ans, $\mathcal{N}(m_j,S_j)$, $j = 1, 2, ..., J$.Berdasarkan persamaan 1.6, dimana setiap data di hasilkan dari penjumlahan nilai PDF yang memiliki probabilities

Contoh Soal 1.5.1

Diketahui PDF dengan dimensi 2

$$p(x)=P_1 p(x | 1)+P_2 p(x | 2) \dots\dots\dots (1.7)$$

Dimana $p(x | j), j=1, 2,$ merupakan distribusi normal dengan $m_1=[1,1]^T$ $m_2=[3,3]^T$

$$S_1=\begin{bmatrix}\sigma_1^2 & \sigma_{12}^2\\ \sigma_1^2 & \sigma_2^2\end{bmatrix} S_2=\begin{bmatrix}\sigma^2 & 0\\ 0 & \sigma^2\end{bmatrix}$$

Dimana $\sigma = 0.1, \sigma = 0.2 \sigma = 0.18 \sigma = 0.1$

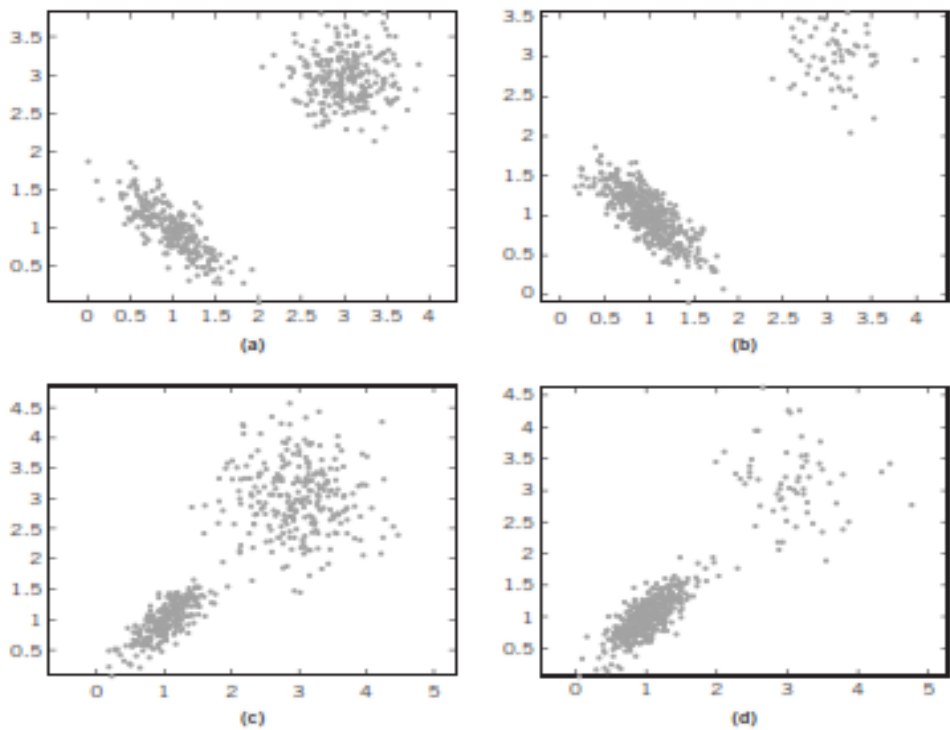
Solusi .

Gunakan fungsi built-in MATLAB *mixt_model* untuk menghasilkan data X

```
randn('seed',0); %initialization of MATLAB's randn generator
m1=[1, 1]';
m2=[3, 3]';
m=[m1 m2];
S(:, :,1)= [0 .1 -0.08; -0.08 0.2];
S(:, :,2)= [0 .1 0; 0 0.1];
P=[1/2 1/2];
N=500;
sed=0; % used for the initialization of MATLAB's rand generator
[X,y]=mixt_model(m,S,P,N,sed);
plot(X(1,:),X(2,:),'.');
```

dimana

- sed digunakan untuk inisialisasi fungsi built-in MATLAB, fungsi *rand*, akan membangkitkan bilangan dari sebuah distribusi dengan interval [0, 1]
- y merupakan vektor dengan element yang berisi *label* distribusi i data dalam vektor that generated the i th data vector



gambar 1.2 case I (a), case II (b), case III (c-d)

1.6. THE EXPECTATION MAXIMUM ALGORITHM

Algoritma *Expectation Maximum* (EM) secara iteratif menghitung nilai perkiraan dalam distribusi gaussian berdasarkan pada nilai awal yang diberikan. [Theo 09, Section 2.5.5].

$$p(x|j) = \frac{1}{(2\pi)^{1/2} \sigma_j} \exp\left(-\frac{(x - m_j)^T (x - m_j)}{2\sigma_j^2}\right), \quad j=1, 2, \dots, J$$

Dimana

Nilai *mean* (m) = m_j dengan $j = 1, 2, \dots, J$

$IJ = IJ$ Parameter total

$J = J$ Parameter Total

Contoh Soal 1.6.1.

Carilah Himpunan X dengan $N = 500$ ruang dimensi 2 dari sebuah distribusi PDF

$$p(x) = \sum_{j=1}^3 P_j p(x|j)$$

Dimana memperlihatkan hubungan ketergantungan, antara masukan dengan *probabilities*

$p(x|j)$ $j=1, 2, 3$ distribusi normal dalam ruang dimensi 2 dengan nilai mean

$m_1 = [1, 1]^T$ $m_2 = [3, 3]^T$ $m_3 = [2, 6]^T$ dengan matriks covariance

$S_1 = 0.1I$ $S_2 = 0.2I$ $S_3 = 0.3I$ dengan I adalah matriks identitas 2×2

$P_1 = 0.4$ $P_2 = 0.4$ $P_3 = 0.2$

$J=3, m_1, ini = [0, 2]^T, m_2, ini = [5, 2]^T, m_3, ini = [5, 5]^T, S_1, ini = 0.15I, S_2, ini = 0.27I$

$S_3, ini = 0.4I, P_1, ini = P_2, ini = P_3, ini = 1/3$

$J=3, m_1, ini = [1.6, 1.4]^T, m_2, ini = [1.4, 1.6]^T, m_3, ini = [1.3, 1.5]^T, S_1, ini = 0.2I, S_2, ini = 0.4I$

$S_3, ini = 0.3, P_1, ini = P_2, ini = P_3, ini = 0.4$

$J=2, m_1, ini = [1.6, 1.4]^T, m_2, ini = [1.4, 1.6]^T, S_1, ini = 0.2I, S_2, ini = 0.4I$

$P_1, ini = P_2, ini = 1/2$

Beri komentar pada hasil yang diperoleh

Solusi .

Gunakan fungsi built-in MATLAB untuk menghasilkan data X kemudian plotnya

```
randn('seed',0) ;
m1=[1,1]'; m2=[3,3]'; m3=[2,6]';
m=[m1 m2 m3];
S(:, :, 1)=0.1*eye(2) ;
S(:, :, 2)=0.2*eye(2) ;
S(:, :, 3)=0.3*eye(2) ;
P=[0.4 0.4 0.2];
N=500;
sed=0;
[X,y]=mixture_model(m,S,P,N,sed) ;
plot_data(X,y,m,1)
```

Kemudian

Langkah 1. Gunakan fungsi built-in MATLAB `em_alg_function` untuk memperkirakan parameter *mixture model*

```
m1_ini=[0;2];
m2_ini=[5;2];
m3_ini=[5;5];
m_ini=[m1_ini m2_ini m3_ini];
s_ini=[.1 5 .27 .4];
Pa_ini=[1/3 1/3 1/3];
e_min=10^(-5);
[m_hat,s_hat ,Pa, iter, Q_tot ,e_tot] =. ..
em_alg_function(X,m_ini,s_ini,Pa_ini,e_min);
```

Dimana

- M_hat adalah matriks lxJ dengan jth kolom untuk memperkirakan nilai *mean* dari jth distribusi
- S adalah vektor dengan J dimensional dimana jth elemen adalah variance untuk (jth distribusi) yang di asumsikan sebagai *covariance matrix* dalam bentuk $s(j)*I$ dimana I adalah matriks identitas
- Pa = vektor dengan dimensi J dengan jth elemen yang digunakan untuk mengestimasi *priori probability* dari distribusi jth .

Hasil estimasinya

$$\hat{m}_1 = [1.02 \ 0.98]^T, \hat{m}_2 = [2.94 \ 3.02]^T, \hat{m}_3 = [2.03 \ 6.00]^T$$

$$\hat{S}_1 = 0.10I, \hat{S}_2 = 0.22I, \hat{S}_3 = 0.30I$$

$$\hat{P}_1 = 0.39, \hat{P}_2 = 0.43, \hat{P}_3 = 0.18$$

Algoritma ini convergen setelah iterasi ke-12 (gambar 1.3a)

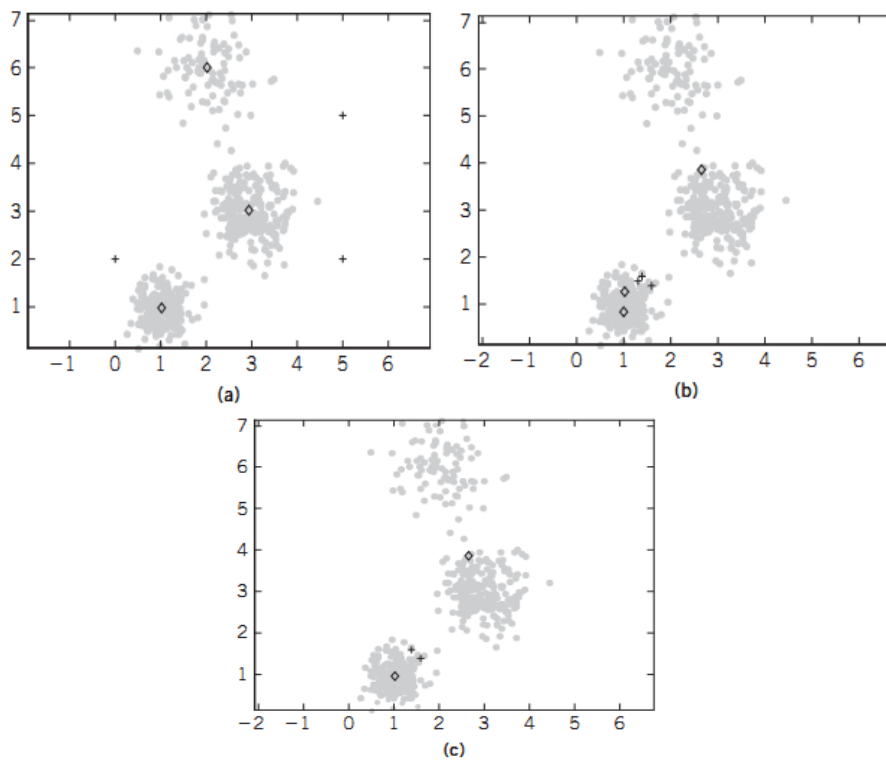
Langkah 2. Ulangi langkah 1 untuk mendapatkan

$$\hat{m}_1 = [1.01 \ 0.84]^T, \hat{m}_2 = [2.66 \ 3.86]^T, \hat{m}_3 = [1.01 \ 1.26]^T$$

$$\hat{S}_1 = 0.09I, \hat{S}_2 = 1.28I, \hat{S}_3 = 0.07I$$

$$\hat{P}_1 = 0.26, \hat{P}_2 = 0.62, \hat{P}_3 = 0.12$$

Algoritma ini convergen setelah iterasi ke-533 (gambar 1.3b)



gambar 1.3 kondisi awal (+) dan kondisi akhir (◇)

Contoh 1.6.1

Berdasarkan gambar 1.3 Diketahui kondisi awal (+) dan kondisi akhir (◇) perkiraan nilai *mean* untuk distribusi normal, untuk 3 kondisi ini.

Langkah 3. Ulangi langkah 1 untuk mendapatkan

$$\hat{m}_1 = [1.01, 0.97]^T, \hat{m}_2 = [2.66, 3.86]^T$$

$$\hat{S}_1 = 0.10I, \hat{S}_2 = 1.27I$$

$$\hat{P}_1 = 0.38, \hat{P}_2 = 0.62$$

Algoritma ini convergen setelah iterasi ke-10

Nilai estimasi yang dihasilkan dari algoritma ini bergantung pada nilai inisialisasi yang diberikan sebagai nilai awal. seperti dalam [Theo 09, Chapter 16] metode ini digunakan untuk mengidentifikasi kerapatan (cluster) yang dibentuk oleh himpunan vektor data.

Contoh 1.6.2

Pada contoh soal ini akan diberikan penerapan algoritma EM untuk klasifikasi. Himpunan data X terdiri dari $N=1000$ yang merupakan, 500 data merupakan class ω_1 dengan distribusi normal

$$p_1(x) = \sum_{j=1}^3 P_{1j} p_1(x|j), \text{ dimana } p(x|j), j=1,2,3$$

Dan nilai mean $m_{11} = [1.25, 1.25]^T$, $m_{12} = [2.75, 2.75]^T$, $m_{13} = [2, 6]^T$ dan

Covariance matrix $S_{1j} = \sigma_1^2 I$ dengan $j=1,2,3$
 Dimana
 $\sigma_{11}^2 = 0.1, \sigma_{12}^2 = 0.2, \sigma_{13}^2 = 0.3$ dan *mixing probabilities* $P_{11} = 0.4, P_{12} = 0.4$, dan $P_{13} = 0.2$

Sedangkan 500 data yang lain, merupakan class ω_2 dengan distribusi normal

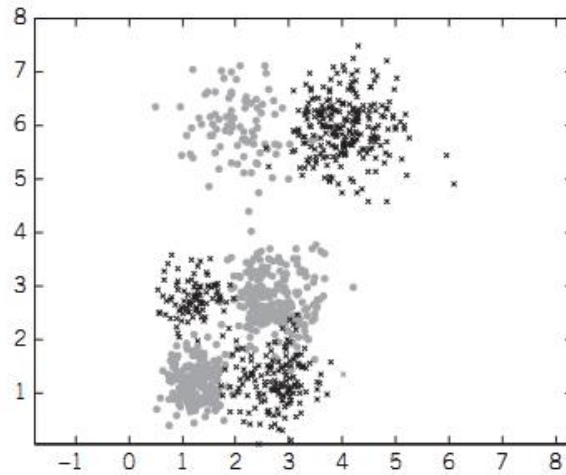
$$p_2(x) = \sum_{j=1}^3 P_{2j} p_2(x|j), \text{ dimana } p(x|j), j=1,2,3$$

Dan nilai mean $m_{21}=[1.25, 1.25]^T, m_{22}=[2.75, 1.25]^T, m_{23}=[4, 6]^T$ dan

Covariance matrix $S_{2j} = \sigma_2^2 I$ dengan $j=1,2,3$

Dimana
 $\sigma_{21}^2 = 0.1, \sigma_{22}^2 = 0.2, \sigma_{23}^2 = 0.3$ dan *mixing probabilities* $P_{11} = 0.4, P_{12} = 0.4$, dan $P_{13} = 0.2$

Dengan menggunakan data X sebagai data pelatihan, gunakan algoritma EM untuk memperkirakan nilai parameter dalam model PDF



gambar 1.4 Himpunan data X

Dengan menggunakan algoritma EM untuk memperkirakan nilai $p_1(x)$ dan $p_2(x)$, berdasarkan himpunan data X yang digunakan sebagai parameter inisialisasi :

- Dan nilai mean $m_{11}=[1.25, 1.25]^T, m_{12}=[2.75, 2.75]^T, m_{13}=[2, 6]^T$ dan

Covariance matrix $S_{1j} = \sigma_1^2 I$ dengan $j=1,2,3$ Dimana $\sigma_{11}^2 = 0.1, \sigma_{12}^2 = 0.2, \sigma_{13}^2 = 0.3$ dan *mixing probabilities* $P_{11} = 0.4, P_{12} = 0.4$, dan $P_{13} = 0.2$

- Dan nilai mean $m_{21}=[1.25, 1.25]^T, m_{22}=[2.75, 1.25]^T, m_{23}=[4, 6]^T$ dan

Covariance matrix $S_{2j} = \sigma_2^2 I$ dengan $j=1,2,3$, Dimana $\sigma_{21}^2 = 0.1, \sigma_{22}^2 = 0.2, \sigma_{23}^2 = 0.3$ dan *mixing probabilities* $P_{11} = 0.4, P_{12} = 0.4$, dan $P_{13} = 0.2$

Solution

Langkah 1, Gunakan fungsi built-in MATLAB untuk menghasilkan himpunan bagian X_1 dari X

```
m11=[1.25 1.25]'; m12=[2.75 2.75]';m1 3= [2 6]';
m1=[m11 m12 m13];
S1(:, :,1)=0.1*eye(2);
S1(:, :,2)=0.2*eye(2);
S1(:, :,3)=0.3*eye(2);
P1=[0.4 0.4 0.2];
N1=500;
sed=0;
[X1,y1]=mixt_model(m1,S1,P1,N1,sed) ;
```

Hal yang sama dilakukan untuk mencari himpunan bagian X_2

nilai Z dihitung dalam dua tahap:

1. Sejumlah (500) titik dibangkitkan dari dari *class* ω_1

```
mZ11=[1.2 5 1.25]'; mZ12=[2.7 5 2.75]';mZ 13= [2 6]';
mZ1=[mZ11 mZ12 mZ13];
SZ1(:, :,1)=0.1*eye(2) ;
SZ1(:, :,2)=0.2*eye(2) ;
SZ1(:, :,3)=0.3*eye(2) ;
wZ1=[0.4 0.4 0.2];
NZ1=500;
sed=100;
[Z1,yz1]=mixt_model(mZ1,SZ1,wZ1,NZ1,sed) ;
```

Sedangkan untuk Sejumlah (500) lainnya, titik dibangkitkan dari dari *class* ω_2 (dengan $\text{sed} = 100$)

Langkah 1 Untuk mengestimasi nilai parameternya digunakan

```
m11_ini=[0; 2]; m12_ini=[5 ; 2]; m13_ini= [5; 5];
m1_ini=[m11_ in i m12_ini m13_ini];
S1_ini=[0.15 0.27 0.4];
w1_ini=[1 /3 1/3 1/3];
m21_ini=[5; 2]; m22_ini=[3 ; 4]; m23_ini= [2 ; 5];
m2_ini=[m21_ in i m22_ini m23_ini];
S2_ini=[0.15 0.27 0.35];
w2_ini=[1 /3 1/3 1/3];
m_ini{1}=m1_ini;
m_ini{2}=m2_ini;
S_ini{1}=S1_ini;
S_ini{2}=S2_ini;
w_ini{1}=w1_ini;
w_ini{2}= w2_ini;
[m_hat,S_hat,w_hat,P_ hat]= ...
EM_pdf_est([X1 X2],[ones(1,500) 2*ones(1 ,5 00 )], m_
ini,S_ini,w_ini) ;
```

Langkah 2. Gunakan fungsi `_ Bayes` untuk mengklasifikasikan vectors Z and function `compute_error` untuk mendefinisikan *classification error*.

```
for j=1:2
le=length(S_ hat{j}) ;
te=[];
for i=1:le
```

```

te(:, :, ) = S_hat {j} (i) * eye( 2);
end
S{j}=te;
end

```

[y_est]= mixture _Bayes(m_ hat ,S ,w _ha t, P_ hat ,Z);
[classification_error]=com ut e_ err or ([ones(1,500) 2*ones(1,500)], y_ est);
classification error i s equal t o 4.20%.

Contoh 1.6.1

Ulangi Contoh 1.6.2 using X1, X2, Z1, Z2 dengan parameter inisial

- Untuk $p1(\omega)$ menggunakan distribusi normal, $m_{11}=[5,5]^T$,.ini $m_{12}=[5.5, 5.5]^T$ dan $m_{13}=[5,5]^T$ 3 dan *mixing probabilities* $P_{11} = 0.2$, $P_{12} = 0.4$, dan $P_{13} = 0.4$, Dimana $\sigma_{11}^2 = 0.2$, $\sigma_{12}^2 = 0.4$, $\sigma_{13}^2 = 0.3$
Untuk $p2(\omega)$ menggunakan distribusi normal, $m_{21}=[2 \ 2]^T$,.ini $m_{22}=[1.98, 1.98]$ dan $m_{23}=[2.4, 2.4]^T$ dan *mixing probabilities* $P_{21} = 0.8$, $P_{22} = 0.1$, dan $P_{23} = 0.1$, Dimana $\sigma_{21}^2 = 0.06$, $\sigma_{22}^2 = 0.05$, $\sigma_{23}^2 = 0.4$
- Untuk $p1(\omega)$ menggunakan distribusi normal, $m_{11}=[1.6,1.4]^T$,.ini $m_{12}=[1.4,1.6]^T$ dan $m_{13}=[1.3,1.5]^T$ 3 dan *mixing probabilities* $P_{11} = 0.2$, $P_{12} = 0.4$, dan $P_{13} = 0.4$, Dimana $\sigma_{11}^2 = 0.2$, $\sigma_{12}^2 = 0.4$, $\sigma_{13}^2 = 0.4$
Untuk $p2(\omega)$ menggunakan distribusi normal, $m_{21}=[1.5 \ 1.7]^T$,.ini $m_{22}=[1.7, 1.5]$ dan $m_{23}=[1.6,1.6]^T$ dan *mixing probabilities* $P_{21} = 0.1$, $P_{22} = 0.8$, dan $P_{23} = 0.1$, Dimana $\sigma_{21}^2 = 0.06$, $\sigma_{22}^2 = 0.05$, $\sigma_{23}^2 = 0.02$
- Untuk $p1(\omega)$ menggunakan distribusi normal, $m_{11}=[0,2]^T$,.ini $m_{12}=[5, 2]^T$ dan $m_{13}=[5,5]^T$ $m_{14}=[3,4]^T$ dan *mixing probabilities* $P_{11} = P_{12}$, $P_{13}= P_{13}= 0.25$, Dimana $\sigma_{11}^2 = 0.15$, $\sigma_{12}^2 = 0.27$, $\sigma_{13}^2 = 0.4$, $\sigma_{14}^2 = 0.2$
Untuk $p2(\omega)$ menggunakan distribusi normal, $m_{21}=[1 \ 2]^T$,.ini $m_{22}=[3.2, 1.5]$ dan $m_{23}=[1,4]^T$ $m_{24}=[4, 2]^T$ dan *mixing probabilities* $P_{11} = P_{12}$, $P_{13}= P_{13}= 0.25$, Dimana $\sigma_{21}^2 = 0.15$, $\sigma_{22}^2 = 0.08$, $\sigma_{23}^2 = 0.27$, $\sigma_{24}^2 = 0.05$
- Untuk $p1(\omega)$ menggunakan distribusi normal, $m_{11}=[0,2]^T$,.ini $m_{12}=[5, 2]^T$ dan *mixing probabilities* $P_{11} = P_{12} = 0.5$, Dimana $\sigma_{11}^2 = 0.15$, $\sigma_{12}^2 = 0.27$
Untuk $p2(\omega)$ menggunakan distribusi normal, $m_{21}=[1 \ 2]^T$ dan *mixing probabilities* $P_{21}1$, Dimana $\sigma_{21}^2 = 0.15$
- Untuk $p1(\omega)$ menggunakan distribusi normal, $m_{11}=[2,2]^T$ dan *mixing probabilities* $P_{11} = 1$, Dimana $\sigma_{11}^2 = 0.4$
Untuk $p2(\omega)$ menggunakan distribusi normal, $m_{21}=[1 \ 2]^T$ dan *mixing probabilities* $P_{21}1$, Dimana $\sigma_{21}^2 = 0.15$

Untuk setiap skenario, ini tentukan prosentase kesalahan algoritma EM

1.7. PARZEN WINDOWS

Merupakan sebuah metode nonparametrik untuk menentukan nilai PDF sebuah himpunan N , $x_i \in \mathbb{R}^1, i = 1, 2, \dots, N$ maka nilai PDF nya dapat di estimasi menggunakan persamaan

$$p(x) = \frac{1}{Nh^1} \sum_{i=1}^N \phi\left(\frac{x - x_i}{h}\right) \dots\dots\dots (1.8)$$

Untuk nilai Nilai N yang besar dan nilai h yang relatif kecil dan ϕ dedefinisikan fungsi kernel,

$$p(x) \approx \frac{1}{N} \sum_{i=1}^N \frac{1}{(2\pi)^{1/2h^1}} \exp\left(-\frac{(x - x_i)^T (x - x_i)}{2h^2}\right) \dots\dots\dots (1.9)$$

Contoh Soal 1.7

Dihasilkan $N = 1000$ data points yang berada pada sumbu real $x_i \in \mathbb{R}^1, i = 1, 2, \dots, 3$ dari PDF dan plot $p(x)$

$$p(x) = \frac{1}{3} \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) + \frac{2}{3} \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x - 2)^2}{2\sigma_1^2}\right)$$

Dimana
 $\sigma_1^2 = \sigma_2^2 = 0.2$ dan $h = 0.1$

Solution .
Gunakan fungsi built-in MATLAB untuk menghasilkan X,

```
m=[0; 2]';
S(:,1)=[0. 2];
S(:,2)=[0. 2];
P=[1/3 2/3];
N=1000;
randn('s ee d' ,0) ;
[X]=gene ra te _gauss_ classes (m, S, P, N);
```

Langkah 1..

```
Jika di aumsikan pdf, aproksimsi diasumsikan  $x \in [-5, 5]$  ,
x=-5:0.1: 5;
pdfx=(1/3 ) (/ sq rt( \2* pi *0. 2) ) * exp ( - (x .^2 )/0 .4 )
+(2/3)*(1 /s qrt (2 *p i*0 .2 )) *ex p( -( (x- 2) .^2)/ 0.4 );
plot(x,pd fx) ; hold;
h=0.1;
pdfx_app rox =Parzen _gauss_ ke rn el( X, h, -5, 5) ;
plot(-5: h:5 ,p df x_a pp ro x,'r' );
```

Latihan 1.7.1

Ulangi contoh soal 1.7.1 dengan nilai $h = 0.01, N = 1000$ and $h = 0.1, N = 10,000$.
Berikan komentar pada hasilnya

Latihan 1.7.2

Di generate sejumlah data point $N = 1000$ dari 2-dime nsio nal pdf

$$p(x) \equiv p(x(1), x(2)) = \frac{1}{3} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x_1^2(1) + x_1^2(2)}{2\sigma^2}\right) + \frac{2}{3} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x_1^2(1) + (x(2) - 2)^2}{2\sigma^2}\right)$$

Ulangi percobaan pada contoh soal 1.7.1 .

Latihan 1. 7. 3

Dengan menggunakan prinsip klasifikasi pada contoh 1.4.3, klasifikasikan himpunan data point x_1 dengan menggunakan Bayesian classifier, dimana nilai $p(x|\omega_1)$, $p(x|\omega_2)$ dapat diestimasi menggunakan parzen window, dengan menggunakan h yang berbeda hitunglah best error performance of the classifier

1.8. k - NEAREST NEIGHBOR DENSITY ESTIMATION

K-Nearest Neighbor merupakan sebuah teknik yang dapat digunakan untuk mengestimasi nilai PDF pada himpunan data sejumlah N yaitu $x_1, x_2, \dots, x_N \in \mathcal{R}^1$ yang diperoleh dari distribusi statistik yang tidak diketahui. Langkah-langkah dalam Teknik k-nearest neighbor adalah sebagai berikut :

- 1.2 Menentukan parameter K (jumlah tetangga paling dekat).
2. Menghitung jarak antara x dengan data pelatihan x_i , $i = 1, 2, \dots, N$ (Perhitungan jarak menggunakan euclidean dan mahalanobis)
3. Pilih k terdekat terhadap jarak x
4. Hitung volume $v(x)$ dimana ditemukan nilai k terdekat
5. Hitung probabilitas dengan

$$p(x) \approx \frac{k}{NV(x)}$$

Jika nilai euclidean distance dari jarak terjauh k dengan nilai x adalah ρ maka volume $v(x)$ dapat dihitung menggunakan

$$v(x) = 2\rho \rightarrow \text{dimensi 1}$$

$$v(x) = \pi \rho^2 \rightarrow \text{dimensi 2}$$

$$v(x) = \frac{4}{3} \rho^3 \rightarrow \text{dimensi 3}$$

Atau yang lebih general dapat digunakan dalam [Theo 09, Section 2.4.1]

Latihan 1.8.1

Himpunan data yang diperoleh pada latihan 1.7.1 gunakan estimator PDF menggunakan k-nearest neighbor untuk $k=21$

Solution

Gunakan fungsi built-in MATLAB `knn_density_estimate`, sebagai estimatornya

```
pdfx_approx=knn_density_estimate(X,21,-5,5,0.1);
plot(-5:0.1:5,pdfx_approx,'r');
```

Latihan 1. 8. 1

Ulangi contoh 1.8.1 untuk nilai k = 5 dan 100 . serta nilai N = 5000 .

Latihan 1. 8. 2

Dengan menggunakan classification task dalam contoh 1.4.3 . Klasifikasikan data points himpunan X 1 menggunakan *Bayesian classifier*. Estimate nilai $p(x|\omega_1)$, $p(x|\omega_2)$ untuk setiap X1 menggunakan metode *k-nearest neighbor density*. Dengan menggunakan nilai k yang berbeda hitunglah *best error performance of the classifier*.

1.9. THE NAIVE BAYES CLASSIFIER

Klasifikasi ini dapat digunakan untuk mengestimasi nilai PDF dari himpunan $x = [x(1), x(1), x(1)]^T \in \mathbb{R}^1$

$$p(x)=\prod_{j=1}^l p(x(j))$$

Himpunan vektor ciri x diasumsikan bersifat statstically independent [Theo 09, Section 2.5.6] asumsi ini dapat digunakan untuk *high dimensional data spaces*.

Contoh Soal 1.9.1

Himpunan x terdiri data N = 50 dalam data 2 dimensi dengan berbentuk himpunan class yang sama dimana class dimodelkan dalam distribusi gauss $m_1=[0,0,0,0,0]$ dan $m_2=[1,1,1,1,1]$ dengan *matrix covariances*

$$S_1=\begin{bmatrix} 0.8 & 0.2 & 0.1 & 0.05 & 0.01 \\ 0.2 & 0.7 & 0.1 & 0.03 & 0.02 \\ 0.1 & 0.1 & 0.8 & 0.02 & 0.01 \\ 0.05 & 0.03 & 0.02 & 0.9 & 0.01 \\ 0.01 & 0.02 & 0.01 & 0.01 & 0.8 \end{bmatrix} \quad S_2=\begin{bmatrix} 0.9 & 0.1 & 0.05 & 0.02 & 0.01 \\ 0.1 & 0.8 & 0.1 & 0.02 & 0.02 \\ 0.05 & 0.1 & 0.7 & 0.02 & 0.01 \\ 0.02 & 0.02 & 0.02 & 0.6 & 0.02 \\ 0.01 & 0.02 & 0.01 & 0.02 & 0.7 \end{bmatrix}$$

Langkah 1, Klasifikasikan titik X2 dengan menggunakan *Bayessian Classifier*, yaitu :

$$p(x|\omega_i)=\prod_{j=1}^5 \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(-\frac{(x(j)-m_{ij})^2}{2\sigma_{ij}^2}\right), i=1,2$$

Dimana

$x(j)$ komponen ke-j dari x . Hitung *error probability*.

Langkah 2, Hitung nilai perkiraan ML berdasar m1, m2, S1 dan S2 menggunakan X1, hitung error probability

Solution .

Generate himpunan vektor data X1 dan X2 , t ype

```
m=[zeros(5,1)ones(5,1)];
S(:, :, 1)= [      0 .8 0.2 0.1 0.05 0.01;
              0.2 0.7 0.1 0.03 0.02;
              0.1 0.1 0.8 0.02 0.01;
              0.05 0.03 0.02 0.9 0.01;
              0.01 0.02 0.01 0.01 0.8];
S(:, :, 2)= [      0 .9 0.1 0.05 0.02 0.01;
              0.1 0.8 0.1 0.02 0.02;
              0.05 0.1 0.7 0.02 0.01;
              0.02 0.02 0.02 0.6 0.02;
              0.01 0.02 0.01 0.02 0.7];
P=[1/2 1/2]';
N_1=100;
randn('state', 0) ;
[X1,y1]=generate_gauss_classes(m,S,P,N_1);
N_2=10000;
randn('state',1 00);
[X2,y2]= generate_gauss_classes(m,S,P,N_2);
```

Dengan asumsi bahwa ciri bersifat independent, dengan menggunakan fungsi Gaussian_ML_estimate untuk menghitung mean dan variance setiap ciri dari sebuah class (berdasarkan himpunan X1).

```
for i=1:5
[m1_hat(i),S1_hat(i)]=Gaussian_ML_estimate (X1(i,find(y1==1)));
end
m1_hat=m1_hat'; S1_hat=S1_hat';
for i=1:5
[m2_hat(i),S2_hat(i)]=Gaussian_ML_estimate(X1(i,find(y1==2)));
end
m2_hat=m2_hat'; S2_hat=S2_hat'
```

Kemudian

Langkah 3 Untuk mengklasifikasikan setiap titik dalam X_2 berdasarkan pada teori klasifikasi Bayes, sebagai berikut :

```
for i = 1:5
perFeature 1(i,:) = normpdf(X2(i,:), m1_hat(i), sqrt(S1_hat(i)));
perFeature 2(i,:) = normpdf(X2(i,:), m2_hat(i), sqrt(S2_hat(i)));
end
naive_probs1 = prod(perFeature1);
naive_probs2 = prod(perFeature2);
classified = ones(1, length(X2));
classified(find(naive_probs1 < naive_probs2)) = 2;
```

Menghitung *classification error*

```
true_labels = y2;
naive_error = sum(true_labels ~= classified) / length(classified)
```

Langkah 3, untuk menghitung perkiraan nilai *maximum likelihood*

```
[m1_ML, S1_ML] = Gaussian_ML_estimate(X1(:, find(y1==1)));
[m2_ML, S2_ML] = Gaussian_ML_estimate(X1(:, find(y1==2)));
```

Untuk mengklasifikasikan setiap titik dalam X_2 berdasarkan pada teori klasifikasi Bayes

```
m_ML(:, 1) = m1_ML;
m_ML(:, 2) = m2_ML;

m_ML(:, 1) = m1_ML;
m_ML(:, 2) = m2_ML;
S_ML(:, :, 1) = S1_ML;
S_ML(:, :, 2) = S2_ML;
P = [1/2 1/2];
z = bayes_classifier(m_ML, S_ML, P, X2);
```

Hitung nilai *classification error*

```
true_labels = y2;
Bayes_ML_error = sum(true_labels ~= z) / length(z)
```

Hasil

classification errors naive_error and Bayes_ML_error bernilai 0.1320 dan 0.1426, sehingga dapat disimpulkan bahwa naive classification lebih baik performanya dari pada ML-based standard

Latihan 1.9.1

1. Klasifikasikan himpunan X_2 pada contoh 1.9.1 menggunakan optimal Bayesian classifier. Dengan means dan covariance matrices dalam ruang 5 dimensi Gaussian pdfs. Badingkan hasilnya
2. Ulangi contoh 1.9.1 dengan X_1 consisting memiliki jumlah $N_1 = 1000$ data

Catatan

Berdasarkan beberapa contoh sebelumnya diperlihatkan bahwa teknik ini akan memberikan nilai optimal jika data yang dipergunakan cukup banyak serta akan memberikan nilai estimasi yang kurang baik, dan ini hanya berlaku pada *high dimensional spaces* [Theo 09, Section 2.5.6]

1.10. THE NEAREST NEIGHBOR RULE

Nearest Neighbor (NN) merupakan satu teknik klasifikasi klasik yang populer, dalam teknik ini setiap data dalam sebuah himpunan anggota x ($x \in \mathcal{R}^l$) akan diklasifikasikan kedalam *class* ($class_{\omega_i}$, $i = 1, 2, \dots, c$), berdasarkan data pelatihan sejumlah N yaitu x_i , $i = 1, 2, \dots, N$, dalam sebuah ruang berdimensi l . Teknik *Nearest Neighbor* (NN) memiliki beberapa langkah sebagai berikut :

1. Menentukan parameter k (tidak harus keelipatan jumlah *class*), yang merupakan sebagai tetangga dari x , menggunakan fungsi jarak yaitu *Euclidean*, *Mahalanobis*).
2. Jumlah k sebagai tetangga terdekat dapat ditentukan lewat persamaan $\sum_{i=1}^c k_i = k$
3. Menetapkan x sebagai *class* ω_i dimana kondisi $k_i > k_j$, $j \neq i$ maka x dimasukan kedalam *class* berdasarkan nilai k terdekat

Untuk nilai N yang besar performa k -NN baik, namun untuk N yang kecil performanya menurun. [Theo09, Section 2.6], masalah utama dalam klasifikasi ini adalah untuk ruang data dengan dimensi tinggi (*high dimensional spaces*)

Contoh 1.10.1

1. Klasifikasi vektor data dalam ruang dimensi 2, dengan vektor data berasal dari *class* ω_1 dan ω_2 dengan obabilitas sama (*equiprobable*). Setiap *class* tersebut dimodelkan menggunakan distribusi gaussian $m_1 = [0, 0]$ dan $m_2 = [1, 1]$ dengan covariance matrices $S_2 = S_1 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$
Carilah vektor himpunan X_1 dan X_2 , untuk $N=1000$ dan 5000
2. Dengan vektor himpunan X_1 sebagai data pelatihan, klasifikasikan sebuah titik dalam vektor himpunan X_2 menggunakan aturan k -NN classifier, dengan $k = 3$, menggunakan fungsi jarak *Euclidean distance*, serta hitung kesalahan klasifikasi (*classification error*)

Jawaban

Langkah 1. Tentukan vektor himpunan X_1 and X_2

```
m=[0 0; 1 2]';  
S=[0.8 0.2; 0.2 0.8];
```

```

S(:, :, 1) = S;
S(:, :, 2) = S;
P = [1/2 1/2]';
N_1 = 1000;
randn('seed', 0)
[X1, y1] = generate_gauss_classes(m, S, P, N_1);
N_2 = 5000;
randn('seed', 100)
[X2, y2] = generate_gauss_classes(m, S, P, N_2);

```

Langkah 2. Untuk proses klasifikasi gunakan fungsi *k_nn_classifier*

```

K = 3;
Z = k_nn_classifier(X1, y1, K, X2);

```

Untuk menghitung nilai *classification error*

```
pr_err = sum(z ~= y2) / length(y2)
```

Nilai *classification error* adalah 15.12%

Latihan 1.10.1

Ulangi langkah 1.10.1 untuk nilai $k = 1, 7, 15$. hitung masing-masing nilai error klasifikasi (*classification error*), serta bandingkan hasilnya dengan nilai error hasil perhitungan menggunakan *optimal Bayesian classifier*, dengan menggunakan nilai rata-rata (*mean*) dan *covariance matrices*.

Latihan 1.10.2

Buatlah contoh yang akan mengklasifikasikan data klasifikasi dalam ruang berdimensi 5 kedalam dua class. Asumsikan bahwa data dalam setiap *class* mengikuti fungsi probabilitas gaussian. Pilih nilai mean dan *covariance matrices*, dua himpunan data yang dihasilkan digunakan sebagai pelatihan dan pengujian, dengan menggunakan *nearest neighbor classifier*, menghitung nilai *classification error*. Lakukan percobaan ulang dengan nilai yang berbeda untuk nilai *mean*, *covariance matrices*, k , dan panjang himpunan data pelatihan. Berikan komentar atas hasil yang diperoleh tersebut.