MSPR 5: Probabilities

Dr. Hendrik Purwins

AAU CPH

October 5, 2015

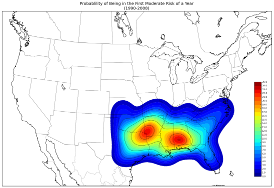
This lecture is based on presentations made in my course 'Advanced Topics in in Music Technology' at Music Technology Group, Universitat Pompeu Fabra between 2007-2011, Barcelona, especially the ones by Stefan Kersten and Srikanth Cherla, closely following

- *Andrew Moore's machine learning tutorial lectures: Gaussians, Gaussian Mixture Models,* `http://www.autonlab.org/tutorials/`
- *Christopher Bishop: Pattern Recognition and Machine Learning: Chapter 1 (Introduction) 1.2.3 (The Gaussian Distribution) p. 24 - 27, 2.3 (The Gaussian Distribution) p. 78, 84 bottom - 85 top.*

Outline

Applications of Probability Density Estimation

- Of few customers we know the income. Do we have more customers with income around 300 000 DKK or 1 000 000 DKK?
- Estimation of crime rates in different areas before buying a house there
- Updating information in computer vision (on-line object tracking) as new data becomes available
- Stormrisks in the US (Image)



http://www.pmarshwx.com/blog/2011/02/03/aotw-storm-prediction-center-moderate-high-risks/
http://www.cs.utah.edu/~lifeifei/papers/kernelsigmod13.pdf

Random Variables

- *Random variable Part* of the world whose *exact value* is uncertain
- *Boolean variables* $\{\mathrm{true}, \mathrm{false}\}$, e.g. *Crown Prince Frederik will win the next KMD Ironman Copenhagen*
- *Discrete variables* take a value from a countable domain, e.g. *this piece's genre is one of* $\{$*jazz, hiphop, country*$\}$
- *Continuous variables* take a value from a continuous domain, e.g. *this piece's tempo is close to 94.5 bpm*

- From the *Kolmogorov's axioms* it follows (a simple foundation of probability theory)

$$
\begin{aligned}
&0 \le P(A) \le 1 \\
&P(\mathrm{true}) = 1 \\
&P(\mathrm{false}) = 0 \\
&P(A \vee B) = P(A) + P(B) - P(A \wedge B)
\end{aligned}
$$

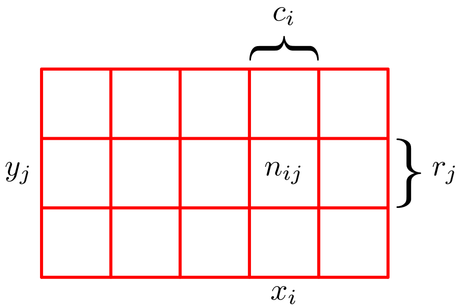
- The *joint distribution* denotes the probabilities of all combinations of two or more random variables
- *Conditional probability*

$$P(Y|X) = \frac{P(Y, X)}{P(X)}$$

- *Sum rule*

$$P(X) = \sum_Y P(X, Y)$$

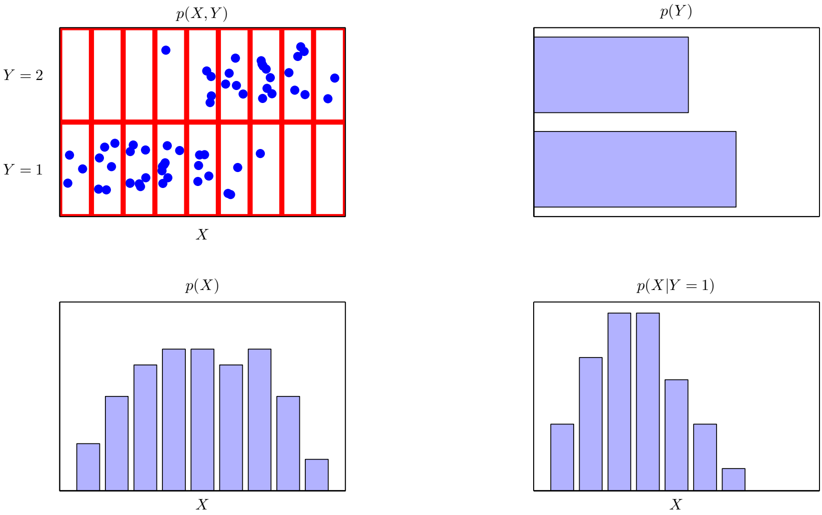
- *Marginalization* over the variable Y given the joint probability $P(X, Y)$



- *Product rule*

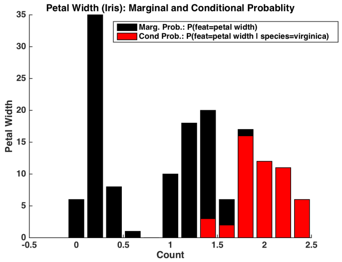
$$P(X, Y) = P(Y|X)P(X)$$

Q: Calculate joint, marginal, and conditional probabilites from a co-occurrence table.



$p(X,Y)$
$Y = 2$
$Y = 1$
X
$p(Y)$
$p(X)$
X
$p(X|Y = 1)$
X

For the Iris data set, the histogram as an estimate for the conditional Probability of petal width under the condition that we have iris species virginica:
P(feat='petal width' — species='virginica'):



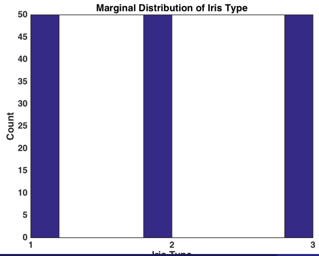

```
adata = textscan(fid,'%f%f%f%f%s','delimiter',',');
X=[adata{1} adata{2} adata{3} adata{4}];
idx_virginica=find(strcmp(species,'Iris-virginica')==1);
idx_versicolor=find(strcmp(species,'Iris-versicolor')==1);
[hvir bvir]=hist(X(idx_virginica,4),(0:0.2: 2.5));
 [hmarg bmarg]=hist(X(:,4),(0:0.2: 2.5));
bar(bmarg,hmarg,'k'); hold on; bar(bvir,hvir,'r');
```

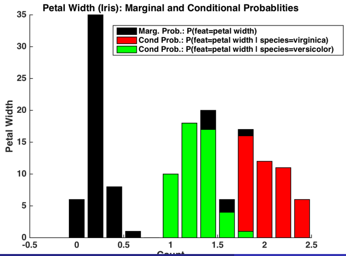
Class Assignment

- *For the Iris data set, the histogram as an estimate for the conditional Probability of petal width under the condition that we have iris species versicolor:*
  P(petal width |species='versicolor').
- *By a histogram, give the marginal probability of the iris type.*

```
idx_virginica=find(strcmp(species,'Iris-virginica')==1);
idx_versicolor=find(strcmp(species,'Iris-versicolor')==1);
[hvir bvir]=hist(X(idx_virginica,4),(0:0.2: 2.5));
[hmarg bmarg]=hist(X(:,4),(0:0.2: 2.5));
[hver bver]=hist(X(idx_versicolor,4),(0:0.2: 2.5)); hold on;
bar(bmarg,hmarg,'k'); bar(bvir,hvir,'r');bar(bver,hver,'g');

iris_type=strcmp(species,'Iris-virginica')+2*strcmp(species,'
    Iris-versicolor')+3*strcmp(species,'Iris-setosa');
hist(iris_type)
```





Q: Derive Bayes rule from the sum and the product rule.

- Bayes' rule is derived from the *product rule*:

$$\begin{aligned} P(X,Y) &= P(Y,X) \\ P(X|Y)P(Y) &= P(Y|X)P(X) \end{aligned}$$

- *Bayes' rule* then becomes

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- Bayes rule can be stated as

$$posterior \propto likelihood \ x \ prior$$

Independence

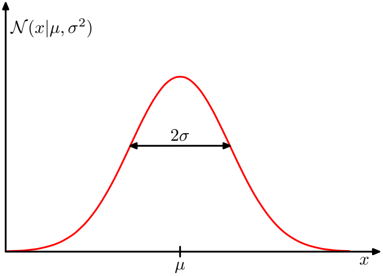
- We say that X is independent of Y, if knowing Y doesn't influence our degree of belief in X
- $P(X|Y) = P(X)$

- *Normal distribution* of a single variable x (*univariate*)

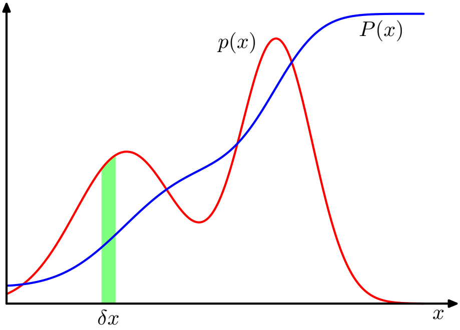
$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

- *Mean* μ and *Variance* σ^2
- Desirable properties for a PDF

  Positive $\mathcal{N}(x|\mu,\sigma^2) > 0$



  Normalized $\int_{-\infty}^{\infty} \mathcal{N}(x|\mu,\sigma^2)dx = 1$



The probability of x is given by probability distribution $p(x)$.

```
mu = 0; sigma = 2;
pd = makedist('Normal',mu,
    sigma);
```

The probability of $x \in (-\infty, z)$ is given by the *cumulative distribution function* (CDF) $P(x)$ Matlab `y = cdf(pd,x)` Consider a probability in the range δx E.g. that a normally distributed random variable falls within $\pm\sigma$:

```
y = cdf(pd,[a b])}
x = [-sigma sigma]; y =
    cdf(pd,x)y(2)-y(1)
```

Result: Probability: 0.6827

Class Assignment

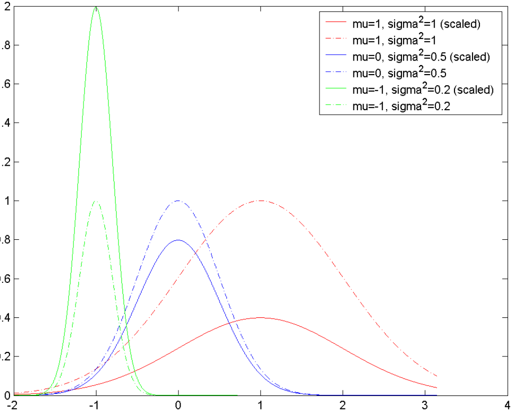
Use the cumulative distribution function in Matlab to calculate what the probability is that a normally distributed random variable falls farther away than 2 σ from the mean μ. How does this value relate to the significance level α of a statistical test?

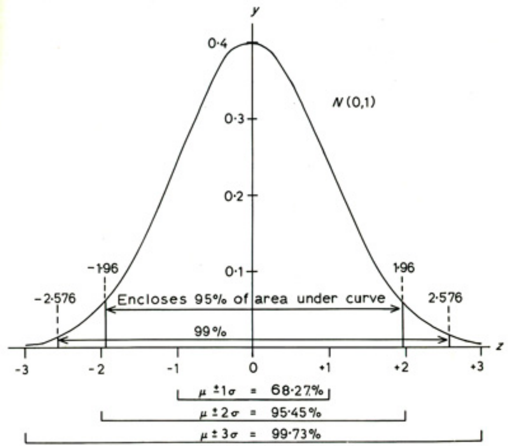
Use the cumulative distribution function in Matlab to calculate what the probability is that a normally distributed random variable falls farther away than 2 σ from the mean μ. How does this value relate to the significance level α of a statistical test?

```
mu = 0; sigma = 2; pd = makedist('Normal',mu,sigma);
x = [-2*sigma 2*sigma]; y = cdf(pd,x)
y(1) + 1-y(2) % or
2*y(1)
```

Result: 0.0455
Significance level $\alpha = 0.05$ gives an upper limit of the probability that, when rejecting the Nullhypothesis it may still be true.

Examples of Gaussians with Different Means / Variances




y
0·4
0·3
0·2
0·1
N (0,1)
−1·96
1·96
−2·576
2·576
Encloses 95% of area under curve
99%
z
-3
-2
-1
0
+1
+2
+3
μ ±1σ = 68·27%
μ ±2σ = 95·45%
μ ±3σ = 99·73%

Fitting a Gaussian to a Histogram of 1-D Data

Data: I observations of just one feature (1-D, just a number=scalar):

$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_I \end{bmatrix}$. Fit a 1-dimensional Gaussian to the data histogram by finding the right estimates $\bar{x}$ for mean μ and s^2 for σ^2:

$$\mathcal{N}(x|\bar{x}, s^2) = \frac{1}{\sqrt{2\pi s^2}} e^{\frac{(x-\bar{x})^2}{2s^2}} \quad (\mathrm{Matlab}: \mathtt{n} = \mathtt{pdf}('\mathtt{norm}', \mathtt{x}, \mathtt{xm}, \mathtt{sqrt}(\mathtt{s2}));)$$

$$\bar{x} = \frac{1}{I} \sum_{i=1}^{I} x_i \quad (\text{sample mean},\ \mathrm{Matlab}: \mathtt{mean}(\mathtt{x}))$$

$$s^2 = \frac{1}{I-1} \sum_{i=1}^{I} (x_i - \bar{x})^2 \quad (\text{sample variance},\ \mathrm{Matlab}: \mathtt{var}(\mathtt{x}))$$

Class Assignment

Load the data `gauss.mat` *and fit a Gaussian to the data A0_tr How good is the fit? Use Matlab functions* `[h p]=hist(x), bar(p,h,'b')` *. Plot the pdf in the same plot. Normalize the height of the histogram bars* `h` *by dividing it with the distance* `b(2) - b(1)` *between two bar positions and the number of components in the data vector of that class.*

Solution

```
M0=mean(A0_tr); S0=var(A0_tr);
tr_no= size(A0_tr,1);
x=-10:0.1:10;
h=figure; n0=pdf('norm',x,M0,sqrt(S0));
[h0 p0]=hist(A0_tr);
bar(p0,h0/size(A0_tr,1)/(p0(2)-p0(1)),'b'); hold on; plot(x,
    n0,'b')
% Plot gaussians
```

Covariance Matrix I

- Covariance between feature j (j-th data matrix column) and k (k-th data matrix column):

$$\mathrm{cov}(\mathbf{X}_{[:,j]}, \mathbf{X}_{[:,k]}) = \frac{1}{I-1} \sum_{i=1}^{I} (x_{ij} - \bar{\mathbf{X}}_{[:,j]})(x_{ik} - \bar{\mathbf{X}}_{[:,k]})$$

- $J \times J$ covariance matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathrm{var}(\mathbf{X}_{[:,1]}) & \mathrm{cov}(\mathbf{X}_{[:,1]}, \mathbf{X}_{[:,2]}) & \dots & \mathrm{cov}(\mathbf{X}_{[:,1]}, \mathbf{X}_{[:,J]}) \\ \mathrm{cov}(\mathbf{X}_{[:,2]}, \mathbf{X}_{[:,1]}) & \mathrm{var}(\mathbf{X}_{[:,2]}) & \dots & \mathrm{cov}(\mathbf{X}_{[:,2]}, \mathbf{X}_{[:,J]}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(\mathbf{X}_{[:,J]}, \mathbf{X}_{[:,1]}) & \mathrm{cov}(\mathbf{X}_{[:,J]}, \mathbf{X}_{[:,2]}) & \dots & \mathrm{var}(\mathbf{X}_{[:,J]}) \end{bmatrix}$$

The diagonal elements are the variances of each features $1, \dots, J$ Matrix is symmetric

Covariance

1. Sample covariance Matrix $\mathbf{C}$ of all features (columns of centirized data matrix $\mathbf{X}_c$):

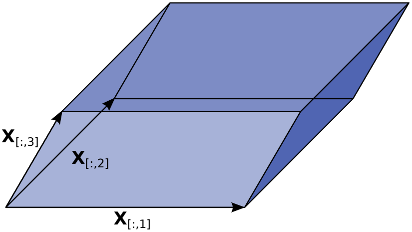
$$\mathbf{C} = \frac{1}{I-1}\mathbf{X}_c^T\mathbf{X}_c$$

2. Matlab: `C=Xc'*Xc/(I-1)` or `C=cov(X)`

Determinant

$$\mathbf{X} := \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

The determinant det assigns a number to square matrix $\mathbf{X}$: $c = \det(\mathbf{X})$, Matlab: `c=det(X)`.



- The volumn V of the object (parallepiped) that is sided by the parallelograms defined by the column vectors $\mathbf{X}_{[:,1]}, \mathbf{X}_{[:,2]}, \mathbf{X}_{[:,3]}$ of the matrix $\mathbf{X}$ is the absolut value of the determinant of $\mathbf{X}$: $V = |det(\mathbf{X})|$.
- If $\det(\mathbf{X}) \neq 0$, $\mathbf{X}$ is invertible, for $\det(\mathbf{X}) = 0$, $\mathbf{X}$ is *not* invertible.

Determinant

Class Assignment

Calculate the determinant of the following matrices and invert them if possible and show that you found the inverse. What was the definition of invertibility for a matrix?

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix}, A_3 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

```
A1=[1 0; 0 1]; A2=[1 2; -1 -2]; A3=[1 -0.5; -0.5 1];
det(A1); A1I=inv(A1); A1*A1I % det(A1)=1
det(A2); inv(A2) % det(A1)=0 (A2 NOT INVERTIBLE!)
det(A3);A3I=inv(A3);A3*A3I % det(A1)=0.75
```

Multi-variate Gaussians

- The multivariate Gaussian distribution for a J-dimensional variable $\mathbf{x}$ is given by

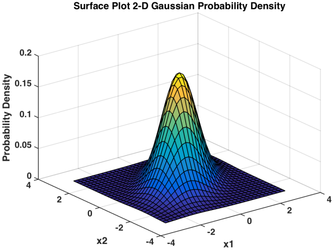
$$\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{2\pi^{J/2} \det \mathbf{\Sigma}^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}$$

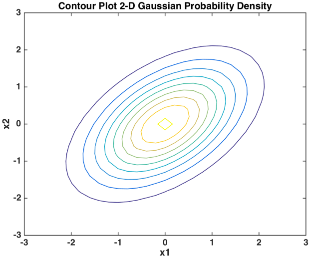
- Defining parameters: μ and $\mathbf{\Sigma}$
- Mean and Covariance
  - Mean (vector) $= \mu$ (J dimensional)
  - Covariance (matrix) $= \mathbf{\Sigma}$ ($J \times J$)
  - $\det \mathbf{\Sigma}$ denotes the determinant of $\mathbf{\Sigma}$

Example Multivariate Gaussian

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

x_1/x_2 axis correlated






```
mu = [0 0]; Sigma1 = [1 .5; .5 1];
x1 = -3:.2:3; x2 = -3:.2:3; [X1,X2] = meshgrid(x1,x2);
F = mvnpdf([X1(:) X2(:)],mu,Sigma1);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
figure; contour(X1,X2,F)
```

Class Assignment

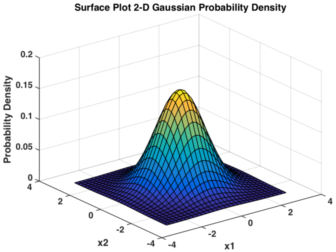
For the following covariance matrices predict the shape of the normal distribution using that covariance matrix and then plot the distribution as a verification:

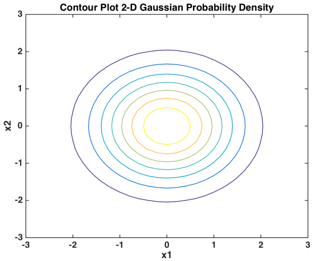
$$\mathbf{\Sigma}_2 = \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{\Sigma}_5 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix} \qquad \mathbf{\Sigma}_3 = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix} \qquad \mathbf{\Sigma}_4 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\mathbf{\Sigma}_2 = \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

x_1 / x_2-axis uncorrelated

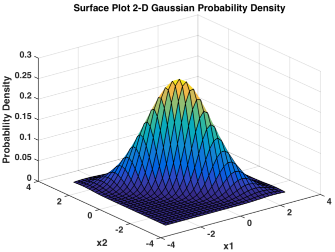


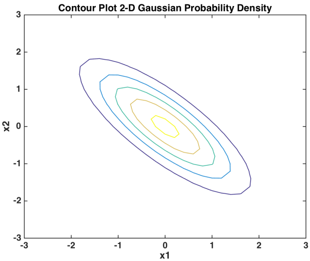



```
mu = [0 0]; Sigma2 = [1 0; 0 1];
x1 = -3:.2:3; x2 = -3:.2:3; [X1,X2] = meshgrid(x1,x2);
F = mvnpdf([X1(:) X2(:)],mu,Sigma1);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
figure; contour(X1,X2,F)
```

$$\mathbf{\Sigma}_3 = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

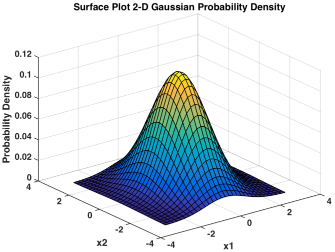
x_1/x_2-axis negatively correlated

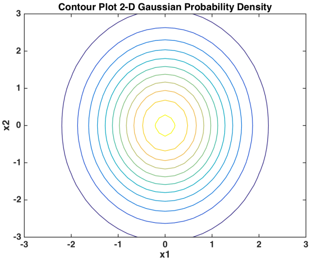




$$\mathbf{\Sigma}_4 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

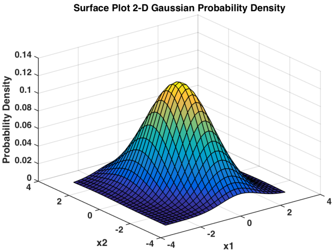
x_1/x_2-axis uncorrelated, stretched in x_2 direction

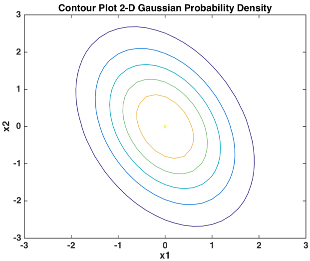




$$\boldsymbol{\Sigma}_5 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}$$

x_1/x_2-axis negatively correlated stretched in x_2 direction





Fitting a Gaussian to Multdimensional Data

Data: I observations of J features:

$$\mathbf{X} := \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2J} \\ x_{31} & x_{32} & \cdots & x_{3j} & \cdots & x_{3J} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{iJ} \\ & & \cdots & & \cdots & \\ x_{I1} & x_{I2} & \cdots & x_{Ij} & \cdots & x_{IJ} \end{pmatrix}$$

Sample mean vector $\bar{\mathbf{x}}$ estimates true mean vector μ (Matlab: `mean(X)`):

$$\bar{\mathbf{x}} = \frac{1}{I} \underbrace{(1, 1 \ldots, 1)}_{I\times} \mathbf{X}$$

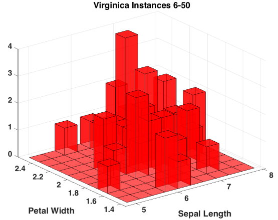
Fit a J-dimensional Gaussian to the data by finding the right estimates for the mean vector μ for covariance matrix σ: $\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma})$

$$= \frac{1}{2\pi^{J/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}$$

Sample covariance matrix $\mathbf{C}$ estimates true covariance matrix $\mathbf{\Sigma}$ (Matlab: `cov(X)`)

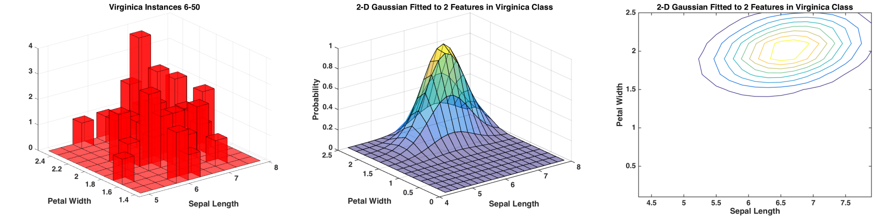
$$C = \frac{1}{I-1}\mathbf{X}'\mathbf{X}$$

3-D Histogram of components 'Sepal Length' and 'Petal Width' only for the Virginica Iris type




```
fid = fopen([data_path 'iris.data']);
adata = textscan(fid,'%f%f%f%f%s','
    delimiter',',');
fclose(fid);
X=[adata{1} adata{4}];
species=adata{5};
idx_virginica=find(...
  strcmp(species,'Iris-virginica')==1);
idx_virginica_tr=idx_virginica(6:50);
mean_virginica=...
  mean(X(idx_virginica_tr,:));
sig_virginica=cov(X(idx_virginica_tr,:))
hist3(X(idx_virginica_tr,:),...
  'FaceAlpha',.65,'FaceColor','red');
```

Fitting Gaussian with Full Cov. Matrix to 3-D Histogram

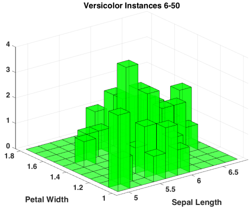


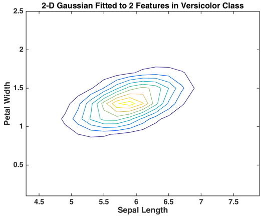

```
x1 = min(X(:,1)):.2:max(X(:,1));
x2=min(X(:,2)):.2:max(X(:,2));
[X1,X2] = meshgrid(x1,x2);
F_virginica = ...
    mvnpdf([X1(:) X2(:)],mean_virginica,sig_virginica);
F_virginica = reshape(F_virginica,length(x2),length(x1));
surf(x1,x2,F_virginica,'FaceAlpha',.5);
contour(X1,X2,F_virginica);
```

Class Assignment

Plot 3-D histogram for instances 5-50 of features 'Sepal Length' and 'Petal Width' the versicolor iris species. Fit a 2-D Gaussian to the histogram and do a surface and contour plot it.

Plot 3-D histogram for instances 5-50 of features 'Sepal Length' and 'Petal Width' the versicolor iris species. Fit a 2-D Gaussian to the histogram and do a surface and contour plot it.





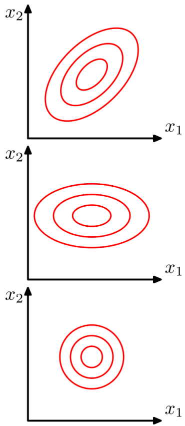

```
...
hist3(X(idx_versicolor_tr,:),'FaceAlpha',.65,'FaceColor','green');
surf(x1,x2,F_versicolor,'FaceAlpha',.5,'FaceColor','green');
```

Parameters to Estimate

- In its most general form, the multi-variate Gaussian of J dimensions has
  - $J(J+1)/2$ parameters for the covariance matrix
  - J parameters for the mean vector.
- The number of parameters increases quadratically with J and hence poses a problem both in parameter estimation and matrix inversion.

Reduction of Parmeters to be Estimated

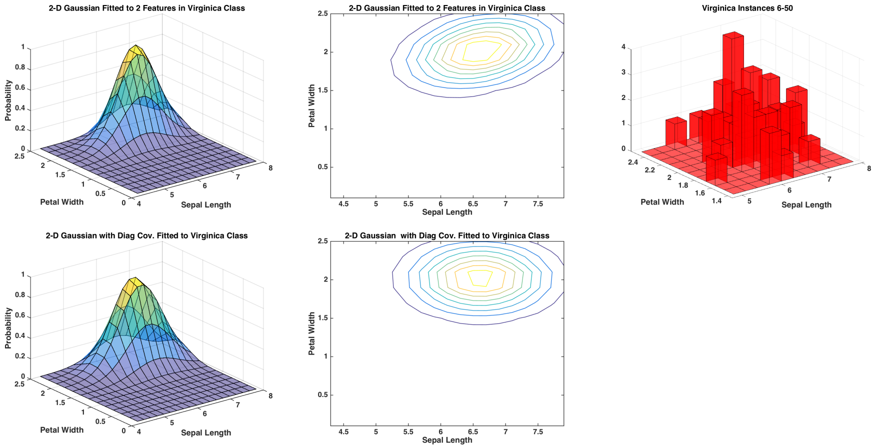
There are two ways of simplifying this problem



- Assume diagonal covariance matrix ($2 \times J$ parameters).
- Assume equal covariance $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ ($(J + 1)$ parameters).

However, these simplifications also limit the modeling capability of the Gaussian

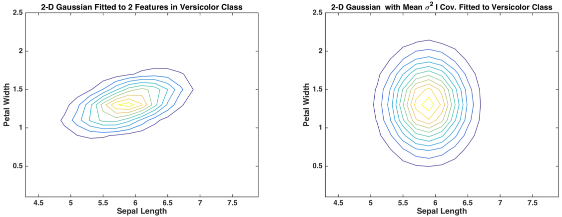
Fit of the Virginica Iris Class with a Gaussian with Diagonal Covariance Matrix



Class Assignment

Fit a Gaussian with covariance matrix of type $\sigma = \sigma^2 \mathbf{I}$ to 3-D histogram of features 'Sepal Length' and 'Petal Width' of the versicolor class considering instance 6-50. For σ^2 take the average of the variances for 'Sepal Length' and 'Petal Width' .

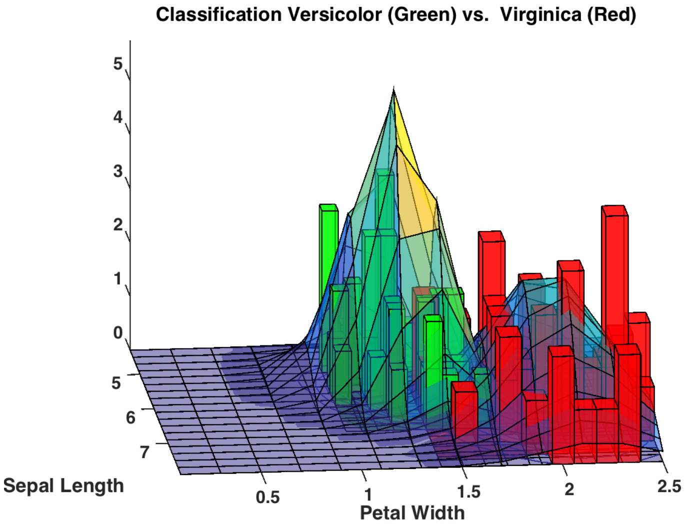
Fit a Gaussian with covariance matrix of type $\sigma = \sigma^2 \mathbf{I}$ to 3-D histogram of features 'Sepal Length' and 'Petal Width' of the versicolor class considering instance 6-50.




```
sig_versicolor=mean(diag(sig_versicolor))*diag(ones(2,1));
```

Gaussian Fit for Predicting the Iris Type Based on Sepal Length and Petal Width only

- We had fitted two Gaussians $\mathcal{N}(\mathbf{x}|\mu_{virg}, \mathbf{\Sigma}_{virg})\mathcal{N}(\mathbf{x}|\mu_{vers}, \mathbf{\Sigma}_{vers})$ to the points 6-50 of Virginica and Versicolor Iris data ('Sepal Length' and 'Petal Width').
- The first 5 points of Virginica and Versicolor iris data ('Sepal Length' and 'Petal Width') have not been used for the fitting.
- Let us use the fitted Gaussians, to determine whether these points belong to Virginica or Versicolor according to the rule: If for a point $\mathbf{x}$ $\mathcal{N}(\mathbf{x}|\mu_{virg}, \mathbf{\Sigma}_{virg}) > \mathcal{N}(\mathbf{x}|\mu_{vers}, \mathbf{\Sigma}_{vers})$ we predict its class to be Virginica, otherwise Versicolor.
- let us count the wrong predictions.


Classification Versicolor (Green) vs. Virginica (Red)
5
4
3
2
1
0
5
6
7
Sepal Length
0.5
1
1.5
2
2.5
Petal Width

```
sig_virginica=cov(X(idx_virginica_tr,:));
sig_versicolor=cov(X(idx_versicolor_tr,:));
idx_virginica_tst=idx_virginica(1:5);
idx_versicolor_tst=idx_versicolor(1:5);
F_versicolor_tst = mvnpdf(X([idx_virginica_tst;
    idx_versicolor_tst],:),...
     mean_versicolor,sig_versicolor);
F_virginica_tst = mvnpdf(X([idx_virginica_tst;
    idx_versicolor_tst],:),...
     mean_virginica,sig_virginica);
[F_virginica_tst'; F_versicolor_tst']
%ans =
%   0.18  0.43  0.69   0.66   0.69  0.03  0.16  0.09  0.02  0.15
%   0.00  0.00  0.00   0.10   0.00  0.08  0.98  0.19  1.30  0.80
```

Perfect Prediction!

Class Assignment

- *Fit two Gaussians* $\mathcal{N}(\mathbf{x}|\mu_{virg}, \mathbf{\Sigma}_{virg})\mathcal{N}(\mathbf{x}|\mu_{vers}, \mathbf{\Sigma}_{vers})$ *using a covariance matrix of type* $\sigma^2\mathbf{I}$ *with* σ^2 *being the mean variance for 'Sepal Length' and 'Petal Width'. Use to the features 'Sepal Length' and 'Petal Width' of instances 6-50 of the Virginica and Versicolor Iris data.*
- *Use the fitted Gaussians, to determine whether these points belong to Virginica or Versicolor according to the rule: If for a point* $\mathbf{x}$ $\mathcal{N}(\mathbf{x}|\mu_{virg}, \mathbf{\Sigma}_{virg}) > \mathcal{N}(\mathbf{x}|\mu_{vers}, \mathbf{\Sigma}_{vers})$ *we predict its class to be Virginica, otherwise Versicolor.*
- *Count the wrong predictions for the first 5 points of Virginica and Versicolor iris data.*

Solution

```
sig_virginica=mean(diag(sig_virginica))*diag(ones(2,1));
sig_versicolor=mean(diag(sig_versicolor))*diag(ones(2,1));
F_versicolor_tst = mvnpdf(X([idx_virginica_tst;
    idx_versicolor_tst],:),mean_versicolor,sig_versicolor);
F_virginica_tst = mvnpdf(X([idx_virginica_tst;
    idx_versicolor_tst],:),mean_virginica,sig_virginica);
[F_virginica_tst'; F_versicolor_tst']
```

```
ans =
    0.33  0.17  0.39  0.48  0.58  0.22  0.34  0.31  0.02  0.36
    0.00  0.32  0.00  0.25  0.02  0.01  0.38  0.02  0.69  0.25
```

4 Errors!