パターン認識
Pattern Recognition

佐藤真一
Shin'ichi Satoh

国立情報学研究所
National Institute of Informatics

Apr 11, 2023

Purpose of this course

- Special emphasis will be laid on statistical pattern recognition theory
- These techniques are indispensable for media analysis, feature extraction, media conversion, and so on
- The course will NOT explain very recent pattern recognition and machine learning tools.
- But the course will focus on basics how core methods work.
- Project works such as actual pattern classification will be assigned upon necessity to deepen the understanding.

Schedule and credits

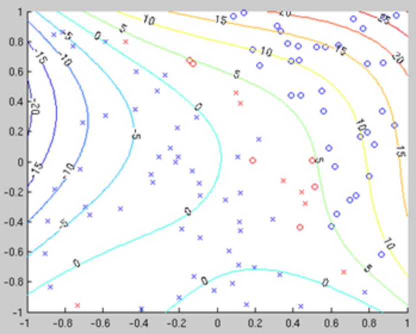
- Course web page: https://research.nii.ac.jp/∼satoh/utpr/
- Course materials, sample codes, and data will be posted (hopefully before each class).
- Language: Japanese, materials: Japanese and English
- Credits will be given based on final report (mandatory) and assignments (3 out of 7 are mandatory)
- Attendance record will NOT be taken

スケジュール (予定)

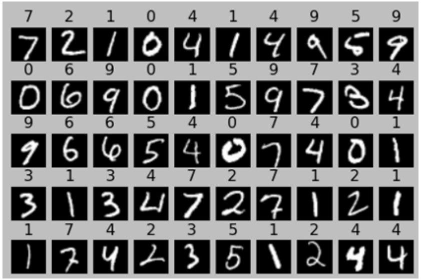
4/11	Orientation, Bayes decision theory, probability distribution
4/18	Random variable, random vector, normal distributions
4/25	Parametric density estimation, discriminant function
5/2	Nonparametric density estimation, Parzen windows, k-nearest neighbor estimate
5/9	k-nearest neighbor classification, classification error estimation
5/16	Bayes error estimation, classification error estimation, cross-validation, bootstrap
5/23	Linear classifier, perceptron, MSE classifier, Widrow-Hoff rule
5/30	neural network, deep learning
6/6	all about SVM
6/13	Orthogonal expansions, Eigenvalue decomposition
6/20	no class
6/27	Clustering, dendrogram, aggromerative clustering, k-means
7/4	Graphs, normalized cut, spectral clustering, Laplacian Eigenmaps
7/11	extra (if needed)

Example of assignment

Exercises and assignments impose analysis of synthetic data



Real data will also be used



Recommended textbooks

- R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, Inc., 1973.
- R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, John Wiley & Sons, 2001.
- K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.
- E. Oja, Subspace Methods of Pattern Recognition, John Wiley & Sons, Inc., 1983.
- 石井健一郎他, わかりやすいパターン認識, オーム社, 1998
- D. A. Forsyth and J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2003.
- D. H. Ballard and C. M. Brown, Computer Vision, Prentice Hall, 1982.

What is pattern recognition?

- To recognize patterns
- Observing raw data, and taking an action based on the category of the pattern of raw data
- Crucial for our life, and living creatures (including us) can do this very well

What can be solved by pattern recognition?

- perception of insects
- face recognition
- speech recognition
- document classification/understanding
- navigation and planning
- scene analysis

Example

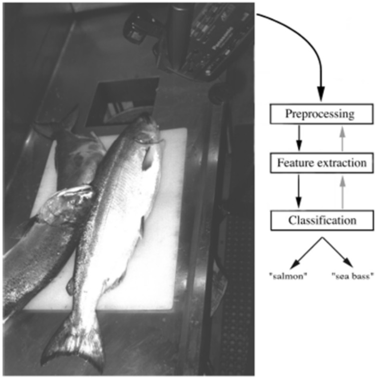


FIGURE 1.1. The objects to be classified are first sensed by a transducer (camera), whose signals are preprocessed. Next the features are extracted and finally the classification is emitted, here either "salmon" or "sea bass." Although the information flow is often chosen to be from the source to the classifier, some systems employ information flow in which earlier levels of processing can be altered based on the tentative or preliminary response in later levels (gray arrows). Yet others combine two or more stages into a unified step, such as simultaneous segmentation and feature extraction. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

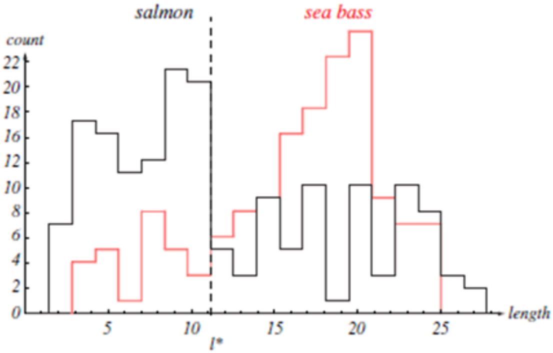


FIGURE 1.2. Histograms for the length feature for the two categories. No single threshold value of the length will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value marked l^* will lead to the smallest number of errors, on average. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

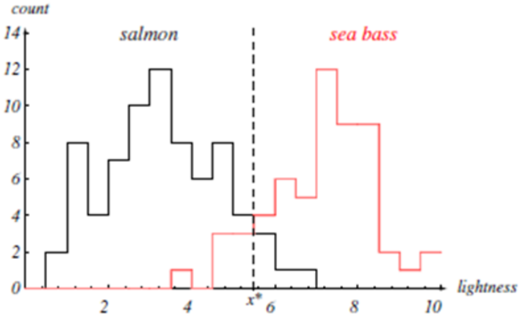


FIGURE 1.3. Histograms for the lightness feature for the two categories. No single threshold value x^* (decision boundary) will serve to unambiguously discriminate between the two categories; using lightness alone, we will have some errors. The value x^* marked will lead to the smallest number of errors, on average. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

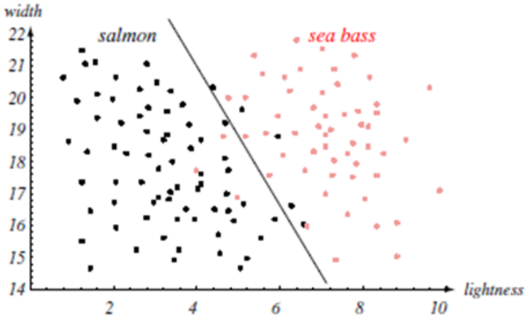


FIGURE 1.4. The two features of lightness and width for sea bass and salmon. The dark line could serve as a decision boundary of our classifier. Overall classification error on the data shown is lower than if we use only one feature as in Fig. 1.3, but there will still be some errors. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

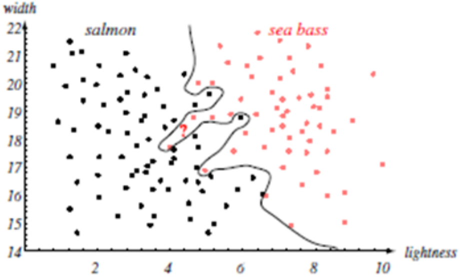


FIGURE 1.5. Overly complex models for the fish will lead to decision boundaries that are complicated. While such a decision may lead to perfect classification of our training samples, it would lead to poor performance on future patterns. The novel test point marked ? is evidently most likely a salmon, whereas the complex decision boundary shown leads it to be classified as a sea bass. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

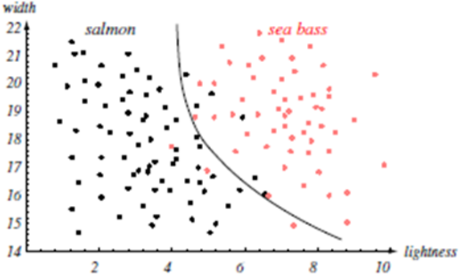
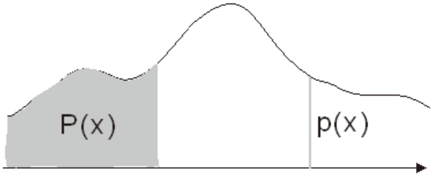
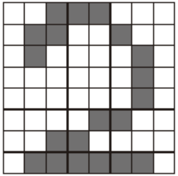


FIGURE 1.6. The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayesian decision theory

Probability: $P(\mathbf{x})$
Probability distribution: $p(\mathbf{x})$





Bayesian decision theory

- ω : a variable to denote the state (state of nature)
  e.g., $\omega = \omega_1$: fish is sea bass, $\omega = \omega_2$: fish is salmon
- Prior (a priori probability): our knowledge of how likely we observe the state e.g.,
  $P(\omega_1)$: prior that the next fish is sea bass
  $P(\omega_2)$: prior that the next fish is salmon
- class-conditional probability (density function): the probability density function for a continuous random variable $\mathbf{x}$ given that the state of nature ω
  e.g., $p(\mathbf{x}|\omega)$

Bayesian decision theory

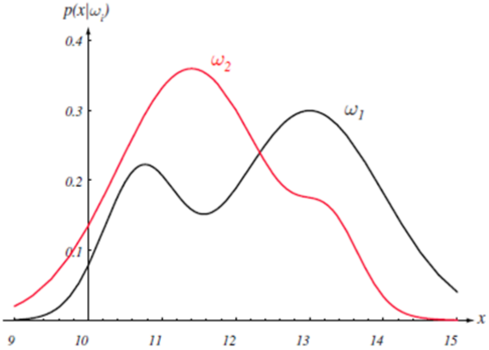


FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i. If x represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayesian decision theory

- A posteriori probability (posterior): the probability of the state of nature given that the feature value x has been observed for the random variable $\mathbf{x}$
  e.g., $P(\omega|x)$
- Likelihood: the likelihood of the state of nature with respect to the feature value
  e.g., $p(x|\omega)$
- Bayes formula

$$P(\omega|x) = \frac{p(x|\omega)P(\omega)}{p(x)}$$

$$p(x) = \sum p(x|\omega)P(\omega)$$

Exercise

- Three different machines M1, M2, and M3 were used for producing a large batch of similar manufactured items.
- Suppose that 20% of the items were produced by M1, 30% by M2, and 50% by M3.
- Suppose further than 1% of the items produced by M1 are defective, that 2% of the items produced by M2 are defective, and that 3% of the items produced by machine M3 are defective.
- Finally, suppose that one item is selected at random from the entire batch and it is found to be defective.
- Which machine produced this item? Determine the probability that this item was produced by M1, M2, and M3, respectively.

Hint

- Prior probability

$$P(M1) = 0.2$$
$$P(M2) = 0.3$$
$$P(M3) = 0.5$$

- conditional probability

$$P(F|M1) = 0.01$$
$$P(F|M2) = 0.02$$
$$P(F|M3) = 0.03$$

(F is the state of defective product)

- Then you can obtain the followings by using the Bayes theorem:

$$P(M1|F)$$
$$P(M2|F)$$
$$P(M3|F)$$

Exercise: Python code

```python
import numpy as np

m1frac = 0.2
m2frac = 0.3
m3frac = 0.5
m1def = 0.01
m2def = 0.02
m3def = 0.03
numprod = 10000.  # total number of products
# m1, m2, m3: 0- flawless, 1- defective
m1 = np.random.rand(int(numprod * m1frac)) < m1def
l1 = np.ones(int(numprod * m1frac), dtype=int)
m2 = np.random.rand(int(numprod * m2frac)) < m2def
l2 = np.ones(int(numprod * m2frac), dtype=int) * 2
m3 = np.random.rand(int(numprod * m3frac)) < m3def
l3 = np.ones(int(numprod * m3frac), dtype=int) * 3
```

Exercise: Python code

```
m = np.r_[m1, m2, m3]
l = np.r_[l1, l2, l3]
print(’defective rate: %g’ % (float(sum(m)) / len(m)))
numtrial = 10000
count = np.zeros(3)
numdef = 0
for i in range(numtrial):
    k = int(np.floor(np.random.rand() * len(m)))
    if m[k]:
        numdef += 1
        count[l[k] - 1] += 1
for i in range(3):
    print(’prob. drawn from M%d: %g’ % (i + 1, count[i] / numdef))
```

Exercise: Matlab code

```
m1frac=0.2; m2frac=0.3; m3frac=0.5;
m1def=0.01; m2def=0.02; m3def=0.03;
numprod=100000; % total number of products
% m1, m2, m3: 0- flawless, 1- defective
m1=rand(numprod*m1frac,1)<m1def;
l1=ones(numprod*m1frac,1);
m2=rand(numprod*m2frac,1)<m2def;
l2=ones(numprod*m2frac,1)*2;
m3=rand(numprod*m3frac,1)<m3def;
l3=ones(numprod*m3frac,1)*3;
m=[m1;m2;m3]; l=[l1;l2;l3];
fprintf(’defective rate: %g\n’,sum(m)/length(m));
```

Exercise: Matlab code

```
numtrial=1000;
count=zeros(3,1);
numdef=0;
for i=1:numtrial
  k=ceil(rand(1)*length(m));
  if m(k)
    numdef=numdef+1;
    count(l(k))=count(l(k))+1;
  end
end
for i=1:3
  fprintf(’prob. drawn from M%d: %g\n’, i, count(i)/numdef);
end
```

Exercise: Scilab code

```
m1frac=0.2; m2frac=0.3; m3frac=0.5;
m1def=0.01; m2def=0.02; m3def=0.03;
numprod=100000; // total number of products
// m?: 0 - flawless, 1 - defective
m1=rand(numprod*m1frac,1)<m1def;
l1=ones(numprod*m1frac,1);
m2=rand(numprod*m2frac,1)<m2def;
l2=ones(numprod*m2frac,1)*2;
m3=rand(numprod*m3frac,1)<m3def;
l3=ones(numprod*m3frac,1)*3;
m=[m1;m2;m3]; l=[l1;l2;l3];
printf(’defective rate: %g\n’, sum(m)/length(m));
```

Exercise: Scilab code

```
numtrial=1000;
count=zeros(3,1);
numdef=0;
for i=1:numtrial
  k=ceil(rand(1)*length(m));
  if m(k)
    numdef=numdef+1;
    count(l(k))=count(l(k))+1;
  end
end
for i=1:3
  printf(’prob. drawn from M%d: %g\n’, i, count(i)/numdef);
end
```

Exercise

- Plot posterior probabilities $P(\omega_1|x)$ and $P(\omega_2|x)$ for priors $P(\omega_1) = \frac{2}{3}$ and $P(\omega_2) = \frac{1}{3}$.

Exercise

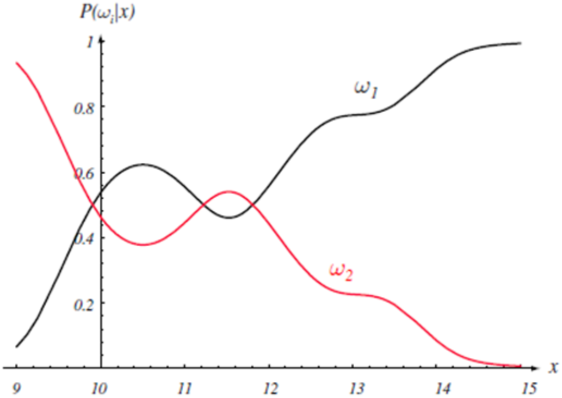


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Exercise

- Assume conditional distributions
  $p(x|\omega_1)$ to be Gaussian (=normal distribution) with mean 2 and standard deviation 1 and $p(x|\omega_2)$ to be also Gaussian with mean 4 and std 1 resp.
- Plot posterior probabilities $P(\omega_1|x)$ and $P(\omega_2|x)$ for priors $P(\omega_1) = 0.3$ and $P(\omega_2) = 0.7$.

bayes2.py

```
import numpy as np
import matplotlib.pyplot as plt

m1 = 2.
s1 = 1.  # mean and std for model 1
m2 = 4.
s2 = 1.  # mean and std for model 2
pr1 = 0.3  # prior for model 1
pr2 = 0.7  # prior for model 2

cond1 = lambda x : 1/np.sqrt(2. * np.pi * s1**2.) \
        * np.exp(-(x - m1)**2./(2. * s1**2.))
cond2 = lambda x : 1/np.sqrt(2. * np.pi * s2**2.) \
        * np.exp(-(x - m2)**2./(2. * s2**2.))
```

bayes2.py

```
x = np.linspace(0, 10)
plt.figure()
plt.plot(x, cond1(x), ’-’, label="cond1")
plt.plot(x, cond2(x), ’x-’, label="cond2")
plt.legend()

all = lambda x : cond1(x) * pr1 + cond2(x) * pr2
po1 = lambda x : cond1(x) * pr1 / all(x)
po2 = lambda x : cond2(x) * pr2 / all(x)

plt.figure()
plt.plot(x, po1(x), ’-’, label=’post1’)
plt.plot(x, po2(x), ’x-’, label=’post2’)
plt.legend()
plt.show()
```

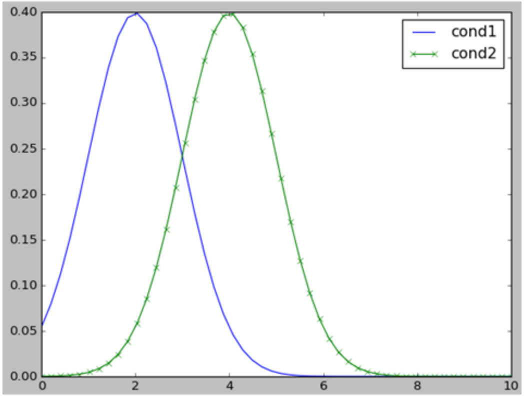
bayes2.m

```
m1=2; s1=1; % mean and std for model 1
m2=4; s2=1; % mean and std for model 2
pr1=0.3; % prior for model 1
pr2=0.7; % prior for model 2
cond1=@(x) 1/sqrt(2*pi*s1^2) * exp(-(x-m1)^2/(2*s1^2));
cond2=@(x) 1/sqrt(2*pi*s2^2) * exp(-(x-m2)^2/(2*s2^2));
x=0:0.1:10;
figure
plot(x,arrayfun(cond1,x),’-’,x,arrayfun(cond2,x),’x-’);
legend(’cond1’,’cond2’);
all=@(x) cond1(x)*pr1+cond2(x)*pr2;
po1=@(x) cond1(x)*pr1./all(x);
po2=@(x) cond2(x)*pr2./all(x);
figure
plot(x,arrayfun(po1,x),’-’,x,arrayfun(po2,x),’x-’);
legend(’post1’,’post2’);
```

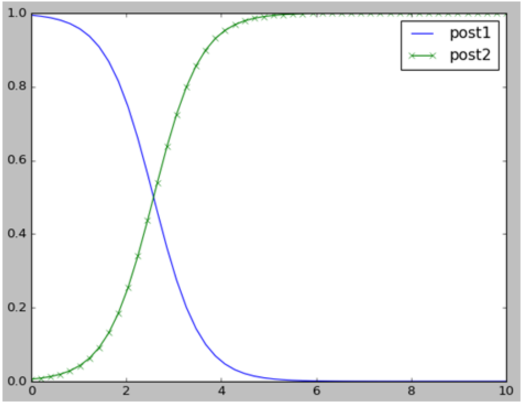
bayes2.sci

```
m1=2; s1=1; // mean and std for model 1
m2=4; s2=1; // mean and std for model 2
pr1=0.3; //prior for model 1
pr2=0.7; // prior for model 2
deff(’y=cond1(x)’, ’y=1/sqrt(2*%pi*s1^2) * exp(-(x-m1).^2/(2*s1^2))’);
deff(’y=cond2(x)’, ’y=1/sqrt(2*%pi*s2^2) * exp(-(x-m2).^2/(2*s2^2))’);
x=0:0.1:10;
figure
plot(x,cond1(x),’-’,x,cond2(x),’x-’);
legend(’cond1’,’cond2’);
deff(’y=all(x)’, ’y=cond1(x)*pr1+cond2(x)*pr2’);
deff(’y=po1(x)’, ’y=cond1(x)*pr1./all(x)’);
deff(’y=po2(x)’, ’y=cond2(x)*pr2./all(x)’);
figure
plot(x,po1(x),’-’,x,po2(x),’x-’);
legend(’post1’,’post2’);
```

Results



cond1
cond2
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10

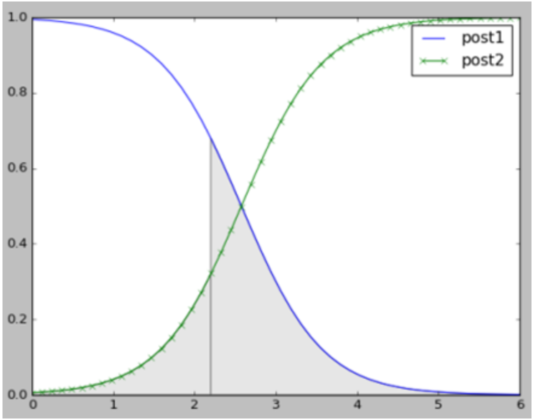


post1
post2
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10

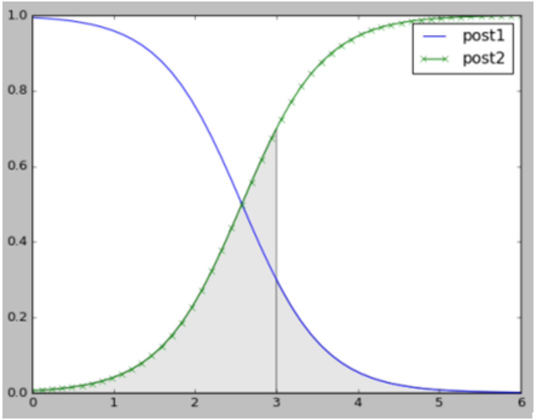
Bayesian decision theory

- $P(\text{error}|x) = P(\omega_1|x)$ if we decide ω_2
  $P(\text{error}|x) = P(\omega_2|x)$ if we decide ω_1
- Bayes decision rule
  Decide $\omega)1$ if $P(\omega_1|x) > P(\omega_2|x)$
  otherwise decide ω_2
- $P(\text{error}|x) = \min(P(\omega_1|x), P(\omega_2|x))$
- other form:
  Decide ω_1 if $p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2)$
  otherwise decide ω_2.
- Bayes error: $\mathrm{E}\{P(\text{error}|\mathbf{x})\}$ with Bayes decision rule

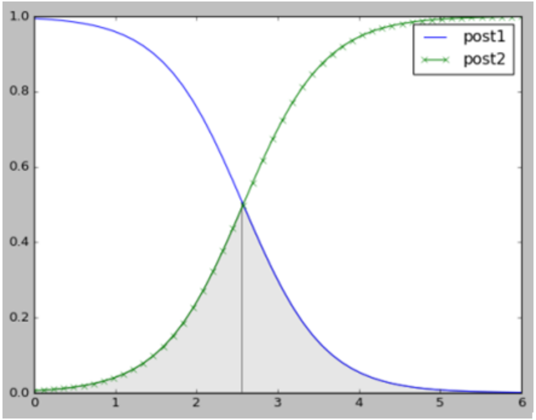
Bayesian decision theory



Bayesian decision theory



Bayesian decision theory



Risk

- The loss function $\lambda(\alpha_i|\omega_j)$ describes the loss incurred for taking action α_i when the state of nature is ω_j.
- Risk: $R(\alpha_i|x) = \sum_j \lambda(\alpha_i|\omega_j)P(\omega_j|x)$
- $R(\alpha_i|x) = \lambda(\alpha_i|\omega_1)P(\omega_1|x) + \lambda(\alpha_i|\omega_2)P(\omega_2|x)$
- zero-one loss
  $\lambda(\alpha_i|\omega_j) = \lambda_{ij} = 0$ if $i = j$;
  otherwise 1
- Then take action minimizing the loss

Risk

- $R(\alpha_1|x) = \lambda_{11}P(\omega_1|x) + \lambda_{12}P(\omega_2|x)$
- $R(\alpha_2|x) = \lambda_{21}P(\omega_1|x) + \lambda_{22}P(\omega_2|x)$
- ω_1: 食べられるキノコである (edible fungi)
- ω_2: 毒キノコである (poisonous fungi)
- α_1 : 食べる (eat!)
- α_2 : 食べない (do not eat)
- $\lambda_{12} \gg \lambda_{21}$

About exercise and assignment

- PC is expected to be used.
- either your own or provided by lab.
- Please install Anaconda, Matlab or Scilab, and play with sample codes
- Google Colaboratory can also be considered (actually very useful)