

# 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

## 本講義の目的

- 統計的パターン認識の理論を学ぶ
- 関連する理論は、メディア解析、特徴抽出、メディア変換、他、様々な技術に必要不可欠
- 本講義では最新のパターン認識や機械学習のツールについて学ぶことは目的としていない
- 一方、こうしたツールや技術がどのような仕組みで機能するのかについて注目する
- より理解を深めることを目的とし、実際のパターン識別器などに関する演習問題を宿題として解いてもらう

## 本講義について

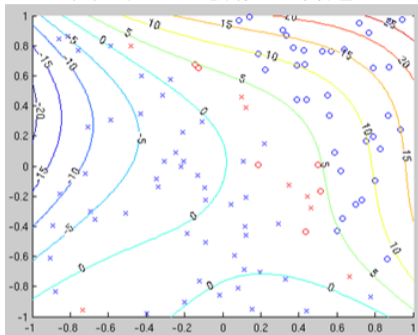
- 講義のページ: <https://research.nii.ac.jp/~sato/utpr/>
- 講義資料、サンプルコード、データなどを (できるだけ各講義前に) 掲示
- 講義は日本語で行う
- 講義資料は日本語版と英語版を Web にて提供
- 単位は最終レポートと宿題によって評価予定 (最終レポートの提出は必須、宿題は全7回 (予定) のうち3回の提出必須)
- 出席は取らない

## スケジュール (予定)

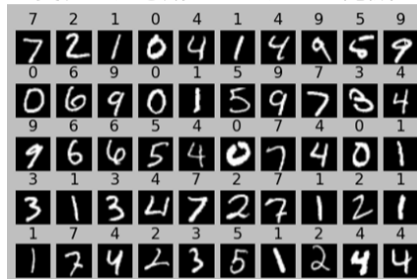
- 4/11 講義概要、ベイズ決定測、確率分布
- 4/18 確率変数、確率変数ベクトル、正規分布
- 4/25 確率分布パラメータ推定、識別関数
  - 5/2 ノンパラメトリック分布推定、パルゼン窓、k 近傍推定
  - 5/9 k 近傍識別器、識別誤り推定
- 5/16 ベイズ誤り推定、識別誤り推定、交差判定法、ブートストラップ法
- 5/23 線形識別器、パーセプトロン、最小自乗誤差 (MSE) 識別器、Widrow-Hoff 則
- 5/30 ニューラルネットワーク、深層学習
  - 6/6 サポートベクターマシン
- 6/13 直交展開、固有値展開
- 6/20 休講予定
- 6/27 クラスタリング、デンドログラム、凝集型 (agglomerative) クラスタリング、k-means
  - 7/4 グラフ、ノーマライズドカット、スペクトラルクラスタリング、ラプラシアンアイゲンマップ
- 7/11 予備日

## 宿題の例

人工データを用いた課題



実データを用いたパターン認識



## 参考書

- 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.

## パターン認識とは何か

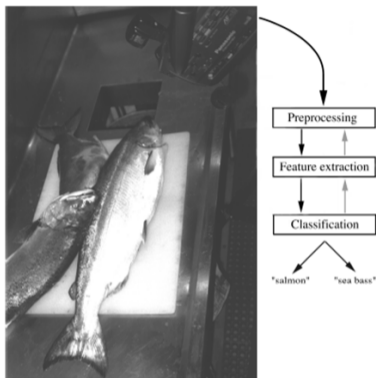
- 「パターン」を「認識」する
- 典型的には、実際の情報を観測し、情報のパターンのカテゴリに基づいてある行動をとる
- われわれを含み、生物にとって極めて重要であり、生物は極めて巧妙に実行できる

## パターン認識の実例

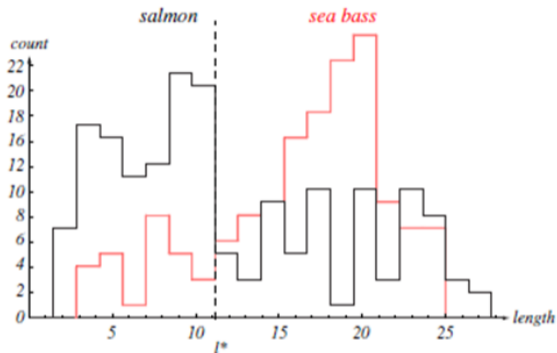
- 昆虫の知覚機能
- 顔検出・認識
- 音声認識
- 文書識別・認識
- ナビゲーション・自動運転
- シーン解析



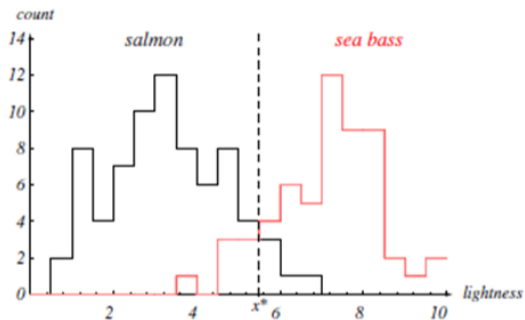
# 例



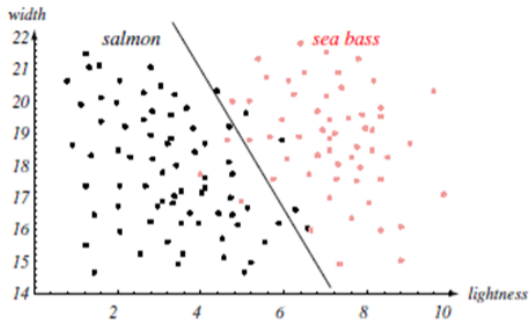
**FIGURE 1.1.** The objects to be classified are first sensed by a transducer (camera), whose signals are preprocessed. Next the features are extracted and finally the classification is emitted, here either "salmon" or "sea bass." Although the information flow is often chosen to be from the source to 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.



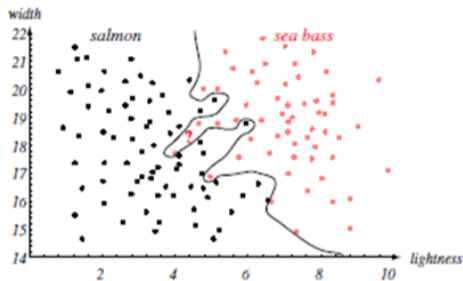
**FIGURE 1.2.** Histograms for the length feature for the two categories. No single threshold value of the length will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value marked  $l^*$  will lead to the smallest number of errors, on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



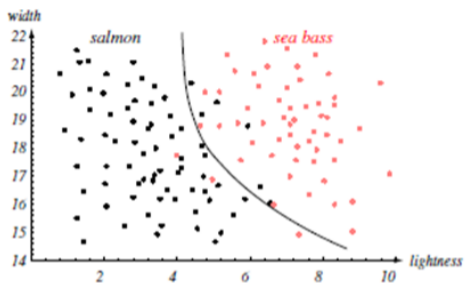
**FIGURE 1.3.** Histograms for the lightness feature for the two categories. No single threshold value  $x^*$  (decision boundary) will serve to unambiguously discriminate between the two categories; using lightness alone, we will have some errors. The value  $x^*$  marked will lead to the smallest number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



**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.



**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.

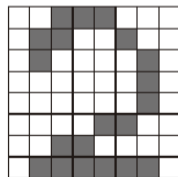
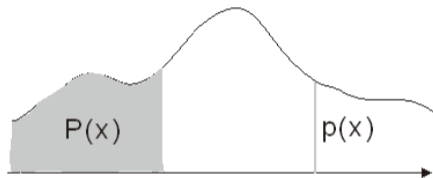


**FIGURE 1.6.** The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001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

# ベイズ決定則

確率:  $P(\mathbf{x})$

確率密度:  $p(x)$

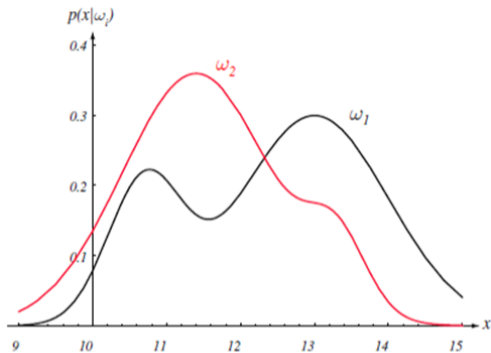


## ベイズ決定則

- $\omega$  : 状態を表す変数 (クラスと考えてよい)  
例)  $\omega = \omega_1$  スズキ (sea bass),  $\omega = \omega_2$  サケ (salmon)
- 事前確率 (prior, a priori probability): 各状態がどのくらい発生しやすいかに関するわれわれの事前知識  
例)  
 $P(\omega_1)$ : 次に観測する魚がスズキである確率  
 $P(\omega_2)$ : 次に観測する魚がサケである確率
- 条件付確率 (class-conditional probability) : (連続) 確率変数  $\mathbf{x}$  について、状態が  $\omega$  であるとしたときの  $\mathbf{x}$  の確率密度関数  
例)  $p(\mathbf{x}|\omega)$



## ベイズ決定則



**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  $\omega_i$ . 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.

## ベイズ決定則

- 事後確率 (posterior):  $\mathbf{x}$  についてある値  $x$  が観測された上での各状態の確率  
例)  $P(\omega|x)$
- 尤度 (likelihood): 確率変数がある値であるときの状態の尤度 (もっともらしさ)  
例)  $p(x|\omega)$
- ベイズの定理

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

## 演習

- ある工業製品を生産する三つの機械 M1, M2, M3 を考える
- 全生産数のうち 20%は M1 により、30%は M2 により、50%は M3 により生産されるとする
- さらに、それぞれ M1 は 1%, M2 は 2%, M3 は 3%の割合で不良品を生産するものとする
- 製品のうち一つを無作為に取り出したところ、不良品であったとする
- これはどの機械により生産されたものか、それぞれの確率を求めよ

## 考え方

- 事前確率

$$P(M1) = 0.2$$

$$P(M2) = 0.3$$

$$P(M3) = 0.5$$

- 条件付確率

$$P(F|M1) = 0.01$$

$$P(F|M2) = 0.02$$

$$P(F|M3) = 0.03$$

(ただし  $F$  は不良品の状態)

- ベイズの定理を用いて

$$P(M1|F)$$

$$P(M2|F)$$

$$P(M3|F)$$

を求めればよい

## 演習: 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
```

## 演習: Python

```
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))
```

## 演習: Matlab

```
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));
```

## 演習: Matlab

```
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
```



## 演習: Scilab

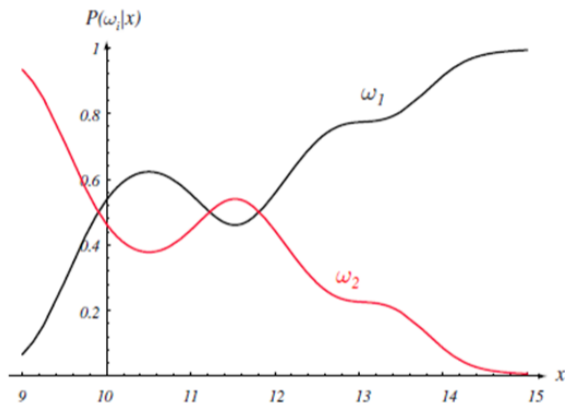
```
m1frac=0.2; m2frac=0.3; m3frac=0.5;
m1def=0.01; m2def=0.02; m3def=0.03;
numprod=100000; // total number of products
// m?: 0 - flawless, 1 - defective
m1=rand(numprod*m1frac,1)<m1def;
l1=ones(numprod*m1frac,1);
m2=rand(numprod*m2frac,1)<m2def;
l2=ones(numprod*m2frac,1)*2;
m3=rand(numprod*m3frac,1)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

## 演習: Scilab

```
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
```

## 演習

- 先のスズキ・サケの例で、事前確率を  $P(\omega_1) = \frac{2}{3}$  と  $P(\omega_2) = \frac{1}{3}$  として、事後確率  $P(\omega_1|x)$  と  $P(\omega_2|x)$  をプロットせよ



**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  $\omega_2$  is roughly 0.08, and that it is in  $\omega_1$  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.

## 演習

- 条件付確率を  
 $p(x|\omega_1)$  は平均 2, 標準偏差 1 の正規分布  
 $p(x|\omega_2)$  は平均 4, 標準偏差 1 の正規分布  
としよう
- 事前確率を  $P(\omega_1) = 0.3$  と  $P(\omega_2) = 0.7$  としたとき、事後確率  $P(\omega_1|x)$  と  $P(\omega_2|x)$  をプロットせよ

```
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.))
```

```
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()
```

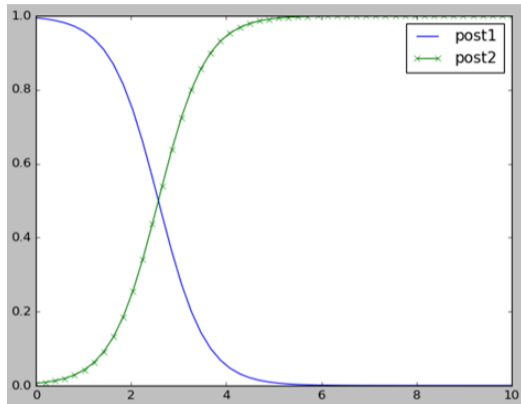
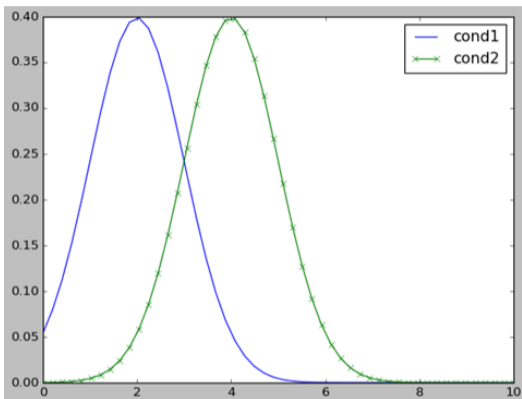
## bayes2.m

```
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');
```



```
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');
```

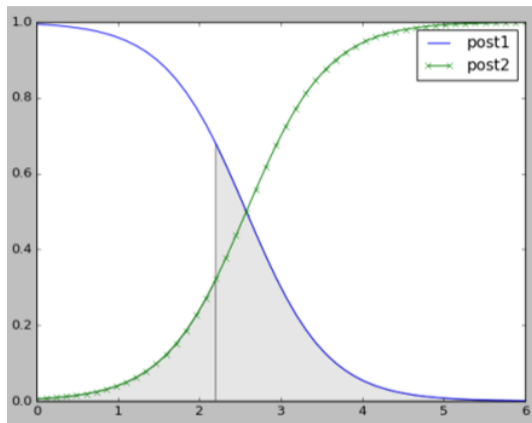
# 実行結果



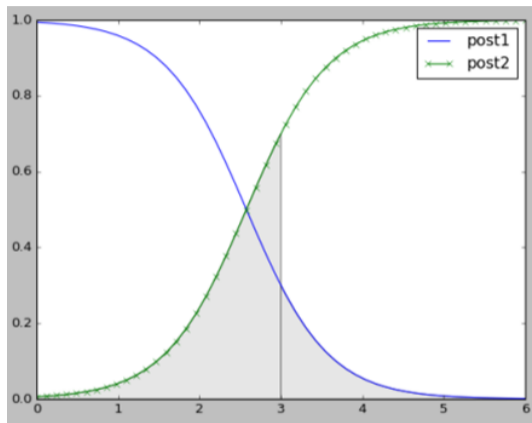
## ベイズ決定則

- $P(\text{error}|x) = P(\omega_1|x)$  ( $\omega_2$  と決定した時)  
 $P(\text{error}|x) = P(\omega_2|x)$  ( $\omega_1$  と決定した時)
- ベイズ決定則では、  
 $P(\omega_1|x) > P(\omega_2|x)$  の時  $\omega_1$ 、それ以外の時  $\omega_2$
- $P(\text{error}|x) = \min(P(\omega_1|x), P(\omega_2|x))$
- ベイズの式で展開すると  
 $p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2)$  の時  $\omega_1$ 、それ以外の時  $\omega_2$
- ベイズ誤り率: ベイズ決定則を使った時の  $E\{P(\text{error}|x)\}$

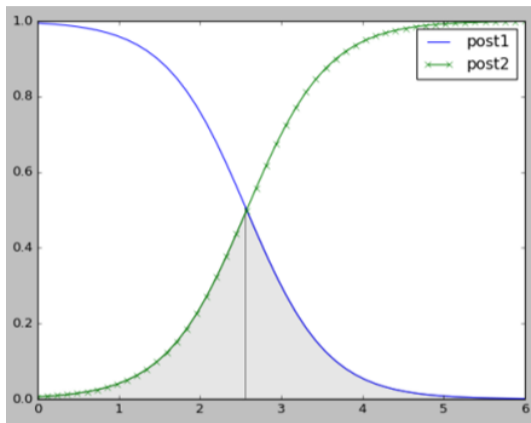
## ベイズ決定則



## ベイズ決定則



## ベイズ決定則



## リスク

- 損失関数  $\lambda(\alpha_i|\omega_j)$  として、状態が  $\omega_j$  の時に  $\alpha_i$  を実行したときの損失を考える
- リスク:  $R(\alpha_i|x) = \sum_j \lambda(\alpha_i|\omega_j)P(\omega_j|x)$
- $R(\alpha_i|x) = \lambda(\alpha_i|\omega_1)P(\omega_1|x) + \lambda(\alpha_i|\omega_2)P(\omega_2|x)$
- 0-1 損失の例  
 $\lambda(\alpha_i|\omega_j) = \lambda_{ij} = 0$  ( $i = j$  の時), それ以外 1
- 損失を最小にするように行動する

## リスク

- $R(\alpha_1|x) = \lambda_{11}P(\omega_1|x) + \lambda_{12}P(\omega_2|x)$
- $R(\alpha_2|x) = \lambda_{21}P(\omega_1|x) + \lambda_{22}P(\omega_2|x)$
- $\omega_1$ : 食べられるキノコである
- $\omega_2$ : 毒キノコである
- $\alpha_1$ : 食べる
- $\alpha_2$ : 食べない
- $\lambda_{12} \gg \lambda_{21}$



## 演習と宿題について

- PCを持っていることを想定している
- 私物、もしくは研究室からの貸与品
- Anaconda (Python) あるいは Matlab もしくは Scilab をインストールして、サンプルコードを動かしてみてください
- Google Colaboratory の利用も便利