

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 25, 2023

本講義について

- 講義のページ: <https://research.nii.ac.jp/~sato/utpr/>
- 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

前回の復習

- 正規分布 (ガウス分布、ガウシアン)
- 正規分布の十分統計量: 平均ベクトルと共分散行列
- 標本推定: 標本平均ベクトルと標本共分散行列

本日の予定

- パラメータで記述できる確率分布の推定
- ベイズ決定則に基づく識別器の設計

識別関数: 一般分布、複数カテゴリの場合

与えられた観測データを c クラスの一つに識別する問題を考えよう
以下の識別関数を使って、一般化して考えよう

$$g_i(x), i = 1, \dots, c.$$

識別器は、特徴ベクトル x が以下を満たす場合にクラス ω_i に識別する

$$g_i(x) > g_j(x) \text{ for all } j \neq i.$$

識別関数: 一般分布、複数カテゴリの場合

ベイズ決定則 (再掲)

Decide ω_i if $P(\omega_i|x) > P(\omega_j|x)$ for all $j \neq i$

以下の識別関数を使えばよい

$$g_i(x) = P(\omega_i|x).$$

より一般的に損失関数 (Risk) も考えると

$$\text{Risk: } R(\alpha_i|x) = \sum_j \lambda_{ij} P(\omega_j|x)$$

識別器は以下のように実現できる

Decide ω_i if $R(\alpha_i|x) < R(\alpha_j|x)$ for all $j \neq i$

識別関数は以下の通り

$$g_i(x) = -R(\alpha_i|x).$$

識別関数: 一般分布、複数カテゴリの場合

ベイズ決定則 (誤り率最小、ベイズ誤り率) の場合

$$g_i(x) = P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{\sum_{j=1}^c p(x|\omega_j)P(\omega_j)}$$

$$g_i(x) = p(x|\omega_i)P(\omega_i)$$

$$g_i(x) = \log p(x|\omega_i) + \log P(\omega_i)$$

識別関数: 一般分布、2カテゴリの場合

2カテゴリのみの場合、 c 個の識別関数の代わりに、単一の二分類器を使えばよい

$$g(x) \stackrel{\text{def}}{=} g_1(x) - g_2(x)$$

この場合の識別則は: Decide ω_1 if $g(x) > 0$ otherwise ω_2 .

他の例:

$$g(x) = P(\omega_1|x) - P(\omega_2|x)$$

$$g(x) = \log \frac{p(x|\omega_1)}{p(x|\omega_2)} + \log \frac{P(\omega_1)}{P(\omega_2)}.$$

正規分布に対する識別関数

正規分布

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

識別関数

$$g_i(x) = \log p(x|\omega_i) + \log P(\omega_i).$$

従って、正規分布に対する識別関数は

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i) - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_i| + \log P(\omega_i).$$

ケース 1: $\Sigma_i = \sigma^2 I$

もっとも単純な場合として、特徴量 (ベクトルの要素の間) が相互に確率的に独立で、かつ各特徴が同じ分散 σ^2 を持つ場合を考える

特に $|\Sigma_i| = \sigma^{2d}$ と $\Sigma_i^{-1} = (1/\sigma^2)I$ となるので
識別関数は以下の通り:

$$\begin{aligned} g_i(x) &= -\frac{\|x - \mu_i\|^2}{2\sigma^2} + \log P(\omega_i) \\ &= -\frac{1}{2\sigma^2} [x^T x - 2\mu_i^T x + \mu_i^T \mu_i] + \log P(\omega_i). \end{aligned}$$

共通項を省略すると、以下のような線形識別関数が得られる

$$g_i(x) = w_i^T x + w_{i0}$$

ただし

$$w_i = \frac{1}{\sigma^2} \mu_i \text{ and } w_{i0} = -\frac{1}{2\sigma^2} \mu_i^T \mu_i + \log P(\omega_i).$$

ケース 1: $\Sigma_i = \sigma^2 I$

2 カテゴリーの場合

$$g(x) = w^T(x - x_0)$$

$$w = \mu_1 - \mu_2$$

$$x_0 = \frac{1}{2}(\mu_1 + \mu_2) - \frac{\sigma^2}{\|\mu_1 - \mu_2\|^2} \log \frac{P(\omega_1)}{P(\omega_2)} (\mu_1 - \mu_2).$$

識別面は二つの平均ベクトル μ_1 と μ_2 をつなぐ線分に直交する超平面となる
もし事前確率が同じ場合 ($P(\omega_1) = P(\omega_2)$) 上記線分の垂直二等分超平面となる
この場合平均ベクトルへの最近傍識別器となる

ケース 1: $\Sigma_i = \sigma^2 I$

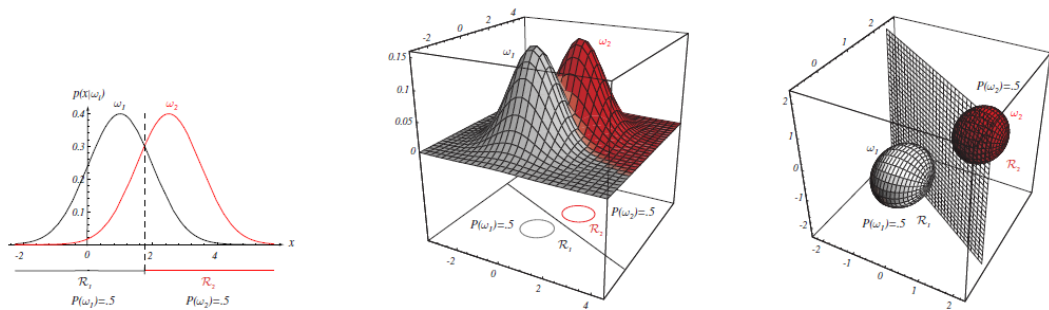


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d - 1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates \mathcal{R}_1 from \mathcal{R}_2 . From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

ケース 1: $\Sigma_i = \sigma^2 I$

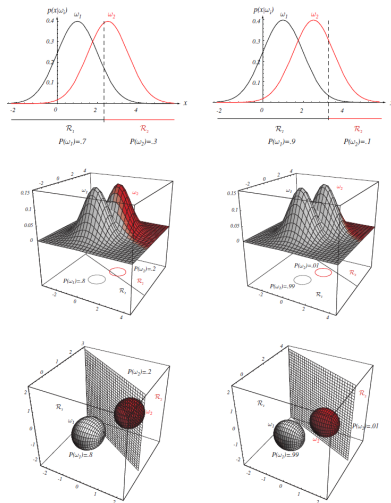


FIGURE 2.11. As the priors are changed, the decision boundary shifts; for sufficiently disparate priors the boundary will not lie between the means of these one-, two- and three-dimensional spherical Gaussian distributions. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley &

ケース2: $\Sigma_i = \Sigma$

別の単純な場合として、共分散行列が同一の場合を考える
この場合、識別関数は以下の通り簡略化される

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma^{-1}(x - \mu_i) + \log P(\omega_i).$$

二次形式部分 $(x - \mu_i)^T \Sigma^{-1}(x - \mu_i)$ を展開すると、共通項を省略することにより以下の通りさらに簡略化できる

$$g_i(x) = w_i^T x + w_{i0}$$

ただし

$$w_i = \Sigma^{-1} \mu_i \text{ and } w_{i0} = -\frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \log P(\omega_i).$$

ケース 2: $\Sigma_i = \Sigma$

2 カテゴリの場合

$$g(x) = w^T(x - x_0)$$

$$w = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$x_0 = \frac{1}{2}(\mu_1 + \mu_2) - \frac{1}{(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)} \log \frac{P(\omega_1)}{P(\omega_2)} (\mu_1 - \mu_2).$$

ケース2: $\Sigma_i = \Sigma$

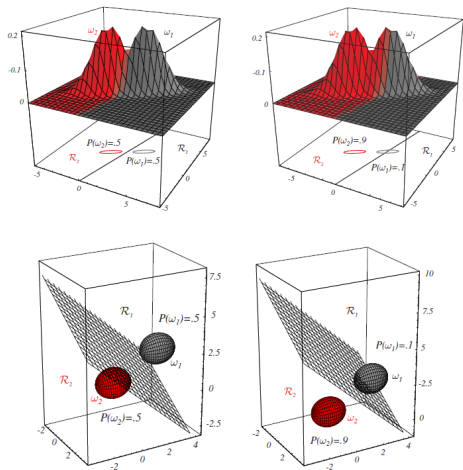


FIGURE 2.12. Probability densities (indicated by the surfaces in two dimensions and ellipsoidal surfaces in three dimensions) and decision regions for equal but asymmetric Gaussian distributions. The decision hyperplanes need not be perpendicular to the line connecting the means. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

ケース 3: $\Sigma_i = \text{arbitrary}$

$(d/2) \log 2\pi$ が省略できるのみ
よって識別関数は二次関数となる

$$g_i(x) = x^T W_i x + w_i^T x + w_{i0}$$

$$W_i = -\frac{1}{2} \Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1} \mu_i$$

$$w_{i0} = \mu_i^T W_i \mu_i - \frac{1}{2} \log |\Sigma_i| + \log P(\omega_i).$$

ケース 3: $\Sigma_i = \text{arbitrary}$

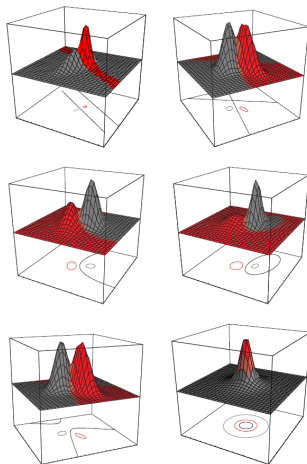


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

ケース 3: $\Sigma_i = \text{arbitrary}$

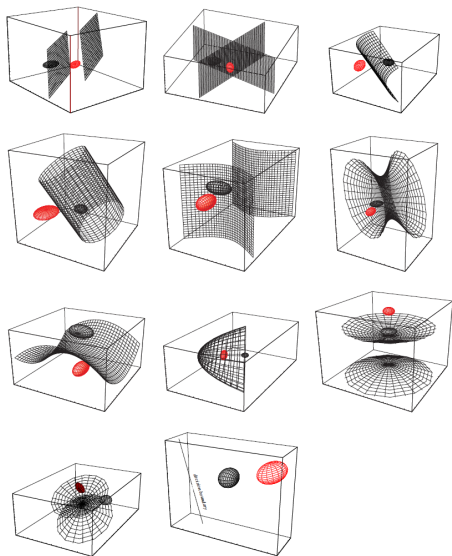


FIGURE 9.15 A list of surfaces that can be realized in \mathbb{R}^3 as the level set of a function.

宿題 1

2つの正規分布にそれぞれ従う2次元点群を2セット用意せよ

Case 1 $\Sigma_i = \sigma^2 I$

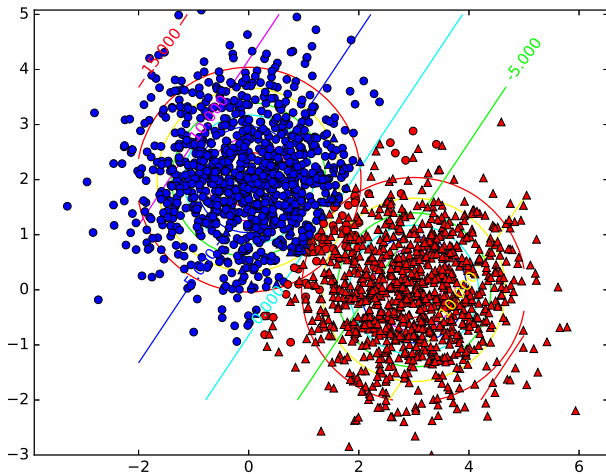
Case 2 $\Sigma_i = \Sigma$

Case 3 $\Sigma_i = \text{arbitrary}$

正規分布のパラメータを既知とし、識別関数を設計し、これらの点群を分類せよ
点群をプロットし(クラスを異なる色とし、正解・不正解の識別を異なる形状の点とするなど)、正規分布の等高線、さらに識別境界線を表示せよ

正規分布のパラメータをいろいろ変えて、識別境界線(もちろん直線とは限らない)の変化を見よ

宿題 1 (cov_diag.py)



宿題 1 (cov_diag.py)

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

def gausscontour(c, m, xx, yy):
    xt = xx - m[0, 0]
    yt = yy - m[1, 0]
    ic = np.linalg.inv(c)
    p = 1. / (2. * np.pi * np.sqrt(np.linalg.det(c))) \
        * np.exp(-1. / 2. * \
            (ic[0, 0] * xt * xt \
             + (ic[0, 1] + ic[1, 0]) * xt * yt \
             + ic[1, 1] * yt * yt))
    return p
```

宿題 1 (cov_diag.py)

```
d = 2
n = 1000
m1 = np.array([0., 2.])[:, np.newaxis]
m2 = np.array([3., 0.])[:, np.newaxis]
p1 = 0.3
p2 = 1 - p1
cov1 = np.eye(2)
cov2 = cov1
x1 = np.random.randn(d, n) + m1.dot(np.ones([1, n]))
x2 = np.random.randn(d, n) + m2.dot(np.ones([1, n]))
w = m1 - m2
x0 = 1. / 2. * (m1 + m2) - 1. / np.linalg.norm(m1 - \
          m2)**2. * np.log(p1 / p2) * (m1 - m2)
l1 = (w.T.dot(x1 - x0) > 0)[-1]
l2 = (w.T.dot(x2 - x0) > 0)[-1]
```

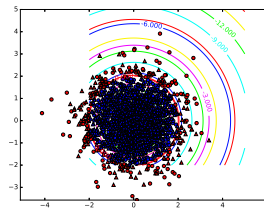
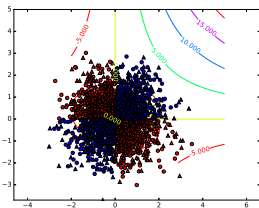
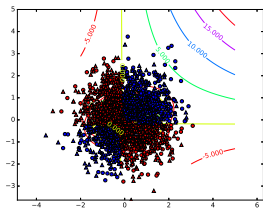
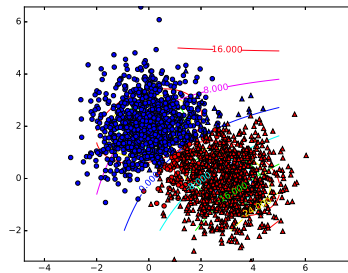
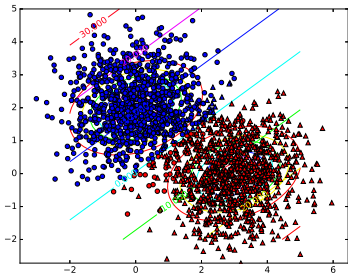

宿題 1 (cov_diag.py)

```
[xx, yy] = np.meshgrid(np.linspace(-2, 5), np.linspace(-2, 5))
plt.figure()
plt.axis('equal')
p1 = gausscontour(cov1, m1, xx, yy)
plt.contour(xx, yy, p1, cmap='hsv')
p2 = gausscontour(cov2, m2, xx, yy)
plt.contour(xx, yy, p2, cmap='hsv')
# correct x1
plt.plot(x1[0, np.where(l1)], x1[1, np.where(l1)], 'bo')
# wrong x1
plt.plot(x1[0, np.where(~l1)], x1[1, np.where(~l1)], 'ro')
# correct x2
plt.plot(x2[0, np.where(1 - l2)], x2[1, np.where(1 - l2)], 'r^')
# wrong x2
plt.plot(x2[0, np.where(l2)], x2[1, np.where(l2)], 'b^')
```

宿題 1 (cov_diag.py)

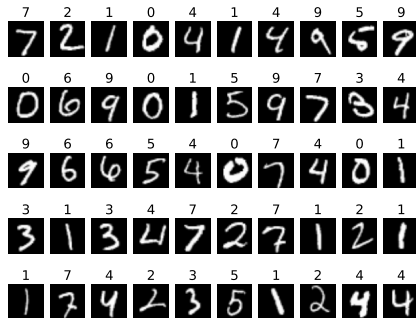
```
xxyy = np.c_[np.reshape(xx, -1), np.reshape(yy, -1)].T
pp = w.T.dot(xxyy - x0 * np.ones([1, xxyy.shape[1]]))
pp = np.reshape(pp, xx.shape)
cs = plt.contour(xx, yy, pp, cmap='hsv')
plt.clabel(cs)
# plt.savefig('cov_diag.eps')
plt.show()
```

宿題 1



手書き数字認識データ MNIST

MNIST は正解データ付きの手書き数字画像データである
各画像は 28×28 画素であり、各画素には輝度値が対応している
学習用データは 60,000 画像、テストデータは 10,000 画像提供されている



MNIST データ利用準備

MNIST データを以下からダウンロードせよ <http://yann.lecun.com/exdb/mnist/>.

train-images-idx3-ubyte.gz	training set images	(9912422 bytes)
train-labels-idx1-ubyte.gz	training set labels	(28881 bytes)
t10k-images-idx3-ubyte.gz	test set images	(1648877 bytes)
t10k-labels-idx1-ubyte.gz	test set labels	(4542 bytes)

MNIST データを読み込むプログラム `mnread.py` を提供している
`mnread.py` を利用するため、プログラム作成の作業ディレクトリに 'mnist' というサブディレクトリを作り、MNIST データを格納せよ

`mnread.readim` : read MNIST images
`mnread.readlabel` : read MNIST labels

宿題 2 (MNIST)

識別器を MNIST 学習データで作成し、テストデータを識別してみよ
各画像は 784 ($= 28 \times 28$) 次元ベクトルとして扱えばよい
各数字は正規分布に従うと仮定せよ

Case 1 $\Sigma_i = \sigma^2 I$

Case 2 $\Sigma_i = \Sigma$

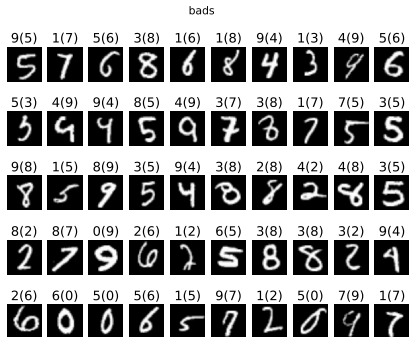
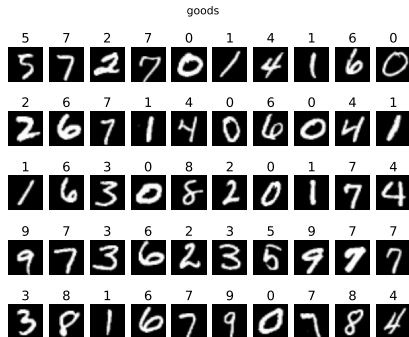
Case 3 $\Sigma_i = \text{arbitrary}$

正規分布のパラメータを MNIST 学習データで推定せよ
推定したパラメータにより識別関数を決定し、テストデータを識別せよ

ヒント: 正規化 (regularization) や疑似逆行列 (pseudo inverse) の使用を検討してもよい

宿題 2 (mntest.py)

同一の等方性正規分布 ($\Sigma_i = \sigma^2 I$) を仮定した場合の認識性能は 82%



宿題 2 (mntest.py)

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

def train(label, data):
    data = np.reshape(data, [data.shape[0], -1])
    lset = set(label)
    model = np.empty((len(lset), data.shape[1]), dtype=float)
    for x in lset:
        model[x, :] = np.mean(data[np.where(label == x), :], axis=1)
    return model

def classify(data, model):
    data = np.reshape(data, [data.shape[0], -1])
    label = np.empty(data.shape[0], dtype=int)
    for i in range(data.shape[0]):
```


宿題 2 (mntest.py)

```
trlabel = mnread.readlabel(mnread.trlabelfz)
trdata = mnread.readim(mnread.trdatafz)
tstlabel = mnread.readlabel(mnread.tstlabelfz)
tstdata = mnread.readim(mnread.tstdatafz)

model = train(trlabel, trdata)
estlabel = classify(tstdata, model)
print('accuracy: %g' % (float(sum(estlabel == tstlabel)) / len(tstlabel)))
```

宿題 2 (mntest.py)

```
plt.figure()
plt.suptitle('goods')
goods = np.random.permutation(np.where(estlabel == tstlabel)[-1])[range(50)]
for i, good in enumerate(goods):
    plt.subplot(5, 10, i + 1)
    plt.axis('off')
    plt.imshow(tstdata[good, :, :], cmap='gray')
    plt.title(estlabel[good])
plt.savefig('good.eps')
plt.figure()
plt.suptitle('bads')
bads = np.random.permutation(np.where(~(estlabel == tstlabel))[-1])[range(50)]
for i, bad in enumerate(bads):
    plt.subplot(5, 10, i + 1)
    plt.axis('off')
    plt.imshow(tstdata[bad, :, :], cmap='gray')
    plt.title('%s(%s)' % (estlabel[bad], tstlabel[bad]))
plt.savefig('bad.eps')
```

宿題 2 のヒント

Case 1 `mntest.py` を参照してわかる通り、学習時には各数字の平均ベクトルを求めればよく、識別時には入力ベクトルに最も近い平均ベクトルを見つければよい

Case 2 学習時には、case 1 と同様に、まず各数字の平均ベクトルを求める
次いで各数字の学習データの平均からの差分ベクトルを求め、全数字共通の共分散行列を求める

Case 3 むしろ単純であり、各数字ごとの平均ベクトルと共分散行列を求める
95% の精度が達成できている

宿題 2 のヒント

共分散行列の逆行列を求める際、特異行列であるため問題が生じる場合がある
その一つの理由は、学習画像の中のある部分について全学習画像における輝度値がすべてゼロとなる領域があるため
この問題は一般化逆行列 (疑似逆行列、ムーア・ペンローズの疑似逆行列などとも呼ばれる) を用いて回避することができる

Matlab では `pinv`, Python では `numpy.linalg.pinv` として利用可能

Inverse:

$$\Sigma \Phi = \Phi \Lambda \quad (\Sigma \text{ の固有値展開})$$

$$\Sigma = \Phi \Lambda \Phi^T$$

$$= \sum \lambda_i \phi_i \phi_i^T$$

$$\Sigma^{-1} = \sum \frac{1}{\lambda_i} \phi_i \phi_i^T$$

一般化逆行列はすべてのゼロ (に極めて近い) 固有値を無視して逆行列を求める
これは等価的に画像すべてで輝度値ゼロの領域を無視することになる (これに追加して線形従属の輝度値についても対応)

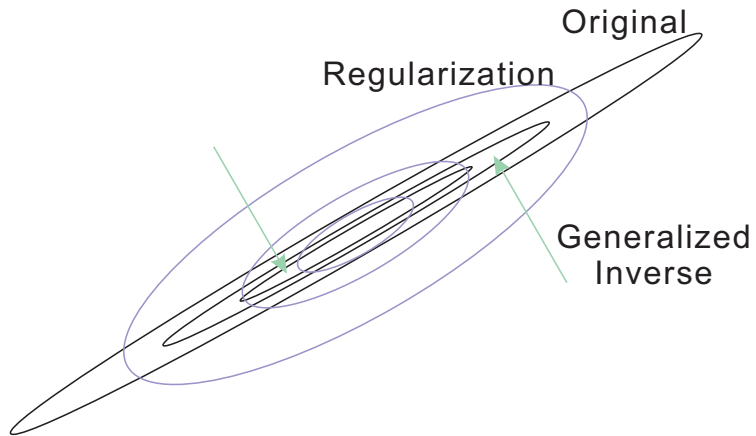
宿題 2 のヒント

正則化 (regularization) を考えてもよい

正則化: Σ が特異行列の場合、 Σ^{-1} を計算する代わりにある定数 α を用いて $(\Sigma + \alpha I)^{-1}$ を計算する

$$\begin{aligned}(\Sigma + \alpha I)^{-1} &= (\Phi \Lambda \Phi^T + \alpha I)^{-1} \\ &= [\Phi \Lambda \Phi^T + \Phi(\alpha I)\Phi^T]^{-1} \\ &= [\Phi(\Lambda + \alpha I)\Phi^T]^{-1} \\ &= [\sum (\lambda_i + \alpha) \phi_i \phi_i^T]^{-1} \\ &= \sum \frac{1}{\lambda_i + \alpha} \phi_i \phi_i^T\end{aligned}$$

宿題 2 のヒント



宿題 2 のヒント

共分散行列 Σ_i が特異行列の場合、項 $\log |\Sigma_i|$ の扱いにも注意が必要となる
なぜこの項が必要だったのか、再考してみよ
おのずと対応法について見えてくるはず