使ってみよう部分空間法
― 部分空間法体験実習 ―

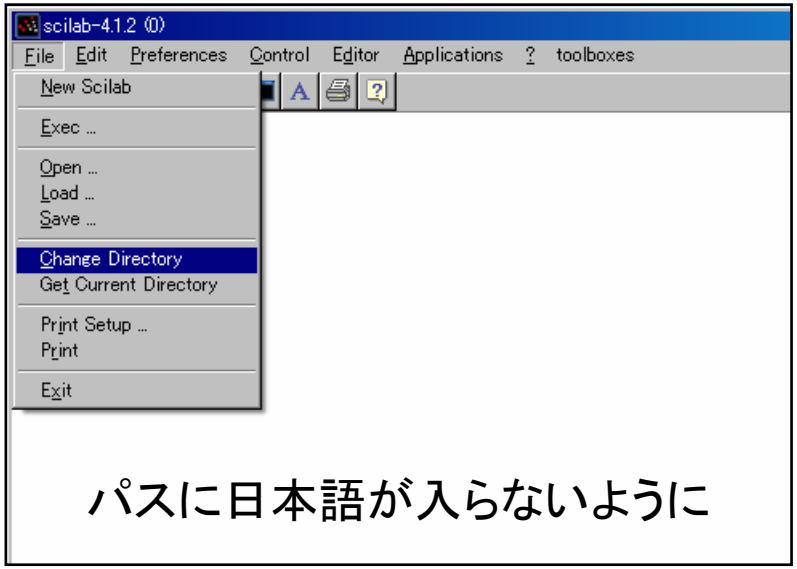
部分空間法研究会

2008年7月28日

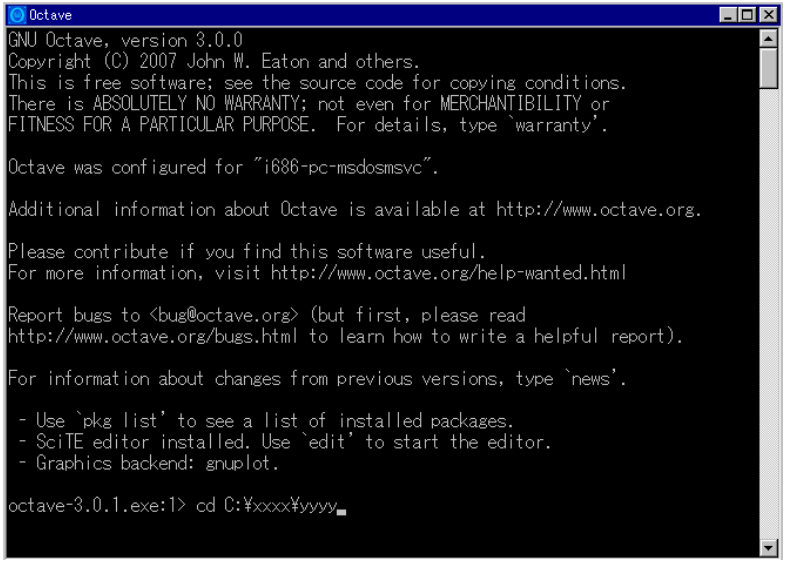
堀田政二，天野敏之，玉木徹

はじめに

準備としてこの演習で使用するScilab, Octave, MATLAB
を起動してダウンロードしたプログラムのあるフォルダや
ディレクトリに移動しておいてください



パスに日本語が入らないように



★ PCを持っていない等，作業できない方は
スライドを見ているだけでも大丈夫（なはず）

実習の目的

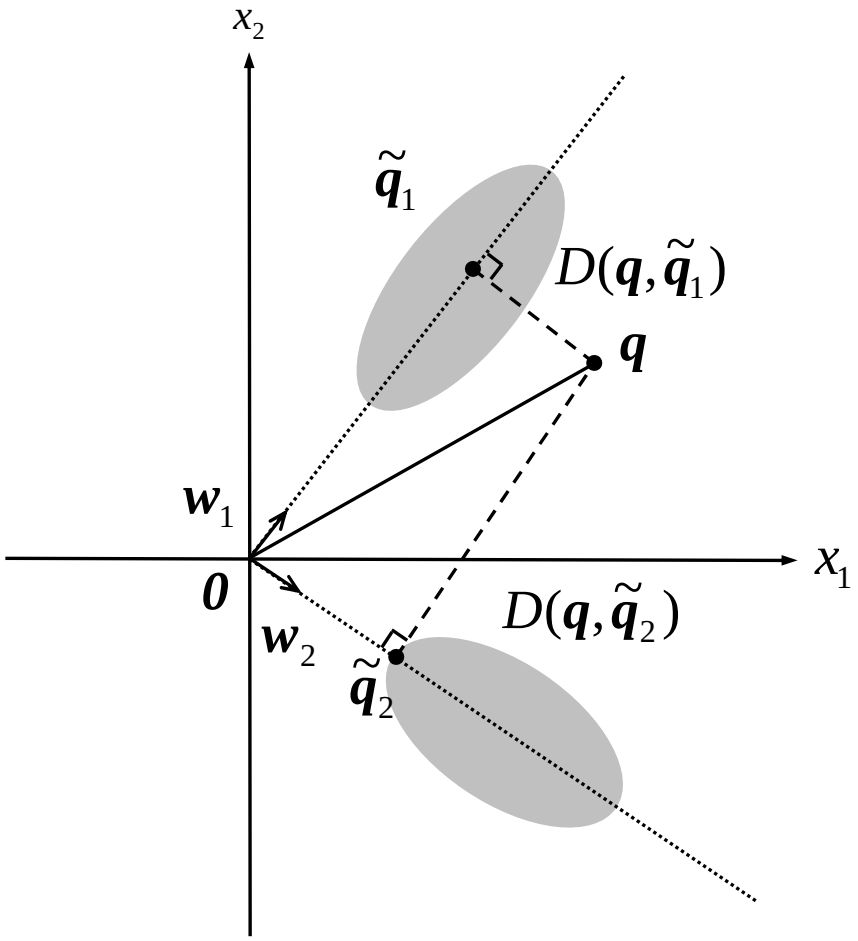
実際に部分空間法のプログラムを動かしてみて
使いやすさやプログラミングの容易さを実感する

実習の内容

- Step1 自己相関行列の固有値と固有ベクトルを見よう
- Step2 画像パターンの直交展開してみよう
- Step3 部分空間法で手書き数字を認識してみよう

部分空間法のCLAFICとは

クラスらしさを部分空間で表現し
識別にも直接部分空間を利用



テストパターンを良く近似できる
部分空間の属すクラスを出力

識別規則

$$\omega = \arg \max_{j=1,\cdots,C} \|\mathbf{W}_j^\top \mathbf{q}\|^2$$

部分空間は原点共通

クラス毎の部分空間の求め方

$$\max_{\mathbf{w}} \sum_{i \in C_j} (\mathbf{w}^\top \mathbf{x}_i)^2 \qquad \text{s.t. } \|\mathbf{w}\|^2 = 1$$

ラグランジュ関数は

$$L(\mathbf{w}, \lambda) = \mathbf{w}^\top (\sum_{i \in C_j} \mathbf{x}_i \mathbf{x}_i^\top) \mathbf{w} - \lambda (\mathbf{w}^\top \mathbf{w} - 1)$$

$\partial L / \partial \mathbf{w} = \mathbf{0}$ として実際に解くと

$$(\textstyle\sum_{i \in C_j} \mathbf{x}_i \mathbf{x}_i^\top) \mathbf{w} = \lambda \mathbf{w}$$

各クラスの正規直交基底は自己相関行列の固有ベクトル

部分空間法の識別

クラス j の自己相関行列の固有値の大きい
上位 r 本の固有ベクトルを並べた行列: $\mathbf{W}_j$

$$\omega = \max_{j=1,\cdots,C} \|\mathbf{W}_j^\top \mathbf{q}\|^2$$

パラメータ r (一つだけ！) は累積寄与率や
パラメータ推定法によって決定（実験例は後述）

メモリ容量や計算時間を削減したい場合にはrで
調節可能な点が便利（SVMでは無理）

線型部分空間を使うと
何がうれしいのか

辞書を少ないメモリ容量で保存できる

高速にパターンを多クラスに分類できる

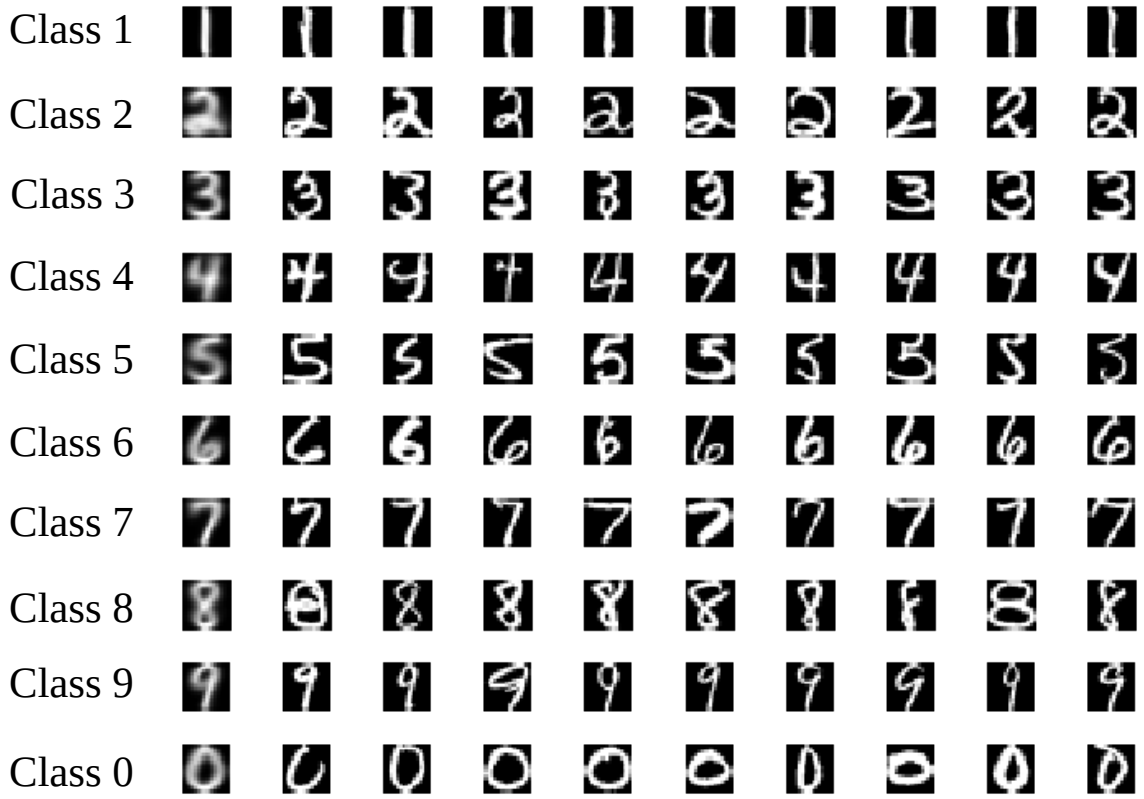
簡単に実装できて高い識別率を達成できる

これらを実際に体感してみましょう

演習で用いる手書き数字データUSPSの詳細

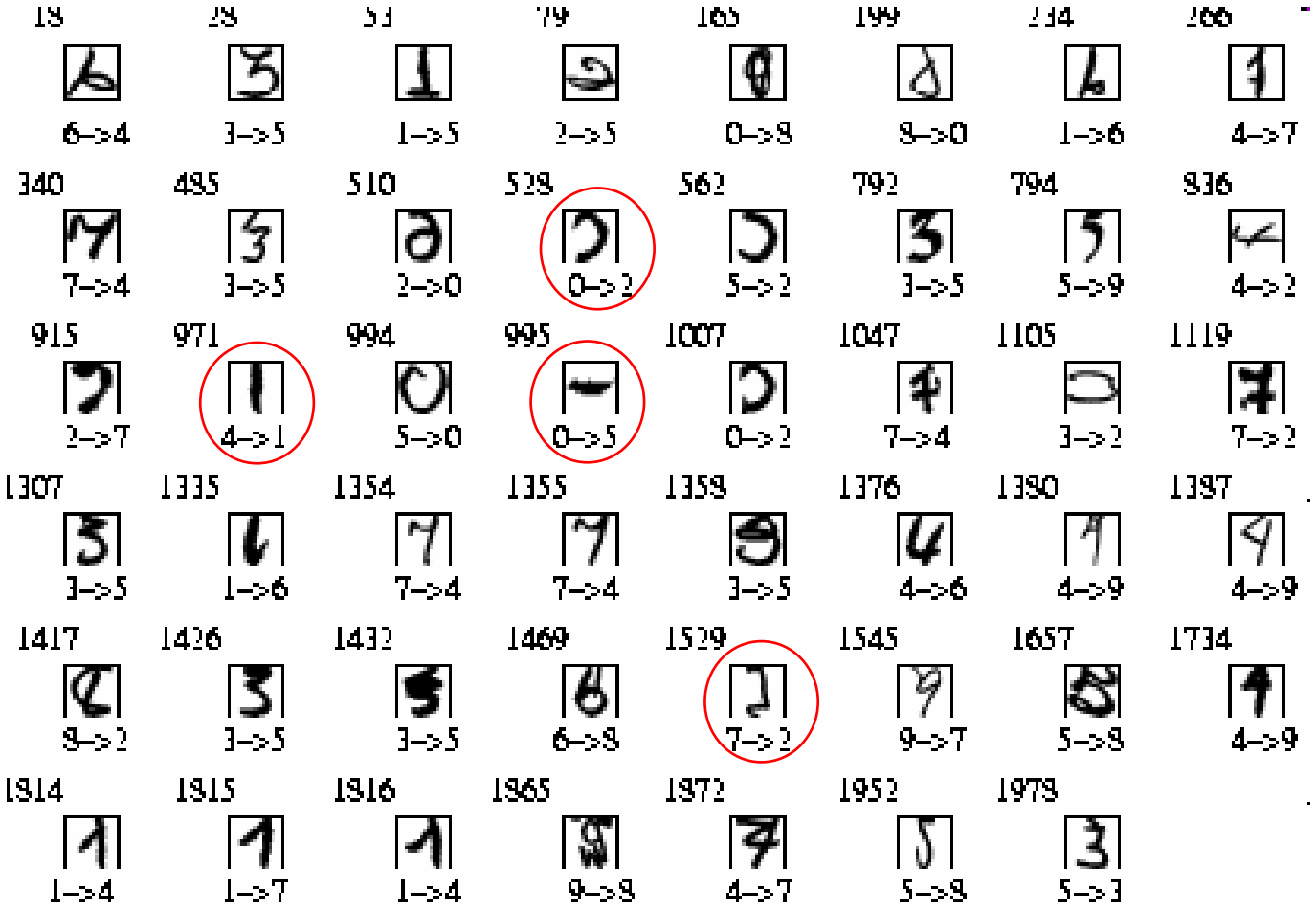
訓練パターン数 7291，未知パターン数 2007

クラスは0から9までの10クラス 16×16pixelの256階調のモノクロ画像



訓練データの一例：左端はクラスの重心

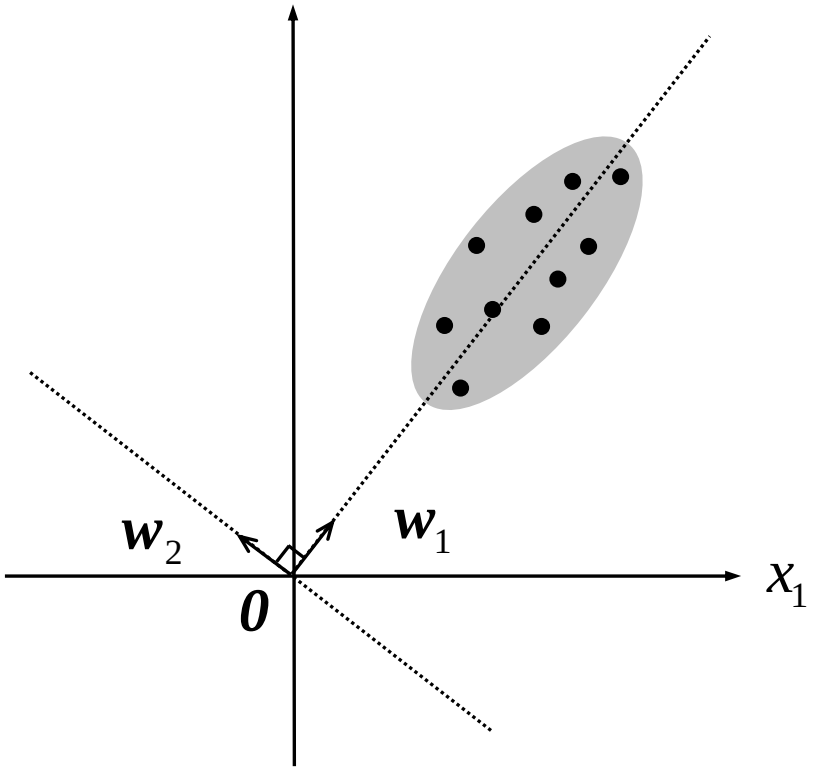
テストデータの例



ミスラベルや意味不明のパターンが含まれている

主成分分析でデータの分布を観察する

データ分布全体を最も良く近似する部分空間を求める



データの分布を楕円で近似して長軸と短軸を求める
上の図の場合，長軸に正射影すれば元のデータ分布を良く近似できる

主成分分析

$$\max_{\mathbf{w}} \ \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w} \qquad \text{s.t.} \ \|\mathbf{w}\|^2 = 1$$

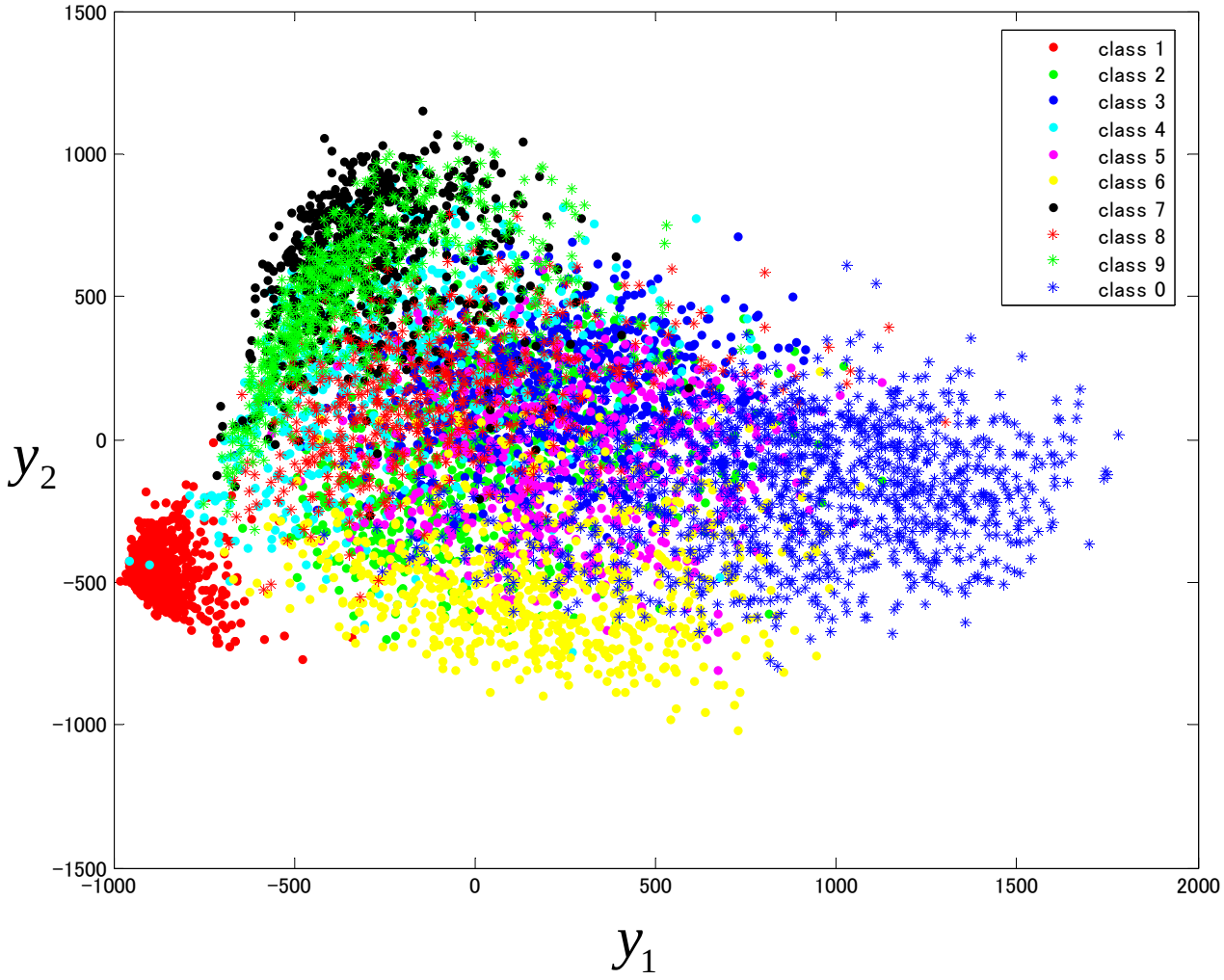
$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top$$ (標本) 共分散行列

ラグランジュ乗数法から解は $\boldsymbol{\Sigma}\mathbf{w} = \lambda \mathbf{w}$

共分散行列の大きい固有値(分散)に
対応する上位r本の固有ベクトルでデータを写像

$$\mathbf{y} = \mathbf{W}^\top (\mathbf{x} - \mathbf{m})$$

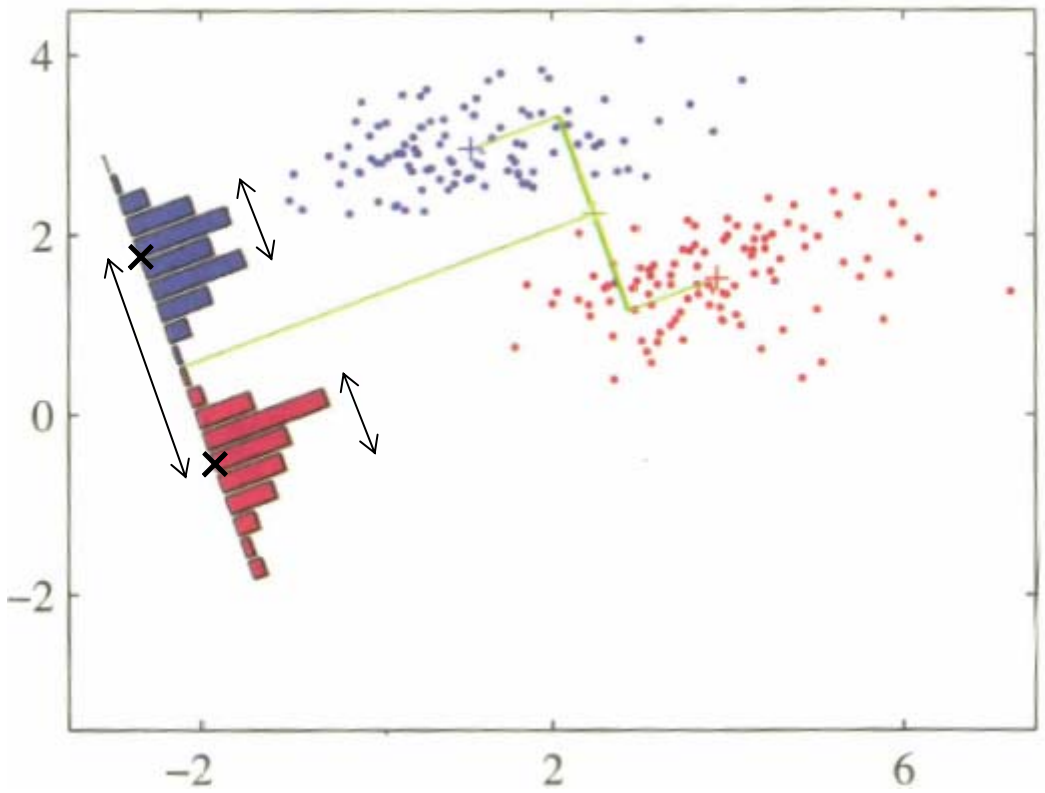
主成分分析による二次元空間への写像



固有顔法は写像後に最近傍決定則を利用

線形判別分析でデータの分布を観察する

異なるクラス同士をなるべく離すような
線型写像を求めて識別に利用する



$$J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}}$$

クラス間共分散行列

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top$$

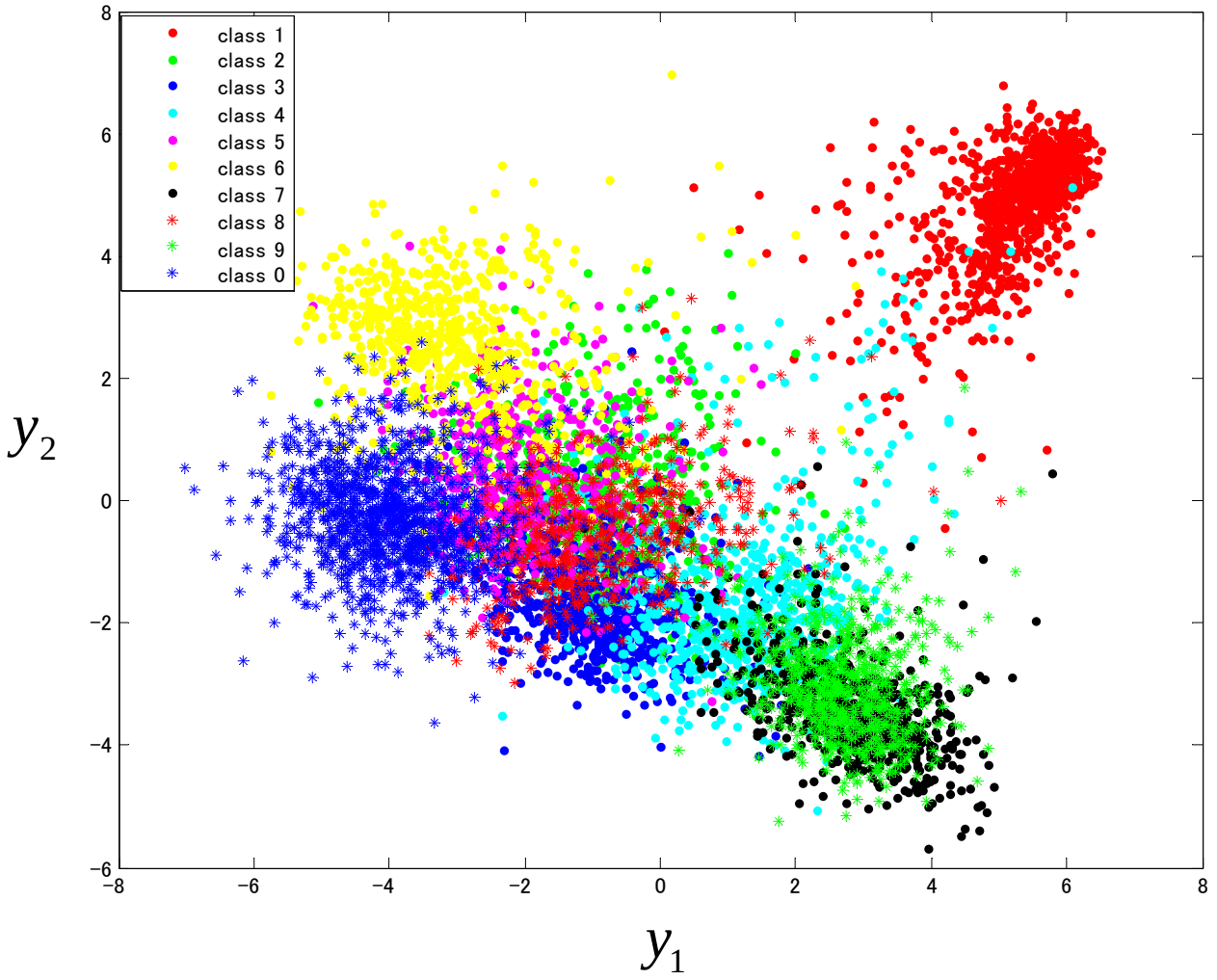
クラス内共分散行列

$$\mathbf{S}_W = \sum_{j=1,2} \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

極大点は微分して0となるから　$\partial J(\mathbf{w}) / \partial \mathbf{w} = \mathbf{0}$

結局求めたいWは　$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$

線形判別分析による二次元空間への写像



識別は閾値処理，最小距離法等を利用

代表的な手法のテストデータに対する誤識別率

識別法	誤識別率 (%)
最近傍決定則	5.5
Tangent Distance [1]	2.6
P2DHMDM with 3-nearest neighbor [2]	1.9
Human error rate [3]	2.5
Human error rate [4]	1.5

[1] Simard et al., NIPS, pp.50-58, 1993

[2] Keysers et al., ICPR, vol.4, pp.511–514, 2004.

[3] Bromley and Sackinger, Tech. Rep., 11359--910819-16TM, AT&T, 1991

[4] Dong, Report. CENPARMI, Concordia Univ., 2001.

演習　Step1
部分空間の直交基底をクラス毎に見てみる

演習で用いるデータファイルの内訳

画像は画素値を左上から順に縦に並べてベクトル化

test_data.txt　　テストデータを格納した256x2007 の行列

test_label.txt　　テストデータのラベルを格納した
2007x1 のベクトル

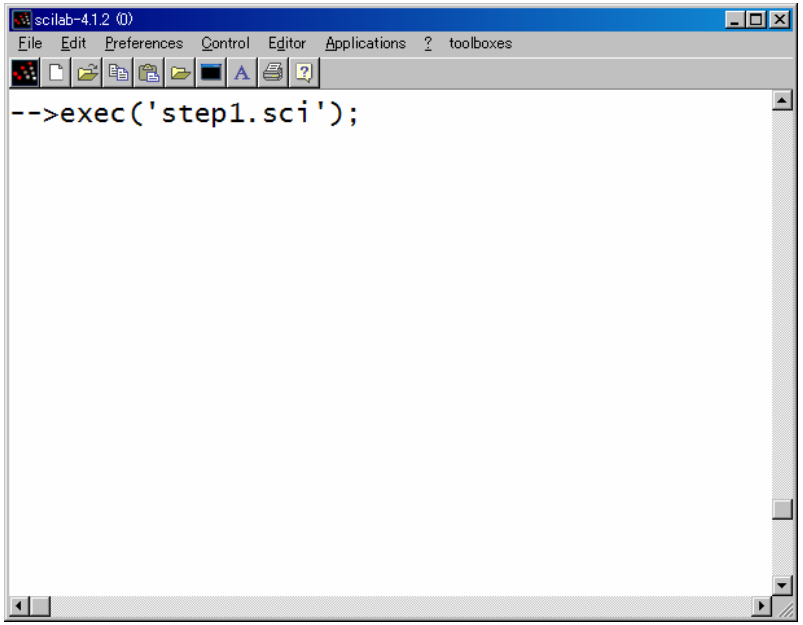
trai_data.txt　　訓練データを格納した256x7291 の行列

trai_label.txt　　訓練データのラベルを格納した
7291x1 のベクトル

Step1のプログラムを実行する

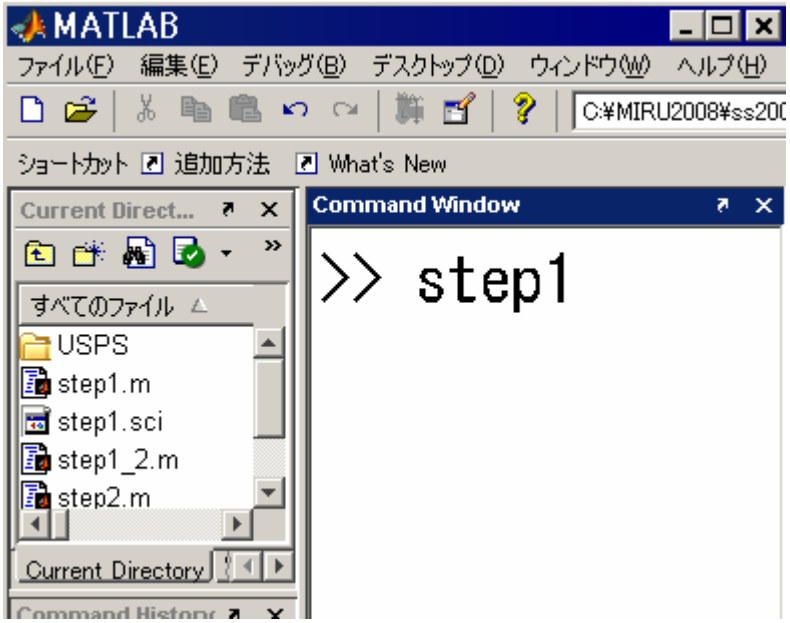
Scilab

以下のように入力

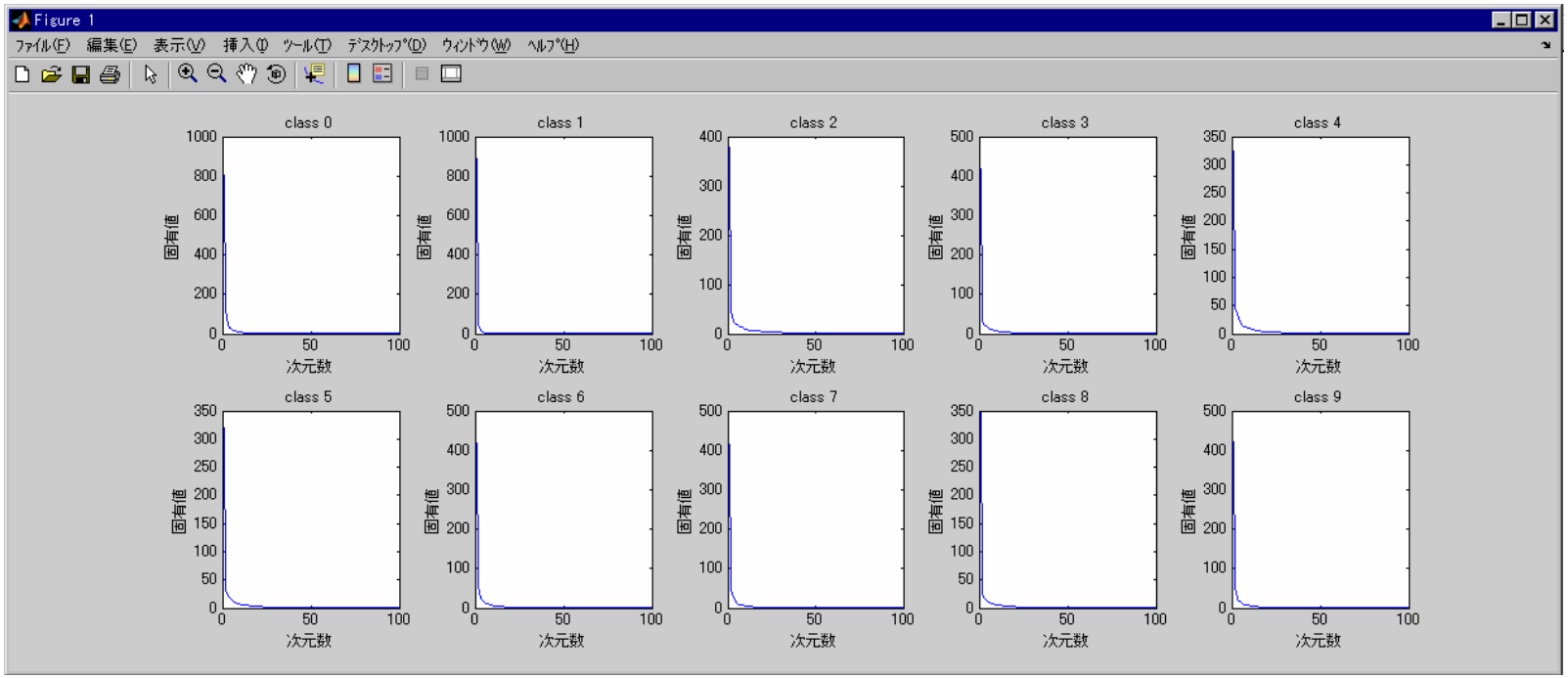


Octave, MATLAB

プロンプトでstep1と入力



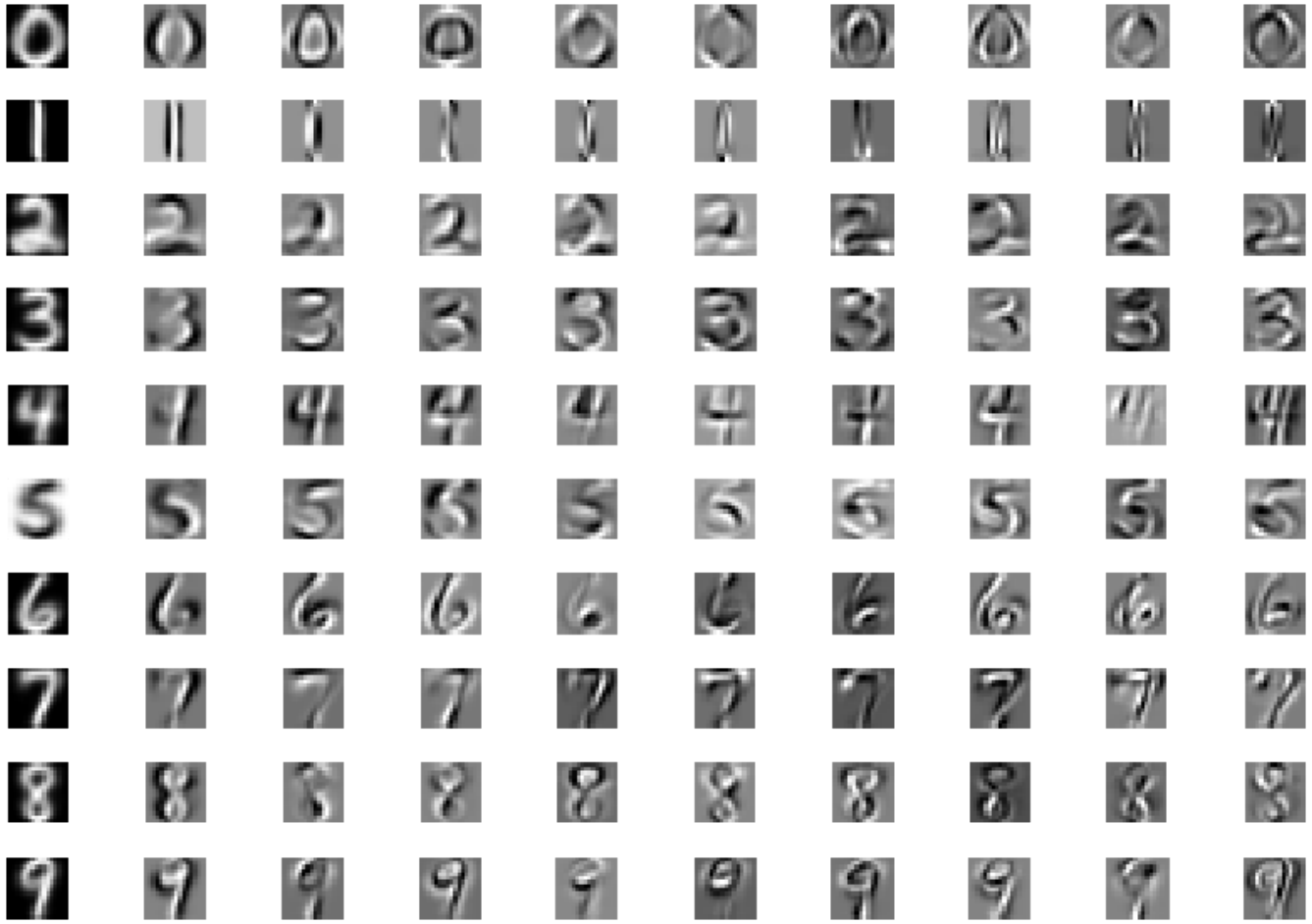
実行画面



自己相関行列の固有値の変化
縦軸が固有値，横軸が固有値の番号 (降順ソート)

少ない基底数でクラスの分布が近似できる

各クラスの直交基底



各クラスの基底を使えばパターン変動が表現できる

変動の大きいパターンに有効

プログラムの中身（Octave，MATLABの例）

```
%% 各クラスで部分空間を求める
W=zeros(Dim,100,10); % 各クラスの直交基底 次元数×基底数×クラス数
figure(1), clf, axis square;
for j = 0 : 9
  X=D(:,find(trai_label==j)); % ラベルがjの訓練パターンをXに格納
  C=X*X';                     % 自己相関行列を計算
  [eig_vec, eig_val]=eig(C);  % 固有ベクトル，固有値を計算
  [value index]=sort(-diag(eig_val)); % 固有値を降順に並べ替え
  W(:,:,j+1)=eig_vec(:,index(1:100)); % 固有値の大きい上位100本に対応する固有ベクトルをWに格納
  figure(1),subplot(2,5,j+1), plot(-value(1:100)); % 固有値をプロット
  s=sprintf('class %d', j);, title(s);
  xlabel('次元数');,ylabel('固有値');
  fprintf('class %d ... OK¥n',j);
end
```

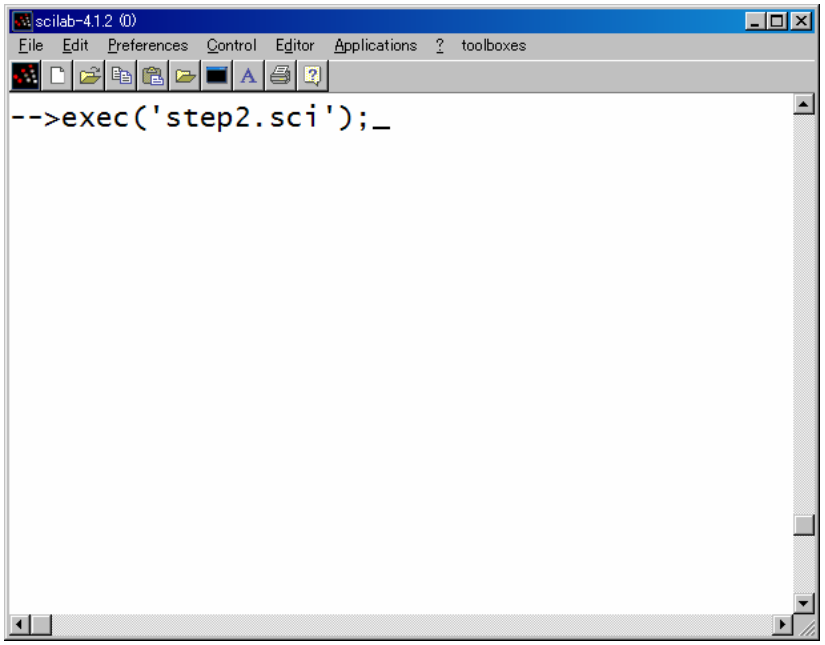
各クラスの直交基底を求めるプログラムは
５，６行で書けてしまう！

演習　Step2
画像パターンを直交展開してみる

Step2のプログラムを実行する

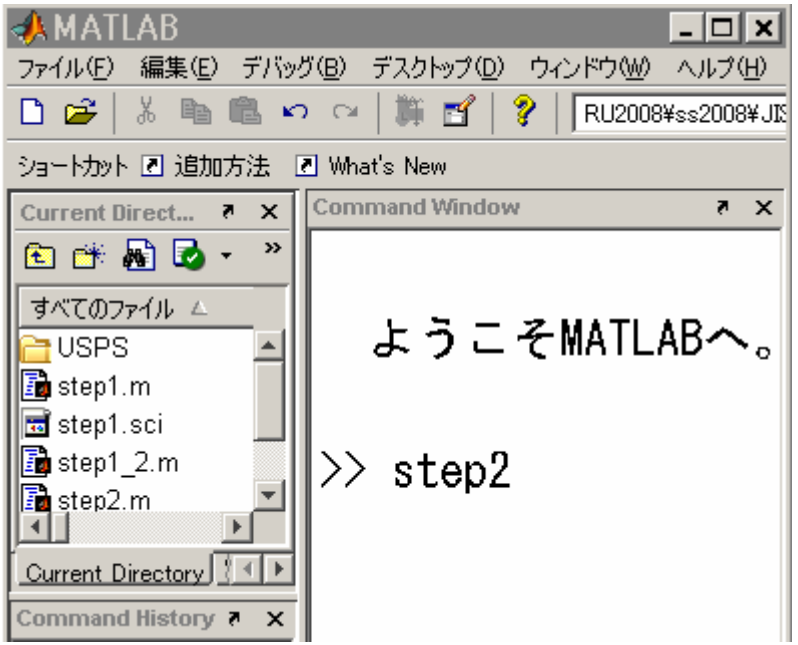
Scilab

以下のように入力



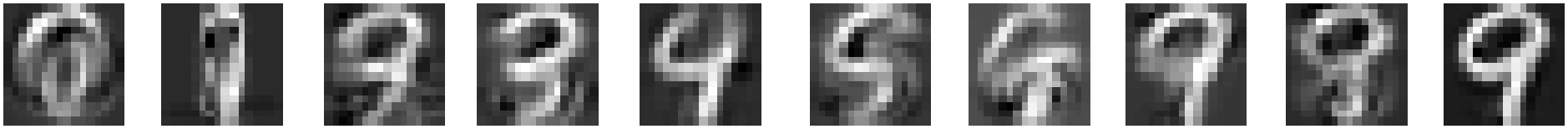
Octave, MATLAB

プロンプトでstep2と入力



実行画面

テストパターン



0　1　2　3　4　5　6　7　8　9

各クラスで30本の正規直交基底の線型結合によって
近似されたテストパターン

プログラム中のrの値を変えると近似の
程度が変わるので変えて実行してみよう

テストパターンを変えたいときはプログラムの先頭付近の
xの部分を1から2007の値で指定してください

プログラムの中身（Octave，MATLABの例）

どのように近似パターンを作成しているか

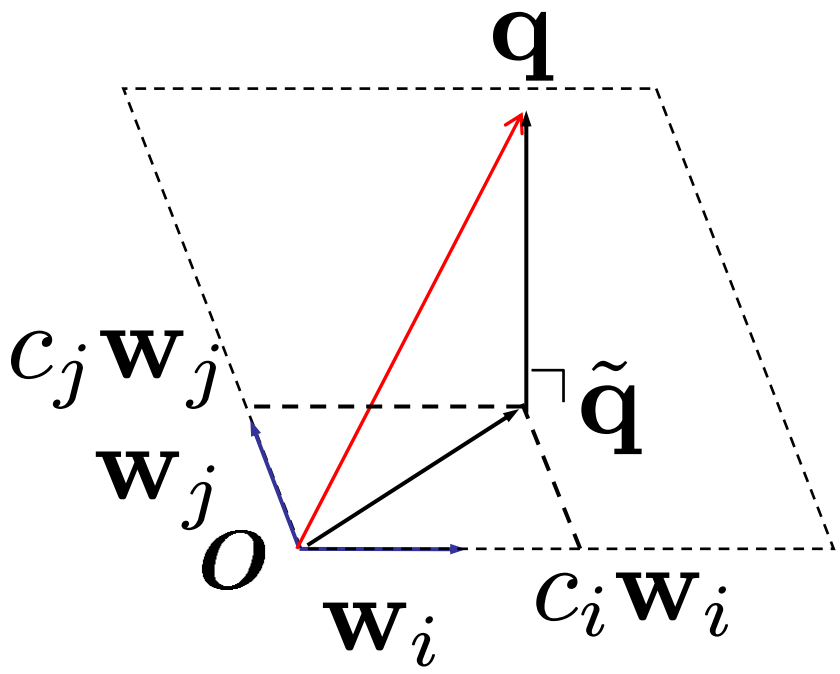
```
for j = 0 : 9
  c=W(:,1:r,j+1)'*Q(:,x); % 係数を計算
  QA=W(:,1:r,j+1)*c;        % 線型近似パターンの作成
  for i = 1 : 16, IMG(i,:)=QA((i-1).*16+1:i.*16,1)';,end
  IMG=IMG-min(min(IMG));, IMG=IMG./max(max(IMG));
  figure(1),subplot(2,10,11+j),imshow(IMG);,
  s=sprintf('class %d',j);, title(s);
end
```

なぜこれで近似パターンを作成できる？

正規直交基底による信号の展開

元信号　　近似信号　　正規直交基底

$$\mathbf{q} \simeq \tilde{\mathbf{q}} = c_1\mathbf{w}_1 + c_2\mathbf{w}_2 + \cdots + c_r\mathbf{w}_r$$



係数は元信号と正規直交基底との内積値

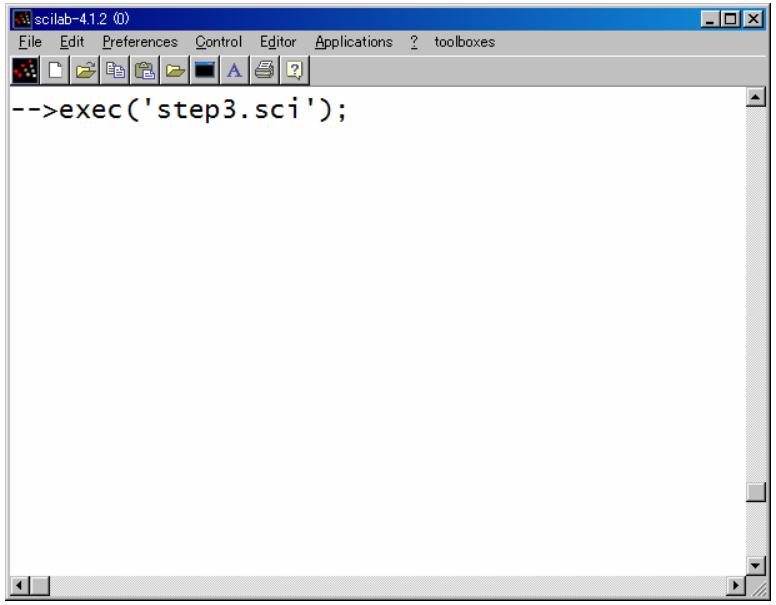
$$c_i = \mathbf{w}_i^\top \mathbf{q}$$

演習　Step3
部分空間法で手書き数字認識

Step3のプログラムを実行する

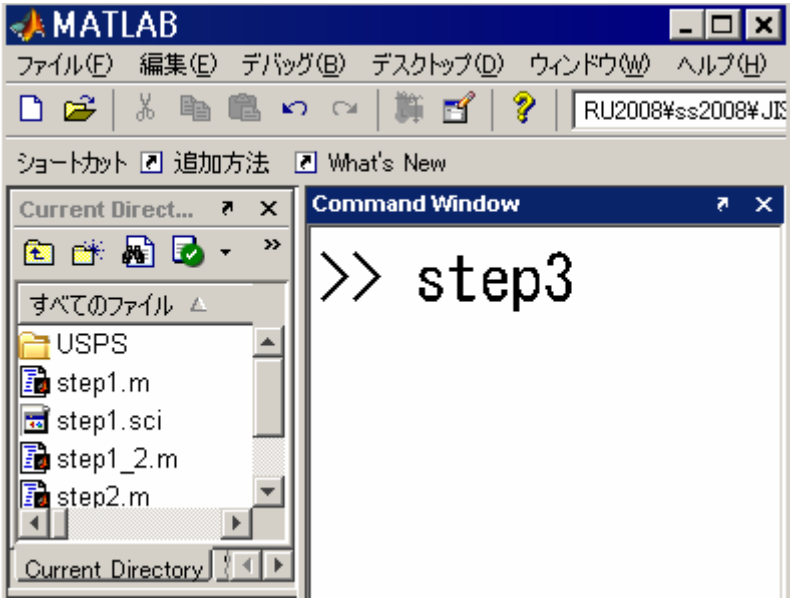
Scilab

以下のように入力

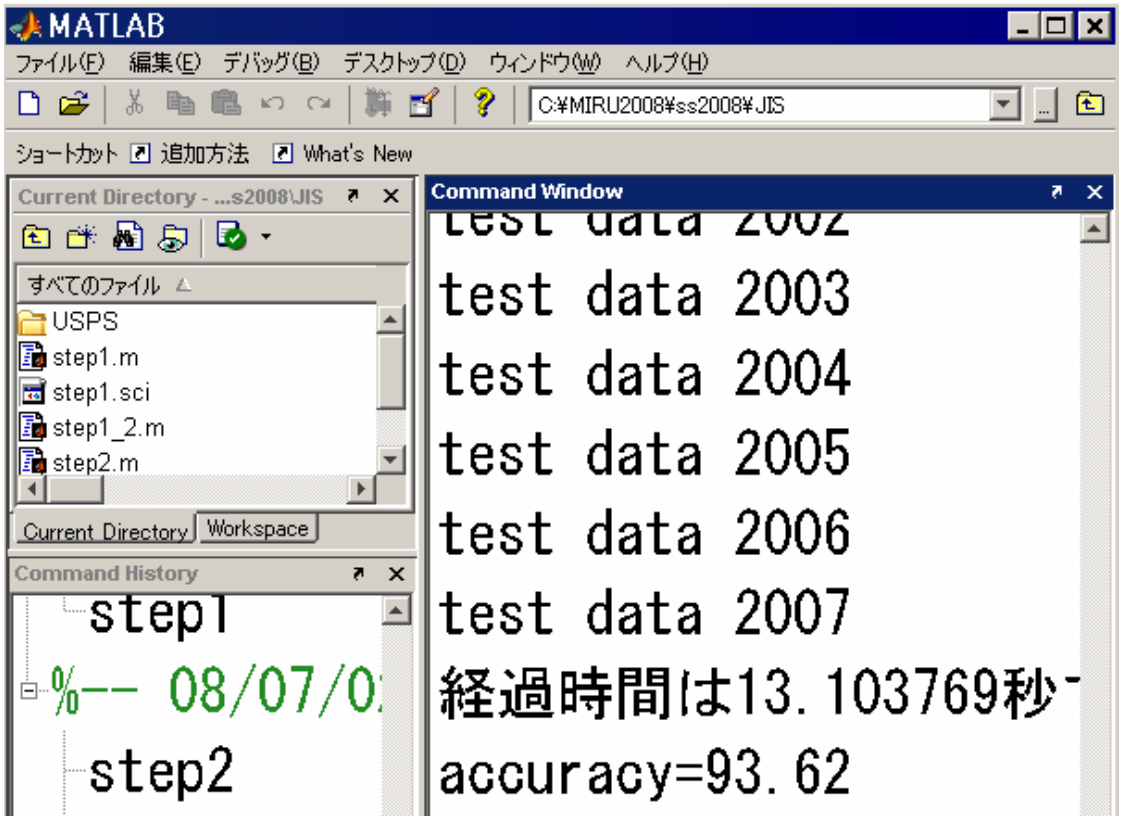


Octave, MATLAB

プロンプトでstep3と入力



実行画面



識別に要した時間と識別精度が表示される

※ 環境によって遅いかもしれませんがATLASで高速化できます

CONFと入力

```
>> CONF

CONF =

   351     3     3     0     1     0     0     0     1     0
     0   257     0     0     3     1     3     0     0     0
     7     0   180     2     3     1     0     1     4     0
     2     0     3   147     0    10     0     0     4     0
     0     3     2     0   182     1     0     1     2     9
     4     1     0     3     1   148     0     0     1     2
     0     0     1     0     2     5   161     0     1     0
     0     1     1     0     3     0     0   138     1     3
     4     5     3     4     0     2     2     0   146     0
     0     3     0     0     0     1     0     3     1   169
```

混同行列（confusion matrix）が表示される

- プログラム中のrの値を変えて識別精度や識別時間がどう変化するか観察しましょう

プログラムの中身

```
for i = 1 : test_num
  for j = 0 : 9, S(j+1)=sum((W(:,1:r,j+1)'*Q(:,i)).^2);, end
  [value index]=max(S);
  CONF(test_label(i)+1,index)=CONF(test_label(i)+1,index)+1;
  fprintf('test data %d¥n',i);
end
```

識別部分は一行で書ける！

パラメータの決め方

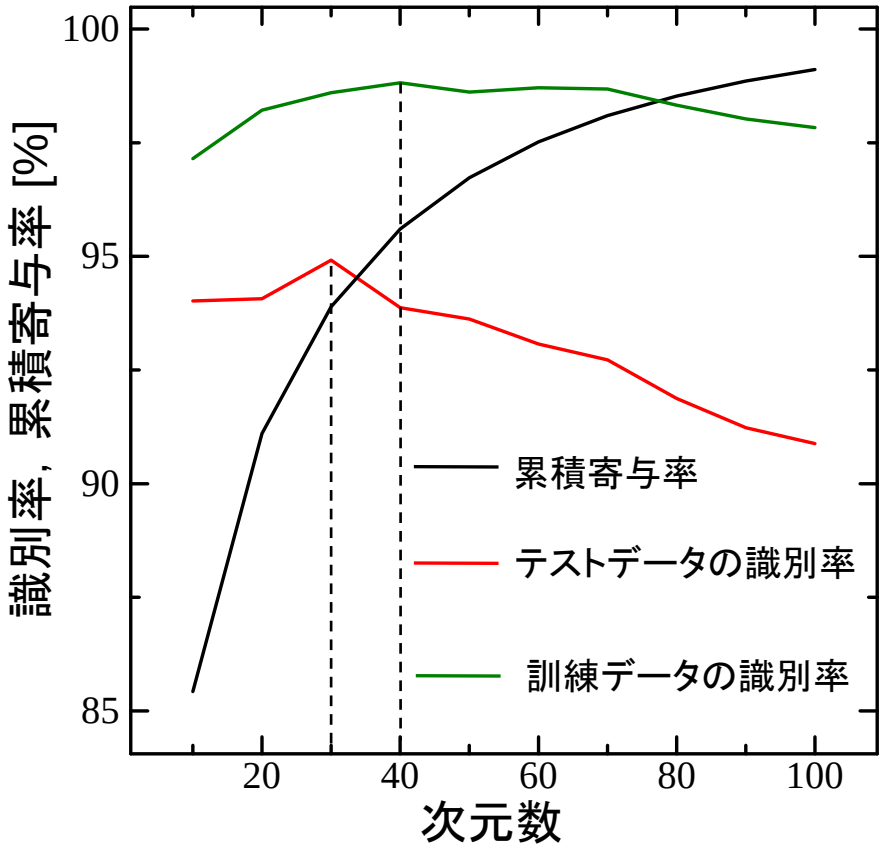
累積寄与率：r次元の部分空間を張る基底に対する固有値の総和を全固有値の総和で割ったもの

$$\sum_{i=1}^{r} \lambda_i / \sum_{j=1}^{d} \lambda_j$$

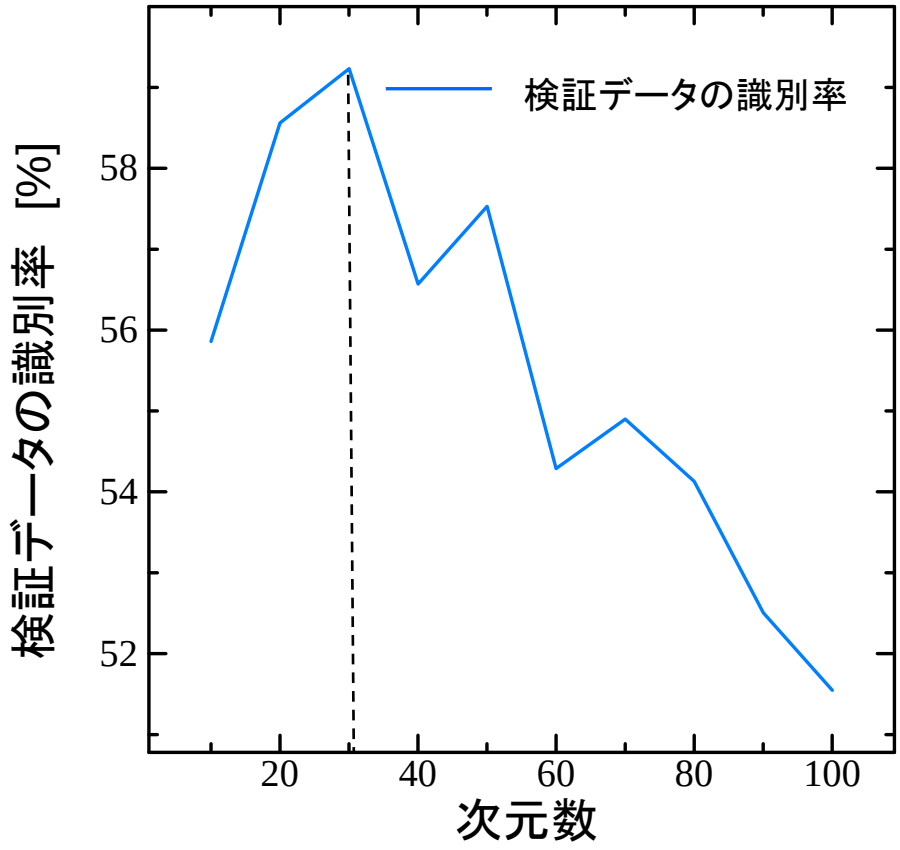
分割学習法：交差確認法などのパラメータ推定法

訓練データをテストデータと識別部を設計するためのデータに分けて識別率を算出しパラメータを推定

実験例



累積寄与率



分割学習法

パラメータ推定法を用いた方が良い

比較実験

CPU 1.86GHz，メモリ 2GbのPC上のMATLABを使用

識別法	
最小距離法	各クラスの重心との距離で識別
最近傍決定則	最近傍パターンのラベルを出力
判別分析(次元数9)	写像先の空間で最小距離法
固有数字(次元数40)	写像先の空間で最近傍決定則
部分空間法(次元数30)	CLAFIC
SVM (RBF Kernel)	SVMLIB※ を利用

※ http://www.csie.ntu.edu.tw/~cjlin/libsvm

各手法の比較

テストデータに対する誤識別率，1パターンあたりの識別時間，辞書サイズ

識別法	誤識別率 (%)	識別時間	辞書サイズ
最小距離法	18.6	3.8×10^{-5}	256×10
最近傍決定則	5.5	0.02	256×7291
判別分析(次元数9)	11.5	3.9×10^{-5}	9×10
固有数字(次元数40)	4.9	0.03	40 ×7291
部分空間法(次元数30)	5.1	0.001	256×300
SVM (RBF Kernel)	4.6	0.005	256×3220

部分空間法の特長 (1)

★ クラスを表現する直交基底を
そのまま識別に利用できる

固有顔や固有空間法では識別則は別途必要

★ 2クラス専門の識別器でないので
多クラス化が容易

SVMは多クラス化は面倒

学習により他クラスを考慮できる

部分空間法の特長 (2)

辞書サイズを部分空間の
次元数で調節できる

SVMでは辞書サイズの調節は困難

大量の訓練データを扱える

カーネル行列を使う手法では訓練データ数×訓練データ数の行列が必要なため，訓練データ数が多いと学習できない

理論的な拡張が容易

相互部分空間法や相対KL展開

さらに学びたい人のために

参考文献

- E. Oja, ``Subspace methods of pattern recognition,'' Research Studies Press, 1983. (小川英光，佐藤 誠訳，パターン認識と部分空間法，産業図書，1986) 絶版
- 黒沢由明, `` Subspace2006 部分空間法入門," 部分空間法研究会 Subspace2006, pp.136-143, 2006.
- 石井ら，わかりやすいパターン認識，オーム社，1998.
- 黒沢由明, ``部分空間法の今昔（上）: 歴史と技術的俯瞰 : 誕生から競合学習との出会いまで ,'' 情報学誌，vol.49, no.5, pp.76-82, 2008.
- 福井和広 , ``部分空間法の今昔（下）: 最近の技術動向 : 相互部分空間法への拡張とその応用事例 ,'' 情報学誌，vol.49, no.6, pp.82-87, 2008.