

パターン情報学 プログラミングレポート課題

03-140299 東京大学機械情報工学科 3 年 和田健太郎

2015 年 2 月 12 日

1 課題 1

課題 1

2 クラス (ω_1, ω_2) の識別問題を考える．データは 2 次元とする．配布するデータセットの説明を以下に示す．

- Train1.txt, Train2.txt : ω_1, ω_2 に属する訓練データ集合．各データ数 50．
- Test1.txt, Test2.txt : ω_1, ω_2 に属するテストデータ集合．各データ数 20．

2 クラスで，2 次元のデータに対するウィドロー・ホフのアルゴリズムを実装し，訓練データから分離超平面を学習せよ．また，テストデータの識別率（全テストデータ数に対する正しく識別されたテストデータ数の比率）を求めよ．さらに，訓練データ，テストデータ，学習された識別面を図示せよ．

ウィドロー・ホフのアルゴリズムを初期の重みはランダムとし，指定した回数だけ繰り返し重みの更新を行うように実装した．

2 次元の訓練データ 100 件を用いて識別器の学習を行い，40 件のテストデータで性能を測定したところ，0.875 という結果が出た．

また，訓練データ，テストデータのそれぞれ 2 クラスと識別面を図示したものが図 1 である．

2 課題 2

課題 2

擬似逆行列を計算するプログラムを書き，課題 1 と同じ訓練データから分離超平面を学習せよ．また，テストデータの識別率を求めよ．クラスラベルについて， ω_1 に属するものを 1， ω_2 に属するものを -1 などとせよ．さらに，学習された識別面を課題 1 と同じ図に示せ．

擬似逆行列を数値計算ライブラリである numpy を利用して実装した．

$$A^+ = (A^T \cdot A)^{-1} \cdot A^T$$

擬似逆行列を用いて訓練データに関して重みを計算し，テストデータによって識別性能を測定したところ，1 と同様に 0.875 という結果だった．

訓練データ，テストデータおよび識別面を図示したものが図 2 で，識別面の位置をウィドロー・ホフのアルゴリズムによるものと比べてみると，ほぼ同じ位置にあることがわかる．

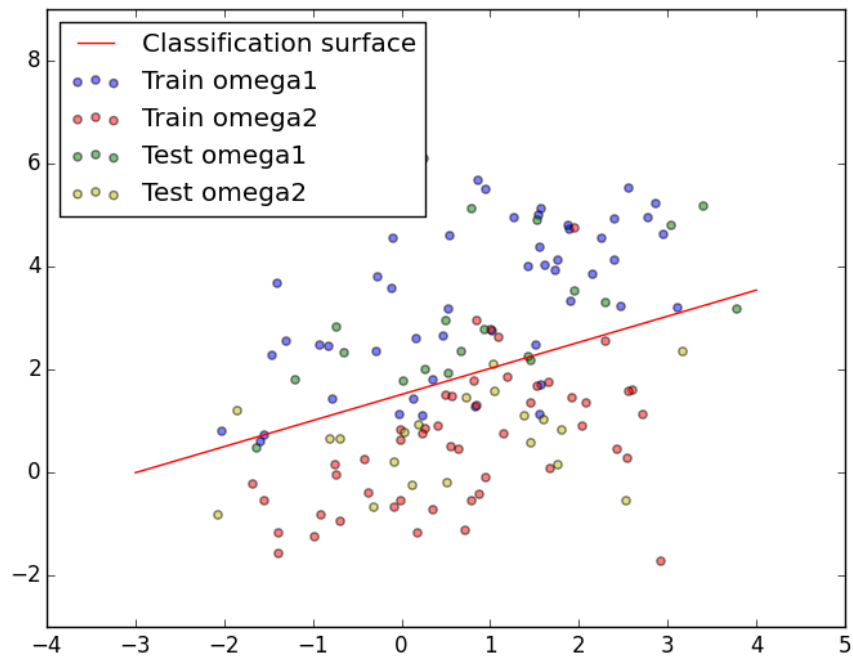


図 1: データおよび識別面

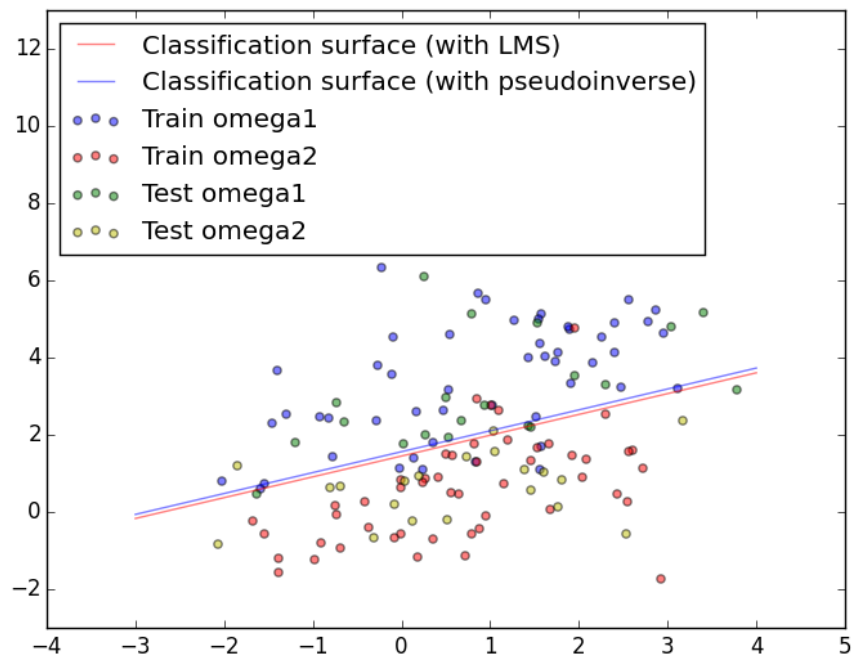


図 2: データおよび識別面

3 課題 3

課題 3

本課題も課題 1 と同じデータセットを利用する。

1. テストデータの集合を k 近傍法 (kNN) を用いて識別することを考える。訓練データに対して一つ抜き出し, (LOO: leave-one-out) 法により k の値を 1 から 10 まで変化させ, 最適な k の値を求めよ。また, 横軸に k , 縦軸に識別率としてグラフを作成せよ。
2. LOO により得られた k の値を用いてテストデータを識別せよ。そして, 識別率を求めよ。

訓練データに対して LOO により識別を行い, k の値を 1 から 10 まで変化させて識別率を測定した。その関係を示したのが図 3 である。図 3 より, k が 3 の時に最も識別性能が高くなっていることがわかる。

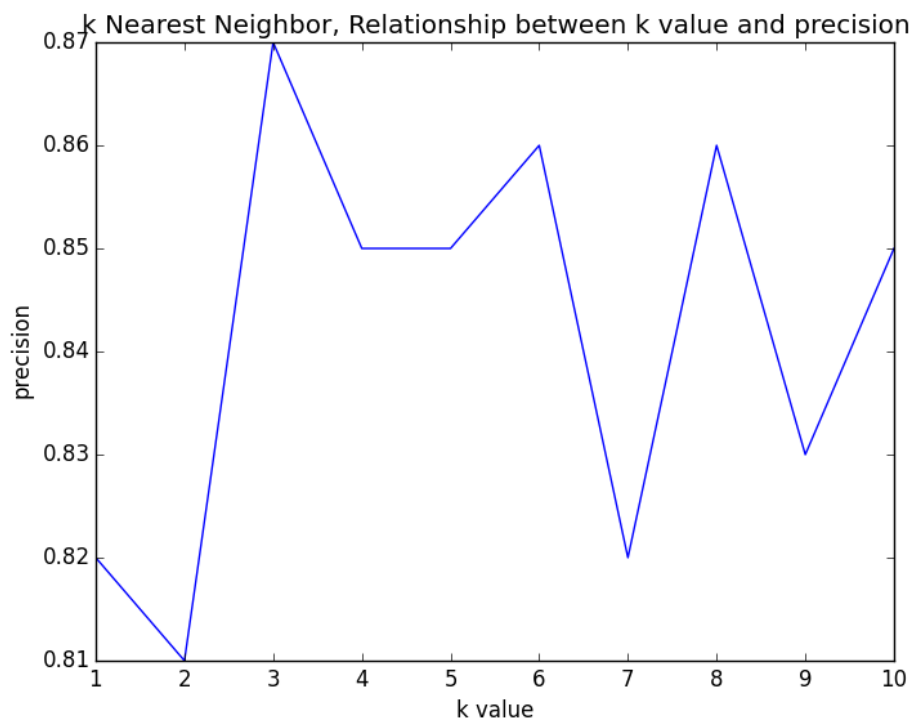


図 3: k 値と LOO 法による kNN の識別率の関係

$k=3$ の時, テストデータに対してテストデータの識別を行ったところ, 識別率は 0.85 になった。

4 課題 4

課題 4

表にあるデータを利用する．また潜在的な確率密度分布は正規分布であるとする． $P(\omega_i)=1/3$ とする．表にあげた各クラスのデータセットは omega1.txt , omega2.txt , omega3.txt である．このとき次の問いに答えよ．

1. テスト点： $(1, 2, 1)^T$, $(5, 3, 2)^T$, $(0, 0, 0)^T$, $(1, 0, 0)^T$ と各クラスの平均との間のマハラノビス距離を求めよ．
2. これらの点を識別せよ．
3. 次に $P(\omega_1)=0.8$ かつ $P(\omega_2) = P(\omega_3)=0.1$ と仮定し，テスト点をもう一度識別せよ

テスト点： $(1, 2, 1)^T$, $(5, 3, 2)^T$, $(0, 0, 0)^T$, $(1, 0, 0)^T$ に関して，各クラス集合の平均とのマハラノビス距離

$$M_D(x) = \sqrt{(x - \mu_i)^T \sum (x - \mu_i)} \quad (1)$$

を表 1 に計算した．

sample points	ω_1	ω_2	ω_3
$(1, 2, 1)^T$	1.0149706212	0.85805119543	2.67475703681
$(5, 3, 2)^T$	1.557138211	1.75568068865	0.647009014093
$(0, 0, 0)^T$	0.489961541569	0.268432411153	2.24150137149
$(1, 0, 0)^T$	0.487236758687	0.451834352153	1.46233640166

表 1: テスト点の各クラス集合の平均とのマハラノビス距離

確率的生成モデルを用いて，これらのテスト点を識別したところ，表 2 に示す識別結果となった． $(P(\omega_i) = 1/3)$
 $P(\omega_1) = 0.8$, $P(\omega_2) = P(\omega_3) = 0.1$ として識別を行ったところ表 3 に示す識別結果となり， ω_1 にすべてのテスト点が分類されるものとなった．

$(1, 2, 1)^T$	$(5, 3, 2)^T$	$(0, 0, 0)^T$	$(1, 0, 0)^T$
ω_2	ω_3	ω_1	ω_1

表 2: $P\omega_i = 1/3$ での識別結果

$(1, 2, 1)^T$	$(5, 3, 2)^T$	$(0, 0, 0)^T$	$(1, 0, 0)^T$
ω_1	ω_1	ω_1	ω_1

表 3: $P\omega_1 = 0.8$, $P\omega_2 = P\omega_3 = 0.1$ での識別結果

5 チャレンジ課題 1

チャレンジ課題 1

主成分分析, 多クラスフィッシャー判別分析を実装せよ. また, 3 クラス, 4 次元の iris データセット iris.txt に主成分分析とフィッシャー判別分析をそれぞれ適応して 1 次元に次元削減し図示せよ. 次元削減後のクラス間データの分離の違いを確認せよ. なお iris データセットの各行はデータのインデックス, 第 5 列はクラス番号 (1, 2, 3 クラス) を示している. 各クラス 50 サンプル合計 150 サンプルとなる.

主成分分析とフィッシャー線形判別による特徴空間の変換結果を 4 に示す.

PCA に比べ FisherLDA ではクラス 1 とクラス 2, 3 とのクラス間分散が大きくなり, クラス 3 のクラス内分散が小さくなっていることがわかる. また, クラス 2, 3 の重なりも FisherLDA の方が小さくなっている.

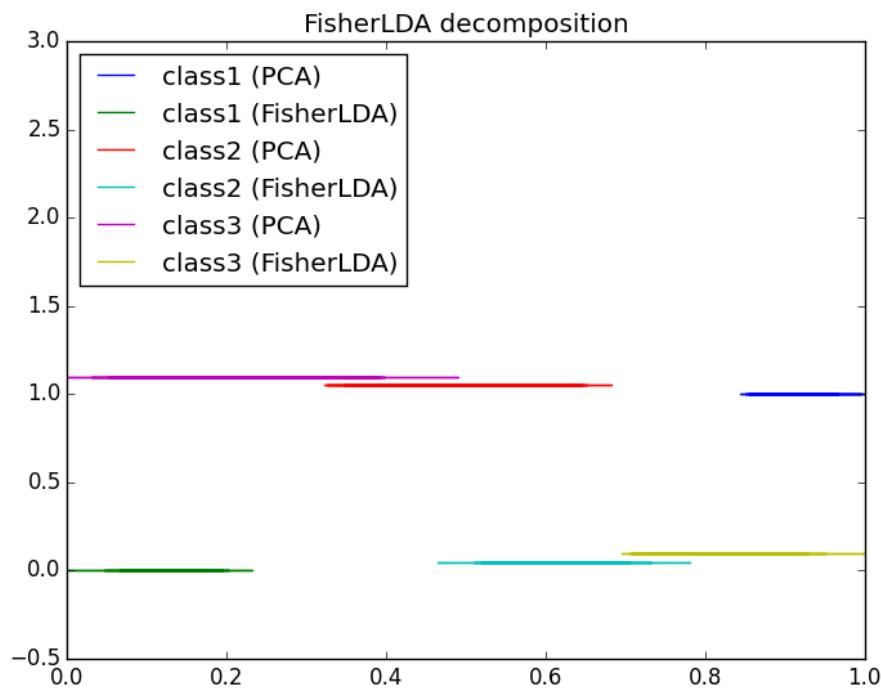


図 4: PCA と FisherLDA による特徴空間の変換 (データ:iris.txt)

6 チャレンジ課題 2

チャレンジ課題 2

ロジスティック回帰を実装し, 課題 1 のデータに適用してテストデータの識別率を求めよ.

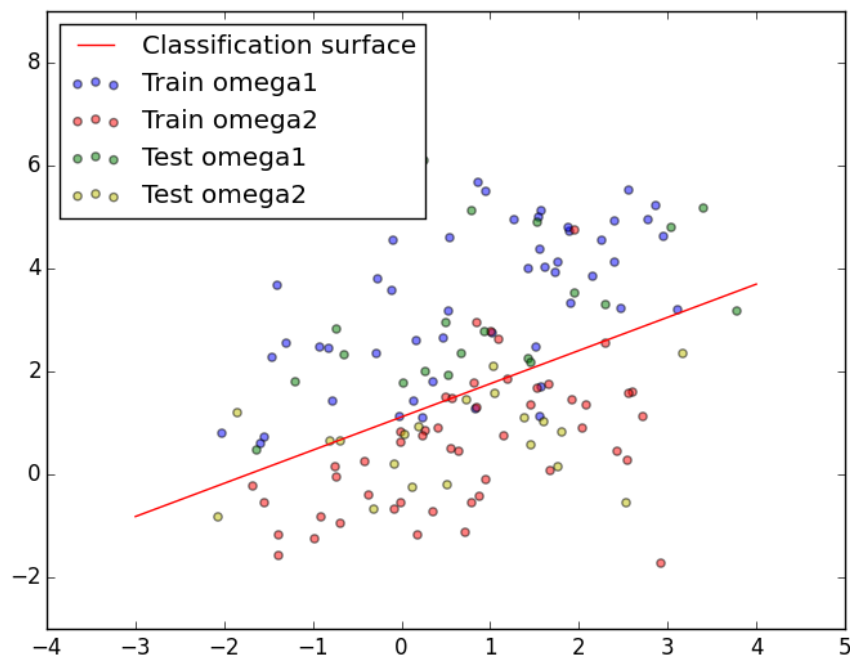


図 5: ロジスティック回帰による識別

7 チャレンジ課題3

チャレンジ課題3

表情認識を行うプログラムを作成せよ。データセットには日本人女性の顔のデータセット (JAFFE データセット) を用いよ。JAFFE データセットには 10 名の被験者から得られた 213 枚の画像が含まれ、7 つの表情 (幸福, 悲しみ, 驚き, 怒り, 不快, 恐れ, 無表情) を行っている。このデータセットで、10 名のうち 9 名の画像を用いて学習を行った後、残りの 1 名に対して表情識別テストを行え。この試行を 10 名に対して繰り返すことで 10 名分の平均識別率を求めよ。画像からの特徴量, 表情の識別手法は各自好きなものを用いて良い。もちろん独自に勉強した手法も大歓迎。7 つの表情を全て識別させるのは難しいので、幸福と悲しみの 2 クラスだけでもよい。

特徴抽出を HOG, Autoencoder [1] を用いて行い, 識別器は SVM [2] とロジスティック回帰を利用した。特徴抽出と識別率は表 4 のようになった。

特徴抽出	識別器	識別率
None	SVM	0.5812836
HOG	SVM	0.5887888
Autoencoder	SVM	0.2878157
HOG+Autoencoder	SVM	0.4934857
None	ロジスティック回帰	0.669559570864
HOG	ロジスティック回帰	0.6126256

表 4: 特徴抽出, 識別器と識別率

参考文献

- [1] 2008-2010, LISA lab., <http://deeplearning.net/tutorial/dA.html>
- [2] 2010-2014, scikit-learn developers, <http://scikit-learn.org/stable/modules/svm.html>