Homework 4 #Solution

April 3, 2014

Problem 1

1. When the underlying distributions are Gaussian with equal covariance matrix $\mathbf{\Sigma}$, then the decision boundary found by Bayesian decision theory is the same with that obtained by FLD.

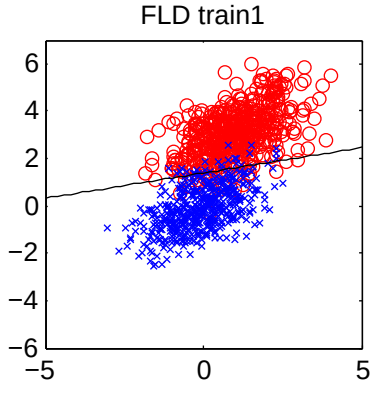
2. Error rate on **test1**:

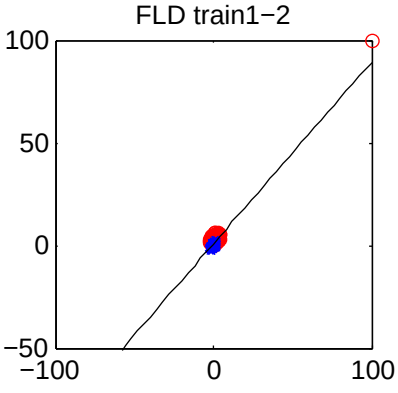
Method	**train1**	**train1_2**
FLD	7%	15.1%
Logistic regression	6.7%	6.7%

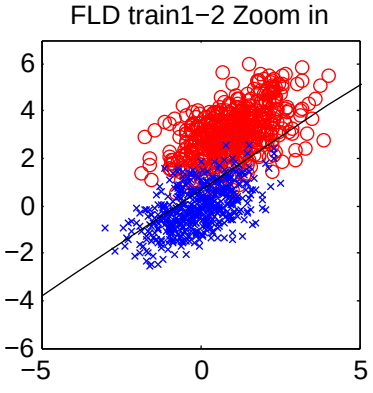
   For logistic regression, the error rate on **test1** does not change no matter **train1** or **train1_2** is used for training. On the contrary when FLD is used, training with **train1_2** gives much larger testing error rate.

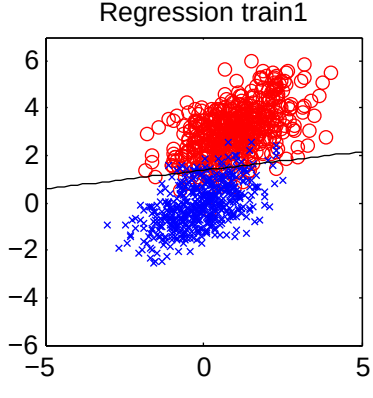
   When **train1** is used, FLD and logistic regression provides similar testing error rate. Because **train1** data is generated from Gaussian distributions with the same covariance matrix, under the condition of which FLD gives the same decision boundary as with Bayesian decision theory (if the number of training samples is sufficiently large). The decision boundary that corresponds to minimum error rate is correctly found. In the same situation, logistic regression also find the decision-theoretical optimal boundary. Hence, both FLD and logistic regression works well and provides similar testing error rate.

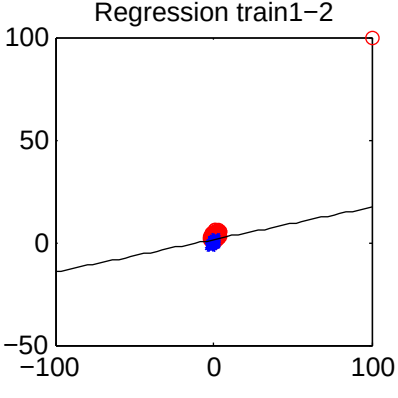
   We observe that the addition of the outlier does not affect logistic regression, but has a strong negative impact on FLD. Since the new point is correctly classified and far from the boundary, the logistic model assigns to the outlier a probability $p(y|x)$ exponentially close to 1. Adding this probability to the likelihood has almost no effect on the criterion. Because the probability is already maximum at the point, so logistic regression is not affected. On the other hand the FLD sees the data as if each class is Gaussian, and the addition of a single point very far from the current mean can greatly affect the estimate of the mean and variance of that Gaussian.

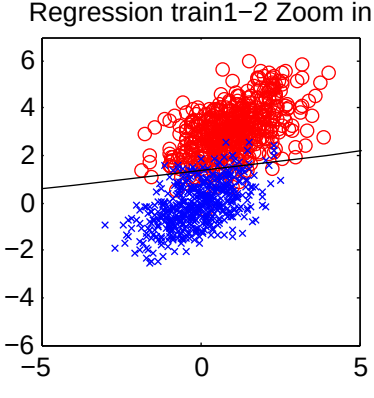

FLD train1
6
4
2
0
−2
−4
−6
−5
0
5


FLD train1−2
100
50
0
−50
−100
0
100


FLD train1−2 Zoom in
6
4
2
0
−2
−4
−6
−5
0
5


Regression train1
6
4
2
0
−2
−4
−6
−5
0
5


Regression train1−2
100
50
0
−50
−100
0
100


Regression train1−2 Zoom in
6
4
2
0
−2
−4
−6
−5
0
5

3. Testing error rate on **test2**:

Method	FLD	Logistic regression
Error rate	23%	14.25%

Both FLD and logistic regression have higher testing error on **test2**. The Gaussian distribution assumption is not reasonable for these binary images. In this representation of digital image, the 64×1 feature vector has components equal to either 1 or 0. Some digits are written in more than one way (they can be written in different styles) and also with rotations, shear, scaling, or translations. These variants violate the Gaussian model. FLD is sensitive to the Gaussian assumption. It is the same as Bayesian estimation only under the Gaussian assumption as mentioned above. On the contrary, logistic regression is less sensitive to the Gaussian assumption. It does not assume the conditional density of the classes, but directly models the decision boundary. That is the reason why logistic regression performs better than FLD on **test2**.

Problem 2

1.

$$\begin{aligned} K_3(\mathbf{x}, \mathbf{x}') &= K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}') = \phi^{(1)}(\mathbf{x}) \cdot \phi^{(1)}(\mathbf{x}') + \phi^{(2)}(\mathbf{x}) \cdot \phi^{(2)}(\mathbf{x}') \\ &= \begin{bmatrix} \phi^{(1)}(\mathbf{x}) \\ \phi^{(2)}(\mathbf{x}) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \phi^{(1)}(\mathbf{x}') \\ \phi^{(2)}(\mathbf{x}') \end{bmatrix} = \begin{bmatrix} \phi^{(1)}(\mathbf{x}) \\ \phi^{(2)}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \phi^{(1)}(\mathbf{x}') \\ \phi^{(2)}(\mathbf{x}') \end{bmatrix} \end{aligned}$$

i.e. $\phi^{(3)}(\mathbf{x}) = \begin{bmatrix} \phi^{(1)}(\mathbf{x}) \\ \phi^{(2)}(\mathbf{x}) \end{bmatrix}$

2. Let $\phi^{(1)}(\mathbf{x}) = [a_1, a_2, \ldots, a_k]$, $\phi^{(1)}(\mathbf{x}') = [a'_1, a'_2, \ldots, a'_k]$ and $\phi^{(2)}(\mathbf{x}) = [b_1, b_2, \ldots, b_l]$, $\phi^{(2)}(\mathbf{x}') = [b'_1, b'_2, \ldots, b'_l]$, then we have

$$K_1(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{k} a_i a'_i \quad \text{and} \quad K_2(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{l} b_j b'_j$$

Hence,

$$K_4(\mathbf{x},\mathbf{x}') = K_1(\mathbf{x},\mathbf{x}')K_2(\mathbf{x},\mathbf{x}') = \left(\sum_{i=1}^{k} a_i a_i'\right)\left(\sum_{j=1}^{l} b_j b_j'\right)$$

$$= \sum_{i=1}^{k}\sum_{j=1}^{l}(a_i b_j)(a_i' b_j') = \begin{bmatrix} a_1b_1 & a_1b_2 & \cdots a_ib_j \cdots & a_kb_l \end{bmatrix} \begin{bmatrix} a_1'b_1' \\ a_1'b_2' \\ \vdots \\ a_i'b_j' \\ \vdots \\ a_k'b_l' \end{bmatrix}$$

$$= \phi^{(4)}(\mathbf{x}) \cdot \phi^{(4)}(\mathbf{x}')$$

Finally, we have: $\phi^{(4)}(\mathbf{x}) = [a_1b_1, a_1b_2, \cdots a_ib_j \cdots a_kb_l] = \phi^{(1)}(\mathbf{x}) \bigotimes \phi^{(2)}(\mathbf{x})$
(The notation $\bigotimes$ represents kronecker product.)

3.

$$\begin{aligned} K(\mathbf{x},\mathbf{x}') &= 1 + x_1x_1' + x_2x_2' + 4(x_1x_1' + x_2x_2')^2 \\ &= 1 + x_1x_1' + x_2x_2' + 4x_1^2x_1'^2 + 4x_2^2x_2'^2 + 8x_1x_2x_1'x_2' \\ &= \phi(\mathbf{x}) \cdot \phi(\mathbf{x}') \end{aligned}$$

where $\phi(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & 2x_1^2 & 2x_2^2 & 2\sqrt{2}x_1x_2 \end{bmatrix}$

Problem 3

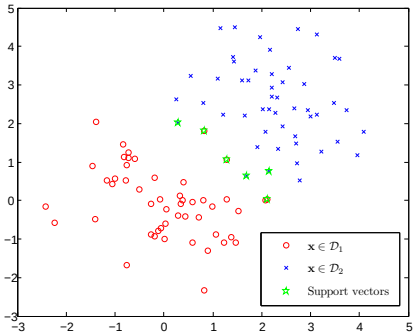
SVM classification errors with different kernels on different data sets:

Table 1: Classification error on different data sets

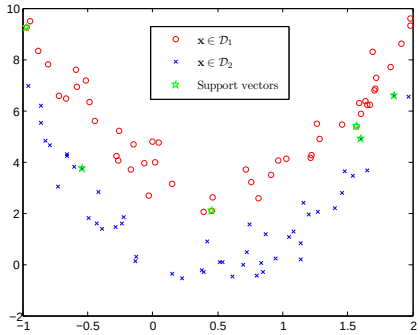
Data set	Classification error		
	linear	polynomial	radial basis
set 1	**95.54%**	94.86%	94.13%
set 2	72.70%	**98.90%**	98.40%
set 3	51.10%	85.80%	**100%**
set 4	86.25%	88.00%	**83.5%**

Plots of support vectors for set 1—3:

(a) set 1 (linear SVM)



(b) set 2 (polynomial SVM)



(c) set 3 (radial basis kernel)