SGN-41007 Pattern Recognition and Machine Learning
Exam 10.4.2019
Heikki Huttunen

- ▷ Use of calculator is allowed.
- ▷ Use of other materials is not allowed.
- ▷ The exam questions need not be returned after the exam.
- ▷ You may answer in English or Finnish.

1. Are the following statements true or false? No need to justify your answer, just T or F. Correct answer: 1 pts, wrong answer: $-\frac{1}{2}$ pts, no answer 0 pts.

   (a) Maximum likelihood estimators are unbiased.

   (b) Least squares estimator minimizes the squared distance between neighboring samples.

   (c) The LDA maximizes the following score:

   $$J(\mathbf{w}) = \frac{\text{Squared distance of class means}}{\text{Variance of classes}}$$

   (d) A neural network classifier has a linear decision boundary between classes.

   (e) L1 and L2 regularization add a penalty that forces the model weights towards zero.

   (f) Maxpooling computes the maximum over input channels, i.e., `max(x, axis = -1)` for `x` of shape `(height, width, channels)`.

2. Two measurements $x(n)$ and $y(n)$ depend on each other in a linear manner, and there are the following measurements available:

n	0	1	2
$x(n)$	7	9	4
$y(n)$	12	15	4

   We want to model the relationship between the two variables using the model:

   $$y(n) = ax(n) + b.$$

   Find the L_2-regularized least squares estimates $\hat{a}$ and $\hat{b}$ that minimize the squared error using penalty $\lambda = 10$.[1]

$$\hat{\theta} = \begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \left(X^T X + \lambda J\right)^{-1} \cdot X^T y$$

$$\begin{matrix} 1 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

[1] Alternatively, the unregularized solution will give you max. 4 points.

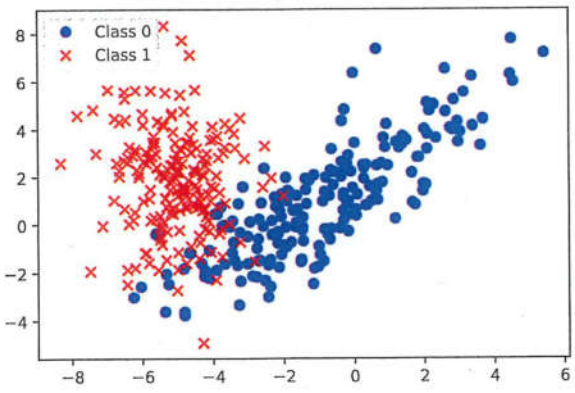

Figure 1: Training sample of question 3.

	Prediction	**True label**
Sample 1	0.8	1
Sample 2	0.3	1
Sample 3	0.5	0
Sample 4	0.25	0

Table 1: Results on test data for question 5.

3. A dataset consists of two classes shown in Figure 1. The means and covariances of the classes are

$$\mu_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \qquad \mu_1 = \begin{pmatrix} -5 \\ 2 \end{pmatrix}$$

$$\mathbf{C}_0 = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix} \qquad \mathbf{C}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}.$$

Find the Linear Discriminant Analysis (LDA) weight vector $\mathbf{w}$ for this data and find the threshold c at the center of the projected class means.

Present the decision rule for sample $\mathbf{x} \in \mathbb{R}^2$ in the following format:

$$\text{Class}(\mathbf{x}) = \begin{cases} 1, & \text{if } \boxed{\textit{something}} \\ 2, & \text{otherwise} \end{cases}$$

4. Consider the Keras model defined in Listing 2. Inputs are 64×64 color images from two categories, and all convolutional kernels are of size $w \times h = 5 \times 5$.

   (a) Draw a diagram of the network.

   (b) Compute the number of parameters for each layer.

   (c) How many scalar multiplications take place on the first convolutional layer?

w·h·N·input+N

```
model.summary()
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_49 (Conv2D)           (None, 64, 64, 32)
_________________________________________________________________
max_pooling2d_47 (MaxPooling (None, 16, 16, 32)
_________________________________________________________________
conv2d_50 (Conv2D)           (None, 16, 16, 32)
_________________________________________________________________
max_pooling2d_48 (MaxPooling (None, 4, 4, 32)
_________________________________________________________________
flatten_15 (Flatten)         (None, 512)
_________________________________________________________________
dense_29 (Dense)             (None, 100)
_________________________________________________________________
dense_30 (Dense)             (None, 2)
=================================================================
Total params: 79,566
Trainable params: 79,566
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: A CNN model summary as reported by Keras

5. A random forest classifier is trained on training data set and the `predict_proba` method is applied on the test data of five samples. The predictions and true labels are in Table 1. Draw the receiver operating characteristic curve. What is the Area Under Curve (AUC) score?

Related Wikipedia pages

Inversion of 2 × 2 matrices [edit]

The *cofactor equation* listed above yields the following result for 2 × 2 matrices. Inversion of these matrices can be done as follows:[6]

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

ROC space [edit]

The contingency table can derive several evaluation "metrics" (see infobox). To draw a ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

A ROC space is defined by FPR and TPR as *x* and *y* axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 − specificity, the ROC graph is sometimes called the sensitivity vs (1 − specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space.

Another complication in applying LDA and Fisher's discriminant to real data occurs when the number of measurements of each sample (i.e., the dimensionality of each data vector) exceeds the number of samples in each class.[4] In this case, the covariance estimates do not have full rank, and so cannot be inverted. There are a number of ways to deal with this. One is to use a pseudo inverse instead of the usual matrix inverse in the above formulae. However, better numeric stability may be achieved by first projecting the problem onto the subspace spanned by Σ_b.[12] Another strategy to deal with small sample size is to use a shrinkage estimator of the covariance matrix, which can be expressed mathematically as

$$\Sigma = (1 - \lambda)\Sigma + \lambda I$$

where I is the identity matrix, and λ is the *shrinkage intensity* or *regularisation parameter*. This leads to the framework of regularized discriminant analysis[13] or shrinkage discriminant analysis.[14]