SGN-41007 Pattern Recognition and Machine Learning
Exam 9.10.2019
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) The Receiver Operating Characteristics curve plots the probability of detection versus the probability of false alarm for all thresholds.

   (c) Least squares estimator minimizes the squared distance between the data and the model.

   (d) The number of support vectors of a support vector machine equals the total number of samples.

   (e) The LDA maximizes the variance of samples in each classes.

   (f) Cross-validation is used for model accuracy evaluation.

2. The *Poisson distribution* is a discrete probability distribution that expresses the probability of a number of events $x \geq 0$ occurring in a fixed period of time:

$$p(x; \lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$$

   We measure N samples: $x_0, x_1, \ldots, x_{N-1}$ and assume they are Poisson distributed and independent of each other.

   (a) Compute the probability $p(\mathbf{x}; \lambda)$ of observing the samples $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$. (1p)

   (b) Compute the natural logarithm of p, *i.e.*, $\log p(\mathbf{x}; \lambda)$. (1p)

   (c) Differentiate the result with respect to λ. (2p)

   (d) Find the maximum of the function, *i.e.*, the value where $\frac{\partial}{\partial \lambda} \log p(\mathbf{x}; \lambda) = 0$. (2p)

	Prediction	True label
Sample 1	0.8	1
Sample 2	0.5	1
Sample 3	0.6	0
Sample 4	0.1	0

Table 1: Results on test data for question 5a.

3. 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]

4. (6 pts) Consider the Keras model defined in Listing 1. Inputs are 64×64 color images from 10 categories.

   (a) Draw a diagram of the network.

   (b) Compute the number of parameters for each layer, and their total number over all layers.

5. (a) (4p) A random forest classifier is trained on training data set and the `predict_proba` method is applied on the test data of Table 1. Draw the receiver operating characteristic curve. What is the Area Under Curve (AUC) score?

   (b) (2p) A binary classifier is trained with 1 million samples from two classes. The AUC of the classifier on test data with another 1 million samples, is 0.768. We choose one sample from the positive class at random and another three samples from the negative class at random. What is the probability that the sample from the positive class has highest score of the four samples [hint: study the literature on the last page]?

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

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]

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

The ROC curve simply plots $T(t)$ against $F(t)$ while varying t from 0 to 1. Thus, if we view T as a function of F, the AUC can be rewritten as follows.

$$\begin{aligned}
\mathrm{AUC} &= \int_0^1 T(F_0)\,\mathrm{d}F_0 \\
&= \int_0^1 P[\hat{p}(\mathbf{x}) > F^{-1}(F_0) \mid y(\mathbf{x}) = 1]\,\mathrm{d}F_0 \\
&= \int_1^0 P[\hat{p}(\mathbf{x}) > F^{-1}(F(t)) \mid y(\mathbf{x}) = 1] \cdot \frac{\partial F(t)}{\partial t}\,\mathrm{d}t \\
&= \int_0^1 P[\hat{p}(\mathbf{x}) > t \mid y(\mathbf{x}) = 1] \cdot P[\hat{p}(\mathbf{x}') = t \mid y(\mathbf{x}') = 0]\,\mathrm{d}t \\
&= \int_0^1 P[\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{x}') \,\&\, \hat{p}(\mathbf{x}') = t \mid y(\mathbf{x}) = 1 \,\&\, y(\mathbf{x}') = 0]\,\mathrm{d}t \\
&= P[\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{x}') \mid y(\mathbf{x}) = 1 \,\&\, y(\mathbf{x}') = 0],
\end{aligned}$$

where we used the fact that the probability density function

$$P[\hat{p}(\mathbf{x}') = t \mid y(\mathbf{x}') = 0] =: f(t)$$

is the derivative with respect to t of the cumulative distribution function

$$P[\hat{p}(\mathbf{x}') \le t \mid y(\mathbf{x}') = 0] = 1 - F(t).$$

So, given a randomly chosen observation $\mathbf{x}$ belonging to *class 1*, and a randomly chosen observation $\mathbf{x}'$ belonging to *class 0*, the AUC is the probability that the evaluated classification algorithm will assign a higher score to $\mathbf{x}$ than to $\mathbf{x}'$, i.e., the conditional probability of $\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{x}')$.

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.

For degree-d polynomials, the polynomial kernel is defined as[2]

$$K(x,y) = (x^{\mathsf{T}} y + c)^d$$

where x and y are vectors in the *input space*, i.e. vectors of features computed from training or test samples and $c \ge 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial. When $c = 0$, the kernel is called homogeneous.[3] (A further generalized polykernel divides $x^{\mathsf{T}}y$ by a user-specified scalar parameter a.[4])

As a kernel, K corresponds to an inner product in a feature space based on some mapping φ:

$$K(x,y) = \langle \varphi(x), \varphi(y) \rangle$$

The nature of φ can be seen from an example. Let $d = 2$, so we get the special case of the quadratic kernel. After using the multinomial theorem (twice—the outermost application is the binomial theorem) and regrouping,

$$K(x,y) = \left(\sum_{i=1}^{n} x_i y_i + c\right)^2 = \sum_{i=1}^{n} (x_i^2)(y_i^2) + \sum_{i=2}^{n}\sum_{j=1}^{i-1} (\sqrt{2} x_i x_j)(\sqrt{2} y_i y_j) + \sum_{i=1}^{n} (\sqrt{2c}\, x_i)(\sqrt{2c}\, y_i) + c^2$$

From this it follows that the feature map is given by:

$$\varphi(x) = (x_n^2, \ldots, x_1^2, \sqrt{2} x_n x_{n-1}, \ldots, \sqrt{2} x_n x_1, \sqrt{2} x_{n-1} x_{n-2}, \ldots, \sqrt{2} x_{n-1} x_1, \ldots, \sqrt{2} x_2 x_1, \sqrt{2c}\, x_n, \ldots, \sqrt{2c}\, x_1, c)$$

Listing 1: A CNN model defined in Keras

```
model = Sequential()

w, h = 3, 3
sh = (64, 64, 3)

model.add(Convolution2D(32, w, h, input_shape=sh, border_mode='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Activation('relu'))

model.add(Convolution2D(32 w, h, border_mode='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Activation('relu'))

model.add(Convolution2D(48, w, h, border_mode='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Activation('relu'))

model.add(Convolution2D(48, w, h, border_mode='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Activation('relu'))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))

model.add(Dense(10, activation = 'softmax'))
```