HW2 Solution sketch

Jonatan Kilhamn

February 2017

1 Theoretical problems

1.1 Bayes classifier

The Bayes classifier posterior:

$$P(c_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{c}) = \frac{P(\mathbf{x}_{\text{new}}|c_{\text{new}} = k, \mathbf{X}, \mathbf{c})P(c_{\text{new}} = k)}{\sum_j P(\mathbf{x}_{\text{new}}|c_{\text{new}} = j, \mathbf{X}, \mathbf{c})P(c_{\text{new}} = j)}$$

(The sum in the denominator ranges over classes, of which there are only 2 in our case.)

The Naive Bayes also assumes:

$$P(\mathbf{x}_{\text{new}}|c_{\text{new}}, \mathbf{X}, \mathbf{c}) = \prod_{d=1}^{D} P((\mathbf{x}_{\text{new}})_d|c_{\text{new}} = k, \mathbf{X}, \mathbf{c})$$

where the d are the dimensions, in this case {rich, married, healthy} indexed $\{1, 2, 3\}$.

The prior distribution can be assumed to be uniform: $P(c_{\text{new}} = k) = 1/2$ i.e. with no other knowledge, a person is as likely to be content as discontent.

Plugging in the numbers, the probability sought in 1.1a) falls out as 2/3.

In b), simply dropping all factors corresponding to “healthy” gives the right answer. Formally, this can be seen by splitting the likelihood into a sum of two cases:

$$P([0, 1, *]|c_{\text{new}} = 1, \mathbf{X}, \mathbf{c}) = P([0, 1, 0]|c_{\text{new}} = 1, \mathbf{X}) + P([0, 1, 1]|c_{\text{new}} = 1, \mathbf{X}, \mathbf{c})$$

Then we state the sought probability as

$$P(c_{\text{new}} = k|\mathbf{x}_{\text{new}} = [0, 1, *], \mathbf{X}, \mathbf{c}) = \frac{P([0, 1, *]|c_{\text{new}} = k, \mathbf{X}, \mathbf{c})P(c_{\text{new}} = k)}{\sum_j P([0, 1, *]|c_{\text{new}} = j, \mathbf{X}, \mathbf{c})P(c_{\text{new}} = j)}$$

Combining this with the previous equation gives the same expression as the informal “drop every x_3 factor”, and the final answer becomes 2/5.

1.2 Extending Naive Bayes

Naive Bayes will have trouble here because it assumes independence of features, but features 1—3 are heavily dependent as exactly 1 of them must be true at any given time.

One way around this is to combine features 1—3 into one feature x_1', and then feed the classifier a new feature vector $[x_1', x_4]$.

2 Practical problems

2.1 Bayes classifier

"Both classes have equal prior" means that we have

$$P(y_{\text{new}} = 1) = P(y_{\text{new}} = 0) = \frac{1}{2}$$

Together with the expression for the dependence on the training data, we get:

$$P(y_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{y}) = \frac{P(\mathbf{x}_{\text{new}} | \hat{\mu}_k, \hat{\sigma}_k^2)}{\sum_j P(\mathbf{x}_{\text{new}} | \hat{\mu}_j, \hat{\sigma}_j^2)}$$

where, again, the sum in the denominator ranges over all classes, i.e. $j = \{1, 2\}$ corresponding to labels $\{+1, -1\}$.

Code and results

The cross-validation is performed like so: the data is randomly split into 5 partitions of equal size. One partition is held aside while the remaining 4 are used to train the classifier. Then the classifier is tested on the set-aside partition. The number of mis-classifications are recorded (the error rate). After repeating this process for all 5 partitions, the average error rate is reported.

The cross-validation here should give an average error rate of 0%. This data set is separated into two classes in a way that's very clear to the given classifier.

2.2 Handwritten digit recognition

A correct solution to this exercise is not provided at this time. The cross-validation average results should be somewhere around 6% error using the original feature vector (256 pixel values) and around 23% error using the alternative feature vector (32 variance values, one for each row and column).

For my grading of the optional task d), see below.

Comments

This is not a complete solution; only a sketch. If students were to hand in a report with as many steps omitted as this one, they would not get full marks.

The purpose of this document is to show students whether their solution was along the right lines. Firstly, this allows them to improve their understanding, if they had not solved the problem correctly; secondly, it gives a hint as to whether they can expect to score points for each task. (Of course, eventually each submission will be graded, and the actual number of points reported to the student.)

2.3 Jonatan's grading policy

If a student submission (a) explains its reasoning and (b) is correct, it will receive full marks.

A correct answer presented poorly can be grounds for deductions, but never for more than half of the point total on the task.

If the answer is incorrect, at least 1 point will be deducted. I will try to follow the reasoning and find where the mistake occurred. Depending on how severe the mistake was, and how good the reasoning was before and after the mistake(s), the deduction might be only the 1 point, or anything up to all the points for that task.

For tasks with more than one sub-problem (a, b, c...) my policy is that a correct answer to one sub-problem guarantees at least 1 point, but an incorrect answer to one of them rules out getting the maximum points for the task. In between those two extremes, each sub-problem does not correspond to an exact number of points; rather, the solution is considered as a whole.

2.3.1 Code

I do not concern myself very much with coding style. If (1) the answers in the report are correct, (2) the code works when I run it, and (3) I don't have reason to suspect that code has been copied, then the quality of the code won't further affect the grade.

If there is a mistake or incorrect answer, clear code might make it easy for me to realise that it was only a typo, and deduct fewer points.

If the answers are correct but I can't see how the code provides them (e.g. it won't run), I will deduct up to half of the task's points.

2.3.2 Optional tasks

An optional sub-problem (e.g. 2.2d in HW2) will count for 1 point if it is well-answered. However, the number of points available on a task is still capped at the number it says in the assignment.

In effect, this allows someone to make a small mistake on 2.2 that would take away 1 point, but recovering that point by answering the optional 2.2d.

No matter what, 2.2 will only give a maximum of 5p, so this “extra point” will not help towards other tasks.