

Machine Learning: Homework 5

Laurent HAYEZ

November 6, 2016

Exercise 1. After 931.554 seconds, Han won a 10 round race against Luke (who finished 2.802 seconds later) and Nien (who finished with an additional 0.928 seconds). The two losers claim that there is no statistical significant difference between Han and them at the 5% level and demand a rematch. Are they correct and should Han race once more against both of them? In the following table you find the individual lap times in seconds. Use the Student's t -test to check the difference between Han-Luke, Han-Nien, and Luke-Nien. Note your observations. You can use a spreadsheet program but without the built-in t -test formula.

Solution. In Table 1 (from file `ML_hayez1_serie5.xlsx`), we computed the different steps to achieve a Student's t -test between the different samples. First, we computed $Han_i - Luke_i$, $Han_i - Nien_i$ and $Luke_i - Nien_i$ for all $i \in \{1, \dots, 10\}$. Then for each of these columns we computed the mean $\overline{x_d}$ and we computed

$$\widehat{\sigma}_d = \sqrt{\frac{\sum_{i=1}^k d_i^2 - \frac{(\sum_{i=1}^k d_i)^2}{k}}{k-1}}.$$

With these values, we computed

$$t_{obs} = \frac{\overline{x_d}}{\frac{\widehat{\sigma}_d}{\sqrt{k}}}.$$

As $t_{lim} = 2.262$ when $\alpha = 5\%$, the t_{obs} for Han-Luke is below the interval of confidence, hence there is a significant difference so Han should not rematch Luke. For the other two differences, we find that there is no significant difference, so Han should rematch Nien and Luke should rematch Nien. \square

Table 1: Table for exercise 1

Lap	Han	Luke	Nien	Han-Luke	Han-Nien	Luke-Nien
1	92.704	93.44	93.866	-0.736	-1.162	-0.426
2	92.8	93.266	92.7	-0.466	0.1	0.566
3	92.757	93.231	92.987	-0.474	-0.23	0.244
4	93.863	93.335	93.152	0.528	0.711	0.183
5	93.001	93.449	93.307	-0.448	-0.306	0.142
6	93.026	93.468	94.237	-0.442	-1.211	-0.769
7	93.052	93.425	93.771	-0.373	-0.719	-0.346
8	93.185	93.367	93.441	-0.182	-0.256	-0.074
9	93.309	93.475	93.125	-0.166	0.184	0.35
10	93.857	93.9	94.698	-0.043	-0.841	-0.798
Sum	931.554	934.356	935.284	-2.802	-3.73	-0.928
$\overline{x_d}$				-0.2802	-0.373	-0.0928
$\widehat{\sigma_d}$				0.34559411	0.615644739	0.471965112
t_{obs}				-2.563904229	-1.915925684	-0.621781906
Accept?				REJECT	ACCEPT	ACCEPT

Exercise 2. *The main goal of a given system is to retrieve only the very cute photos of cats out of the total 54 cat pictures. We know that there are 10 images in the collection that are very cute and the rest is just cute. The system manages to return 8 very cute cat pictures but also retrieves 10 just cute images. Create the confusion matrix and calculate the overall precision, recall, and F1 of this system.*

The 18 retrieved pictures are returned in the following order (the leftmost image is at the first rank; VC = very cute; JC = just cute): VC, VC, JC, VC, JC, VC, VC, VC, VC, JC, JC, JC, JC, JC, JC, JC, VC, JC, JC

Calculate precision and recall at each rank (so, first with “VC”, then “VC, VC”, then “VC, VC, JC”, and so on). You don’t need to create the confusion matrix each time.

Bonus: There is a balance point when precision and recall are the same. Must there always be such a point between precision and recall? What is the relationship between the F1 value and this balance-point? (+2p)

Solution. The confusion matrix is represented in Table 2. The system classifies 8 out of 10 very cute cat pictures as very cute (True Positive), hence the remaining two pictures are classified as just cute (False Negative). It also classifies 10 just cute cat pictures as very cute (False Positive), hence the 34 remaining pictures are classified as just cure (True Negative).

$$\text{Overall precision} = \frac{8}{8+10} = 0.444,$$

$$\text{Recall} = \frac{8}{8+2} = 0.8,$$

$$F_1 = \frac{2 \cdot 0.8 \cdot 0.444}{0.8 + 0.444} = 0.571.$$

To compute the precision and recall at each rank, we assume that if we have i pictures returned as very cute, the rest of the pictures is classified as just cute (don’t know how to classify otherwise). We obtain Table 3.

We have that overall precision = $\frac{TP}{TP+FP}$ and recall = $\frac{TP}{TP+FN}$, hence we have that overall precision = recall $\iff FP = FN$. If we use a cumulative method as the one we used here, there will always be a point when $FP = FN$, and at this point, we will have overall precision = recall.

Furthermore, we have $F_1 = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}$. Expanding this expressions with the formulas for recall and precision, we obtain

$$F_1 = F_1(FP, FN) = \frac{2TP}{(TP+FP)(TP+FN)}$$

(viewed as a function of FP and FN). We can compute the gradient of this function

$$\nabla F_1(FP, FN) = \frac{2TP}{(TP + FP)(TP + FN)} \begin{pmatrix} \frac{1}{(TP+FP)} \\ \frac{1}{(TP+FN)} \end{pmatrix}.$$

Now, the function has a min or a max when $\nabla F_1 = 0$. Here we assume that $TP > 0$, hence $\nabla F_1 > 0$. Hence F_1 will be maximal when $\|\nabla F_1\|$ is minimal, and this is the case when $\left\| \begin{pmatrix} \frac{1}{(TP+FP)} \\ \frac{1}{(TP+FN)} \end{pmatrix} \right\|$. We have

$$\left\| \begin{pmatrix} \frac{1}{(TP+FP)} \\ \frac{1}{(TP+FN)} \end{pmatrix} \right\| = \sqrt{\frac{1}{FP^2 + FN^2 + 2TP(FP + FN) + 2TP^2}}.$$

Hence we deduce that the norm is minimal when $FP+FN$ is minimal and FP^2+FN^2 is also minimal. In other words, F_1 is maximal when both $FP+FN$ and FP^2+FN^2 are minimal. This also means that at this point, the sum between the precision and the recall is maximal, which is close to the balance point between the precision and the recall. \square

Table 2: Confusion matrix for Exercise 2

		Predicted class	
		Very Cute	Just Cute
True state of nature	Very Cute	8	2
	Just Cute	10	34

Table 3: Precision and recall at each rank

Pictures returned	True Positive	False Positive	False Negative	True Negative	Overall Precision	Recall	F_1
VC	1	0	9	44	1.000	0.100	0.182
VC	2	0	8	44	1.000	0.200	0.333
JC	2	1	8	43	0.667	0.200	0.308
VC	3	1	7	43	0.750	0.300	0.429
JC	3	2	7	42	0.600	0.300	0.400
VC	4	2	6	42	0.667	0.400	0.500
VC	5	2	5	42	0.714	0.500	0.588
VC	6	2	4	42	0.750	0.600	0.667
VC	7	2	3	42	0.778	0.700	0.737
JC	7	3	3	41	0.700	0.700	0.700
JC	7	4	3	40	0.636	0.700	0.667
JC	7	5	3	39	0.583	0.700	0.636
JC	7	6	3	38	0.538	0.700	0.609
JC	7	7	3	37	0.500	0.700	0.583
JC	7	8	3	36	0.467	0.700	0.560
JC	7	9	3	35	0.438	0.700	0.538
VC	8	9	2	35	0.471	0.800	0.593
JC	8	10	2	34	0.444	0.800	0.571