Completed

Principles of Machine Learning

Exercises

Victor Verreet *victor.verreet@cs.kuleuven.be*
Laurens Devos *laurens.devos@cs.kuleuven.be*

Fall, 2021

Exercise Session 2: Methodology and Ensembles

2.1 Evaluation Metrics and Confusion Matrices

Here is an overview of the evaluation metrics as discussed in the methodology lecture:

		predicted		
		+	-	
actual	+	TP	FN	TPR $= \frac{\text{TP}}{\text{TP+FN}} =$ recall $=$ sensitivity
	-	FP	TN	FPR $= \frac{\text{FP}}{\text{FP+TN}}$
		Precision $= \frac{\text{TP}}{\text{TP+FP}}$		

a. Give an intuitive interpretation for precision, recall, and false positive rate (FPR).

b. Construct the confusion matrix for the results in table (a) below. y is the actual label, $\hat{y}$ is the predicted label (using probability threshold 0.5), $\hat{p}(y = 1)$ is the estimated probability of the label being true.

(a)

i	y	$\hat{y}$	$\hat{p}(y = 1)$
0	0	0	0.11
1	0	1	0.56
2	1	0	0.45
3	0	0	0.01
4	0	0	0.09
5	1	1	0.86
6	1	1	0.95
7	0	0	0.13
8	1	1	0.79
9	0	1	0.51

(b)

i	y	$\hat{y}$	$\hat{p}(y = 1)$
3	0	0	0.01
4	0	0	0.09
0	0	0	0.11
7	0	0	0.13
2	1	0	0.45
9	0	1	0.51
1	0	1	0.56
8	1	1	0.79
5	1	1	0.86
6	1	1	0.95

c. Compute the accuracy and the error for the results in the table above. Write the accuracy in terms of TP, FN, FP, and TN.

d. Draw the ROC curve. Table (b) already has the instances sorted by predicted probability value. Use the thresholds indicated by the horizontal lines (i.e., you only need to consider three confusion

matrices). Interpret these values. Which classifier would you prefer? Are any of these three points in the ROC curve suboptimal?

e. Draw the precision-recall curve.

f. For a given dataset and a precision-recall curve, show that there is exactly one corresponding ROC curve.

2.2 Programming Exercise: K-Fold Cross Validation

You can find a Python script `kfold.py` on Toledo.

- Which model should you pick and why?
- Which model of the two plotted should you pick and why?

2.3 Boosted Decision Trees

When learning a decision tree from a set of instances $I = \{(x_i, y_i) \,|\, i = 1...n\}$ with k target classes $y \in \{T_i \,|\, i = 1...k\}$ the ID3 algorithm typically calculates which attribute to split on using the formulas

$$CE(X) = -\sum_T \frac{|X[y = T]|}{|X|} \log\left(\frac{|X[y = T]|}{|X|}\right)$$

to find the entropy of a set of instances X and then

$$IG(X, \alpha) = CE(X) - \sum_V \frac{|X[\alpha = V]|}{|X|} CE(X[\alpha = V])$$

to find the information gain IG for an attribute α. The sum runs over all values V of α.

How should these formulas be changed when we give each instance (x_i, y_i) a weight $w_i \geq 0$, as is done in boosting algorithms?

2.4 Parallelizable Algorithms

Which of the following procedures are parallelizable? Why?

- Training an ensemble classifier with bagging
- Training an ensemble classifier with boosting
- Prediction for an ensemble classifier with bagging
- Prediction for an ensemble classifier with boosting

2.5 Combining Classifications

In the table below, some instances are given and the predicted probabilities of it belonging to the positive class, as predicted by different binary classifiers. The true class of the instance is also listed. We will now look at some methods to combine the classifiers for an ensemble prediction.

The table shows the predicted probability of classifier C for instance I to be positive, alongside the true instance class T.

	C_1	C_2	C_3	T
I_1	0.3	0.8	0.7	$+$
I_2	0.4	0.4	0.9	$+$
I_3	0.2	0.6	0.4	$-$

a. Assume a classifier predicts the positive class if the probability is higher than 1/2. Calculate the majority vote for the ensemble prediction giving each classifier an equal weight.

b. Now calculate a weighted vote prediction. Use as weights $|p-1/2|$, with p the predicted probability for the instance to be positive. This quantity measures the certainty of the prediction.

c. Use a Bayesian estimate where the probability for each classifier is given by its accuracy, normalized over all the accuracies.

What are some noticeable results about the different ways to predict in terms of accuracy?

2.6 AdaBoost: Determining Weights α_k

The table below shows which of the $n = 3$ instances of a binary classification problem are correctly classified by each weak classifier h_k.

	h_1	h_2	h_3	h_4
$\boldsymbol{x}_1$	✓	✓	·	✓
$\boldsymbol{x}_2$	·	✓	·	·
$\boldsymbol{x}_3$	✓	·	✓	✓

The ensemble prediction is given by

$$F_4(\boldsymbol{x}) = \sum_{k=1}^{4} \alpha_k h_k(\boldsymbol{x}) \in [-1, 1],$$

where k runs over all four classifiers and $\boldsymbol{x}$ is an instance.

The instance weights $w_i^{(1)}$ are initialized to $1/n$, with n the number of instances. Iteratively compute the weights α_k for $k = 1, 2, 3, 4$ such that the error

$$E_k = \sum_{i=1}^{n} \exp(-y_i F_4(\boldsymbol{x}_i))$$

is minimized.

a) Precision → If a model predicts n true label, how many are actually true?

Recall → If there are m true label, how many the model predicts true?

FPR → If there are k false labels, how many are incorrectly predicted?

b)

actual \ Predicted	1	0
1	3	1
0	2	4

$$\therefore \frac{TP+TN}{TP+TN+FP+FN} = \frac{7}{10} = 0.7$$

b) $FPR_1 = \frac{FP}{FP+TN} = \frac{2}{6} = \frac{1}{3} = 0.33$

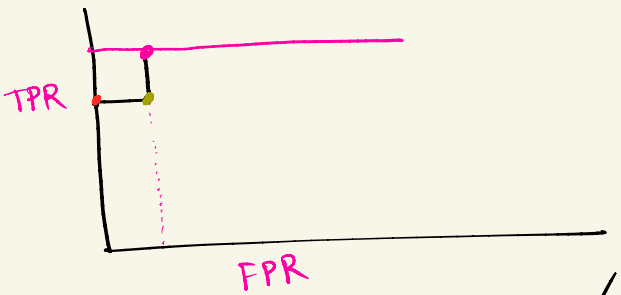
$TPR_1 = \frac{TP}{TP+FN} = \frac{4}{4} = 1$

$FPR_2 = \frac{2}{6} = \frac{1}{3} = 0.33$

$TPR_2 = \frac{3}{4} = 0.75$

$FPR_3 = 0$

$TPR_3 = 0.75$



Red if we want no false positives
yellow not an optimal model.
Pink if TPR is more imp

	1	0
1	4	0
0	2	4

$\therefore P_1 = \frac{4}{6} = \frac{2}{3}$

$R_1 = 1$

	1	0
1	3	1
0	2	4

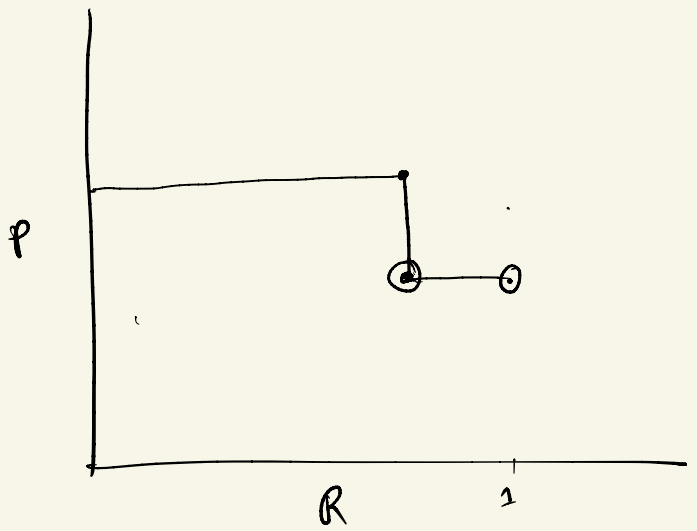
$P_2 = 0.67$

$R_2 = 0.75$

	1	0
1	3	1
0	0	6

$P_3 = 1$

$R_3 = 0.75$



Let a dataset is given, then $TN+FP = N_n$

$TP+FN = N_p$

Let (P, r) be a pt. in PR curve.

Then $P = \frac{TP}{TP+FP} \Rightarrow \frac{FP}{TP} + 1 = \frac{1}{P}$

$$\Rightarrow \frac{FP}{TP} = \left(\frac{1}{P} - 1\right) = \frac{1-P}{P}$$

$$\Rightarrow FP = TP\left(\frac{1-P}{P}\right)$$

$$= N_p r\left(\frac{1-P}{P}\right)$$

$$FPR = \frac{FP}{N_n} = \frac{N_p}{N_n} \cdot r\left(\frac{1-P}{P}\right)$$

$$CE(x) = -\sum p_i \log p_i$$

$$p_i = \frac{\sum_{j=1}^{n} w_j I[x_j | y = T_i]}{\sum_{j=1}^{n} w_j} \qquad I[x_j | y = T_j] = \begin{cases} \end{cases}$$

2.4

yes, Training with ensemble clarifier with bagging, since every model is independent of each other.

No, Training with ensemble clarifier with boosting isn't parallelizable since at time step t, model M_t requires the performance of M_{t-1}.

yes to both.

2.5

	c_1	c_2	c_3	$\hat{y}$
I_1	0	1	1	1
I_2	0	0	1	0
I_3	0	1	0	0

b) $f(1) = -0.2 \times 0.3 + 0.3 \times 0.8 + 0.2 \times 0.7$

$= -0.06 + 0.24 + 0.14$

$= 0.32 > 0$

$f(1) : 1$

$\therefore f(2) = -0.1 \times 0.4 \times 2 + 0.4 \times 0.9$

$= 0.28$

$f(2) : 1$

$\therefore f(3) = -0.4 \times 0.2 + 0.1 \times 0.6 - 0.1 \times 0.4$

$= -0.06$

$f(3) : 0$

c) accuracy $\qquad \frac{2}{3} + 1 = \frac{5}{3}$

$c_1 = \frac{1}{3}$

$c_2 = \frac{1}{3}$

$c_3 = 1$

$\therefore g(I_1) = 0.3 \times \frac{1}{5} + 0.8 \times \frac{1}{5} + 0.7 \times \frac{3}{5}$

$= 0.06 + 0.16 + 0.42$

$= 0.64 > 0.5$

$g(I_1) : +$

$g(I_2) = 2 \times 0.4 \times \frac{1}{5} + 0.9 \times \frac{3}{5}$

$= 0.16 + 0.64$

$= 0.80 > 0.5$

$g(I_2) : +$

$g(I_3) = \frac{0.2}{5} + \frac{0.6}{5} + \frac{0.4 \times 3}{5}$

$= 0.04 + 0.12 + 0.24$

$= 0.40 < 0.5$

$g(I_3) : -$

2.6 $\quad w_i^0 = \frac{1}{n}$

For $t = 1 \dots n$

$\epsilon_t = \sum_{y_i \neq h_i(x)} w_i^0$

$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$

Learn a new hypothesis h_t

$w_i^t = w_i^{t-1} \cdot \beta_t I[y_i = h_t(x_i)]$

$\hat{w}_i^t = \frac{w_i^t}{\sum w_i^t}$

$\alpha_t = \ln \frac{1}{\beta_t}$

$t = 1$

$w^{(1)} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$

$\epsilon_t = \frac{1}{3}$

$\beta_t = \frac{1}{2}$

$w^{(2)} = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{6}\right) \times \frac{3}{2}$

$= \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)$

$\alpha_t = \ln 2 = 0.69$

$t = 2$

$w^{(2)} = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)$

$\epsilon_t = \frac{1}{4}$

$\alpha_t = \ln 3 = 1.10$

$\beta_t = \frac{\frac{1}{4}}{1 - \frac{1}{4}} = \frac{1}{3}$

$w^3 = \left(\frac{1}{12}, \frac{1}{6}, \frac{1}{4}\right) \times 2 = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\right)$

$\frac{2}{6} + \frac{1}{3}$

$\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$

$\frac{1}{12} + \frac{2}{12} + \frac{3}{12}$

$-$

t=3

$\epsilon_t = \frac{1}{2}$

$\beta_t = 1$

$\omega_i^4 = \omega^3$

$\alpha_t = 0$

t=4

$\epsilon_t = \frac{1}{3}$

$\beta_t = \frac{1}{2}$

$\omega^5 = \left(\frac{1}{12}, \frac{1}{6}, \frac{1}{4}\right) \times 2 = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\right)$

$\alpha_t = \ln 2 = 0.69$

$$\omega_i^{t+1} = \begin{cases} \omega_i^t \cdot e^{-\alpha_i} & \text{if } y_i = h_t(x_i) \\ = \omega_i^t e^{-\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}} \\ = \omega_i^t \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \\ \omega_i^t e^{\alpha_i} & \text{if } y_i \neq h_t(x_i) \\ \omega_i^t e^{\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}} \\ \omega_i^t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \end{cases}$$

$$\hat{\omega}_i^{t+1} = \frac{\omega_i^t \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{\sum_c \omega_i^t \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \sum_{ic} \omega_i^t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}$$

$$= \frac{\omega_i^t \cdot \frac{\epsilon_t}{1-\epsilon_t}}{\sum_c \omega_i^t \frac{\epsilon_t}{1-\epsilon_t} + \sum_{ic} \omega_i^t}$$

$$= \frac{\omega_i^t \beta_t}{\sum_c \omega_i^t \beta_t + \sum_{ic} \omega_i^t}$$

$$\therefore \omega_i^{t+1} = \frac{\omega_i^t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{\sum \omega_i^t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + \sum \omega_i^t e^{\frac{\epsilon_t}{\sqrt{1-\epsilon_t}}}}$$

$$= \frac{\omega_i^t}{\sum \omega_i^t \beta_t + \sum \omega_i^t \beta_t}$$