

MLEARN	<i>Tuesday, January 18, 2022</i>
Professors: Eduard Garcia, David Remondo, Josep Vidal, Veronica Vilaplana Duration of exam: 2h 30min Solve the exam in these sheets, do not deliver additional ones	

Exercise 1 (2 points)**Ensembles**

- a) If you have trained five different models on the exact same training data, and they all achieve 95% precision, is there any chance that you can combine these models to get better results? If so, how? If not, why?
- b) What is the difference between hard and soft voting classifiers?
- c) Explain similarities and differences between Random Forest Classifiers and Bagging Decision Trees?
- d) Which of the following is / are true about weak learners used in boosting methods?
1. They have low variance and they don't usually overfit
 2. They have high bias, so they cannot solve hard learning problems
 3. They have high variance and they don't usually overfit
- e) For binary classification, which of the following statements is / are true about AdaBoost?
1. It can be applied to neural networks
 2. It uses the majority vote of learners to predict the class of a data point
 3. The metalearner provides not just a classification but also an estimate of the posterior probability

Answers:

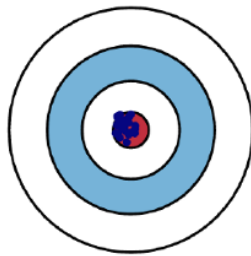
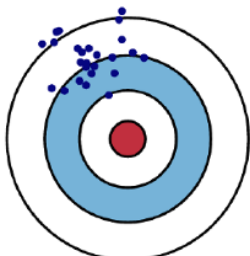
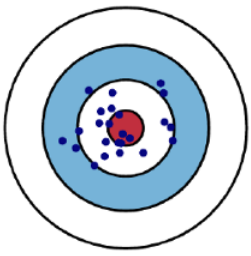
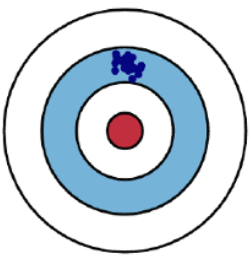
- a) Using a voting ensemble. It works better if models are different (SVM, NN, Decision Tree....)
- b) Hard voting: counts votes and picks the class that gets most votes. Soft voting: computes average estimated class probabilities, and picks the class with highest probability. Works for models that are able to estimate class probabilities.
- c) Similarities: the two are ensemble models; use decision trees as weak learners; each tree in the ensemble is built from a sample drawn with replacement from the training set.
Difference: In RF typically a random subset of features is used when splitting each node instead of using all features as Bagging Decision Trees.
- d) (1) and (2)
- e) (1) and (3)

Exercise 2 (1.5 points)**Bias-variance decomposition**

a) To understand bias and variance, we will create a graphical visualization using a bulls-eye. Imagine that the center of the target is our true model (a model that perfectly predicts the correct values).

As we move away from the bulls-eye, our predictions get worse and worse. Imagine we can repeat our entire model building process to get a number of separate hits on the target. Each hit represents an individual realization of our model, given the chance variability in the training data we gather.

Sometimes we will get a good distribution of training data so we predict very well and we are close to the bulls-eye, while sometimes our training data might be full of outliers or non-standard values resulting in poorer predictions. Consider these four different realizations resulting from a scatter of hits on the target. Characterize the bias and variance of the estimates of the following models on the data with respect to the true model as low or high by circling the appropriate entries below each diagram.

1	2	3	4
			
Bias Low / High Variance Low / High	Bias Low / High Variance Low / High	Bias Low / High Variance Low / High	Bias Low / High Variance Low / High

(1) low bias, low variance, (2) high bias, high variance, (3) low bias, high variance, (4) high bias, low variance

b) Explain what effect will the following operations have on the bias and variance of your model. Fill in with 'increases', 'decreases' or 'no change' in each of the cells

	Bias	Variance
Increasing k in K-nearest neighbor models	increases	decreases
Pruning a Decision Tree to a certain depth	increases	decreases
Removing all the non-support vectors in SVM	no change	no change
Increasing the number of hidden units in a Neural Network	decreases	increases
Using dropout to train a Deep Neural Network	increases	decreases
Regularizing the weights in a Neural Network	increases	decreases
Increasing the number of trees in a Random Forest	increases	decreases

Exercise 3 (1.5 points)**Decision trees**

To build a Decision Tree classifier, we have applied pre-pruning with different values of the minimum leaf size (the minimum number of training vectors associated with a leaf). In the figures below, we can see the resulting training and validation error for different values of the parameter.

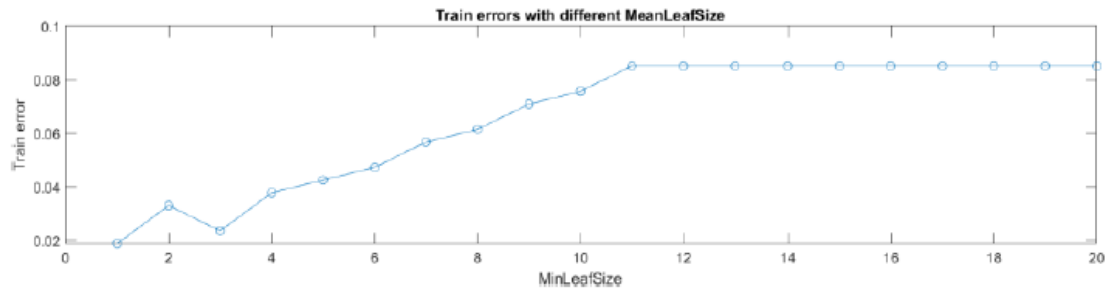


Figure 4 – Train errors curve for MinLeafSize between 1 and 20

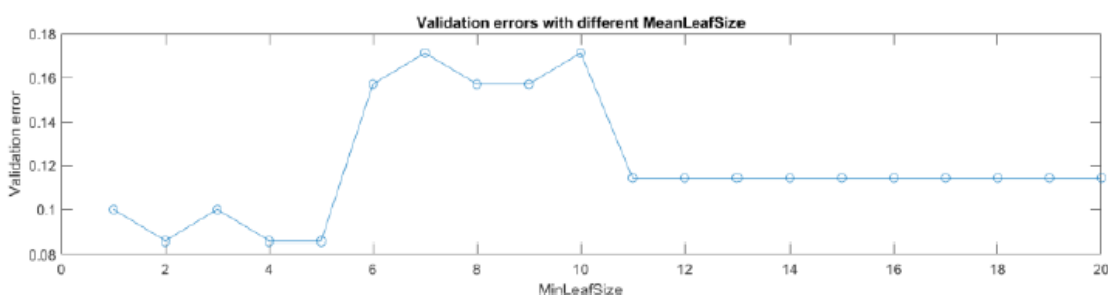


Figure 5 – Validations errors curve for MinLeafSize between 1 and 20

- a) Which of these trees can be expected to be the best classifier? Motivate your answer.

Now, if the confusion matrix for the best classifier is this (here columns represent ground truth and rows represent predicted values):

	Positive	Negative
Positive	1	399
Negative	1	4599

- b) Would you be satisfied with the results? Comment on the quality of the obtained model in terms of the relevant performance factors that you have seen during the course. Calculate all the performance factors that you need to support your discussion.

Answers:

- a) The classifier to be selected should be the one with the lowest validation error, no matter what the training error is. However, in this scenario we have three different candidates with identical validation error (MLS = 2, 4 and 5). Since the MLS parameter is inversely related to the expected number of comparisons that are done when using the model for classification of test data, the best option of the three will be the one with MLS = 5.
- b) The accuracy is pretty good ($= 4600/5000 = 0.92$). While recall/sensitivity is not too bad ($= 0.5$), the precision is very bad ($= 1/400 = 0.0025$). We can observe these discrepancies among performance measures also if we calculate the F-score ($= 2 \times 0.5 \times 0.0025 / (0.5 + 0.0025) = 0.005$). This is a typical situation in scenarios with unbalanced classes.

Exercise 4 (1.5 points)**Bayesian decisions**

These days, the SARS-CoV-2 disease (COVID-19) shows in Barcelona a prevalence of 2500 (14-day cumulative number of cases per 100,000 inhabitants); that is, 2.5 % of the population of Barcelona is currently infected. Before going to the campus, you take a rapid antigen test, just to be sure. Rapid antigen tests are known to have a low **sensitivity/recall** (near 70%) but a good **specificity** (close to 98%) and a good **precision**.

- a) With low sensitivity and high specificity, are we expecting a large number of false positives, false negatives, or both? Briefly explain the difference between those three metrics.

We are expecting a large number of false negatives and a low number of false positives. Precision is related to recall, specificity and prevalence.

Recall: $R = \Pr(x=1|\omega_1) = TP/(TP+FN)$

Specificity: $S = \Pr(x=0|\omega_2) = TN/(TN+FP)$

Precision: $P = \Pr(\omega_1 | x=1) = TP/(TP+FP)$

Assume ω_1 is the class applied to an infected person, and ω_2 the class for people without the disease; $x \in \{0,1\}$, where $x=0$ represents a negative test result, and $x=1$ a positive result. Your test was negative (yeaaaaah!)

- b) According to Bayes' theorem, what's the probability that you really don't have the disease, that is $\Pr(\omega_2 | x=0)$?

$$\Pr(\omega_2 | x=0) = 0.98*(1-0.025) / (0.98*(1-0.025) + (1-0.7)*0.025) = 0.992$$

- c) If the test was positive, apply MAP criterion to decide whether you would be infected (ω_1) or not (ω_2); that is, $\Pr(\omega_1 | x=1) > \Pr(\omega_2 | x=1)$?

$$\Pr(\omega_1 | x=1) = 0.7*0.025 / (0.7*0.025 + (1-0.98)*(1-0.025)) = 0.473$$

$$\Pr(\omega_2 | x=1) = 1 - \Pr(\omega_1 | x=1) = 0.527$$

$\Pr(\omega_2 | x=1) > \Pr(\omega_1 | x=1)$, so you're not infected

- d) If both sensitivity and specificity were 50%, what is the probability of being infected after a positive antigen test? And the probability of being infected after a negative test?

With 50%, tests are useless and both $\Pr(\omega_1 | x=1)$ and $\Pr(\omega_1 | x=0)$ are equal to the prevalence (Prior Probability) = 0.025

- e) Give values to π_{ij} so that the decision made by a Minimum Bayesian Risk (MBR) criterion differs from the MAP criterion in c). Assume that the cost of a correct decision is zero (i.e. $\pi_{11} = \pi_{22} = 0$).

$$C(\omega_1 | x=1) < C(\omega_2 | x=1)$$

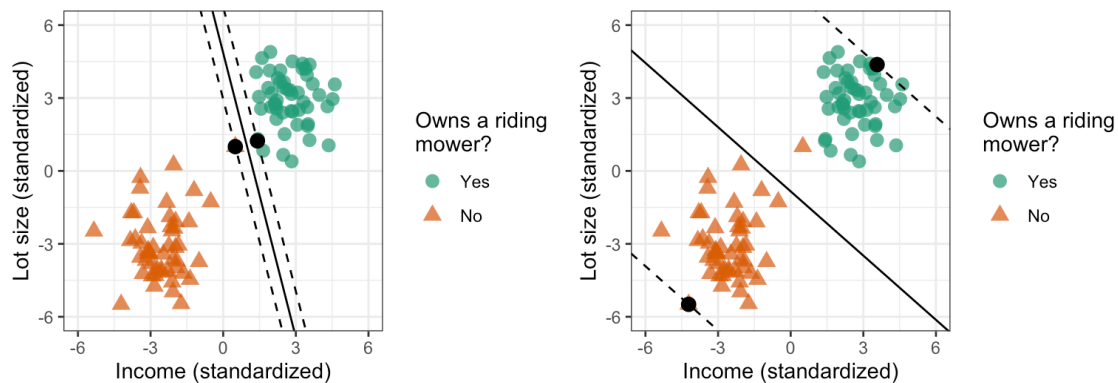
$$\pi_{12} \Pr(\omega_2 | x=1) < \pi_{21} \Pr(\omega_1 | x=1)$$

$$\pi_{12} / \pi_{21} < \Pr(\omega_1 | x=1) / \Pr(\omega_2 | x=1) = 0.473 / 0.527$$

$$\pi_{12} / \pi_{21} < 0.897$$

Exercise 5 (1.5 points)**Support vector machines**

The figures below show the training data of a two-feature problem. On the left-hand side, we can see the decision boundary obtained from the hard-margin SVM (method applicable when data are linearly separable without any feature space transformation); on the right-hand side, we find the result of a soft-margin SVM classifier, where a regularization term (the term P in the course slides) and slack variables ξ_i are used to allow for some data to lie on the wrong side of the corresponding margin boundary.



- If the number of training data samples is 100 (50 of each class), what would be the value of the training error in the first case? And in the second case? Justify your answers.
- Out of the triangle class data, and for the soft-margin case, explain how many samples would have a value of the slack variable ξ_i equal to zero. Also explain how many would have ξ_i equal to one? And how many larger than one? How do you represent the value of ξ_i on the graph for a given data sample in general?

Answers:

- In the first case, the training error would be 0, since no sample lies on the wrong side of the classification boundary. In the second case, it would be $1/100$, since there is one sample on the wrong side of the decision boundary.
- Only one sample has $\xi_i = 0$, since there is only one on the right side and on the boundary margin. All the other samples have $\xi_i > 0$ because they lie on the inner side of the margin boundary; there is only one with $\xi_i > 1$, which is the one on the wrong side of the decision boundary. In general, ξ_i is the distance from a sample to the corresponding boundary margin, if it lies on the inner side, and it will be zero if it lies on the outer side.

Exercise 6 (2 points)**Neural networks**

Briefly discuss the following aspects related to a multi-layer neural network (NN) used in a classification problem:

1. What is the use of the cost function? Describe a cost function for a NN in a classification problem.

When training a NN, the objective is to select those weights that minimize the cost function. The value of the cost function describes how well the NN performs given the training data. Two possible cost functions are the MSE:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^c (t_k - z_k(\mathbf{x}))^2$$

and the logloss (also called cross-entropy):

$$J(\mathbf{w}) = - \sum_{k=1}^c t_k \log z_k(\mathbf{x})$$

where z_k is the k -th output of the NN, \mathbf{x} is the feature vector and t_k is 1 if \mathbf{x} belongs to class k and zero otherwise.

2. What is the role of the activation function? Propose two examples.

The activation function converts the inputs into outputs in a non-linear way, so that the NN can learn complex functions. If the activation function were linear, the NN would behave as a linear classifier, learning a function which is a linear combination of its input data.

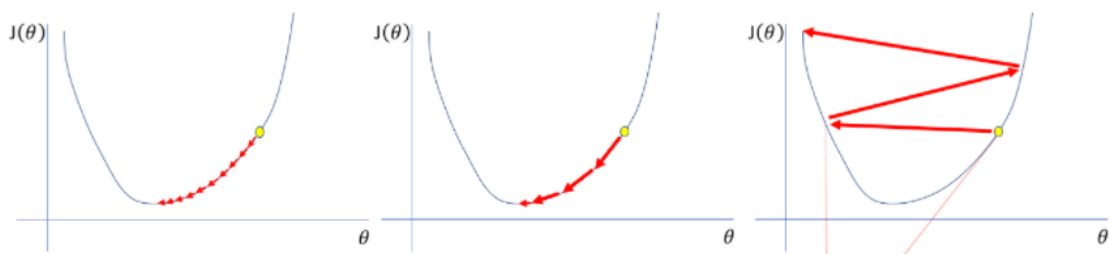
To generate an output, the activation function firstly calculates the weighted sum of inputs and further adds bias with it.

Some examples are: Sigmoid, tanh, ReLu, Leaky ReLu.

3. How to initialize weights when training a NN?

As a rule of thumb, the weights are assigned random values close to zero but not too small.

4. How learning rate affects the training of a NN in these three convergence plots (θ is one parameter of the NN)?



On the left plot, the learning rate parameter is set too low. Training of the model will continue slowly as we are making very small changes to the weights. It will take many iterations before reaching the point of minimum cost function.

On the right plot, the learning rate parameter is set too high, this causes divergent behavior to the cost function due to large changes in weights. It may fail to converge (the model can give a good output) or even diverge.

On the center plot, the learning rate parameter seems to have a proper value.

5. Explain one method to avoid overfitting in NN. What are the hyperparameters that need to be considered during training for a NN that use that overfitting avoidance method?

Some methods used to avoid overfitting:

Dropout: It randomly drops neurons from the neural network during training. Hyperparameters: number of neurons, number of layers, the learning rate, the fraction of weights that are set to zero.

Early stopping: After training with an increasing number of epochs, the model begins to overfit the training data. Early stopping refers to stopping the training process before that point, by checking the number of epochs in which the error in the validation data set starts increasing. Hyperparameters: number of neurons, number of layers, the learning rate, the number of epochs in training.

Regularisation: Add a penalty function to the cost function that depends on the NN weights (norm-2, norm-1, etc. of the weights vector), multiplied by a parameter λ . Hyperparameters: number of neurons, number of layers, λ , the learning rate, the type of norm itself.

6. What is the difference between Epoch, Batch, and Iteration in NN training?

Epoch: It represents the iterations over the entire training dataset.

Batch: This refers to the situation when we are not able to pass the entire dataset into the NN at once due high computational requirements, mainly because the training database is too large. Therefore, we divide the dataset into several batches and update the gradients in the backpropagation algorithm once per batch.

Iteration: It is an update of the NN weights using a gradient technique. If we have X samples as our training dataset and we choose a batch size of Y samples, then an epoch should run X/Y iterations.