Machine Learning

SDI/Metz, 2022-2023

written test, duration 2h

All paper documents allowed, electronic devices are prohibited. Never anwer by only "yes" or "no", you always have to justify your answers.

Answers can be written in French or English.
The exercices count for a total of 21 *points.*

1 The power-ball algorithm (10 point(s))

Let us consider a supervised learning problem. Input-label pairs are $(x_i, y_i) \in \mathbb{R}^d \times \mathcal{Y}$. We use the Euclidian distance d in $\mathbb{R}^d$. The power-ball algorithm is defined as algorithm 1.

Algorithm 1 power-ball

We use a constant parameter $\rho > 0$.
A dataset $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is given.
Let $L = [(x_1, y_1), \ldots, (x_N, y_N)]$ be the list of all the samples in S.
Let H be an empty list. // It will contain the centers of selected balls.
while $L \neq \emptyset$ **do**
    Select one (x, y) randomly from L.
    Remove all the (x', y') in L such as $\mathrm{d}(x', x) \leq \rho$. // those which are in a ball around x.
    Append (x, y) in H.
end while
Define h such as $h(x) \stackrel{\text{def}}{=} y'$ with $(x', y') \stackrel{\text{def}}{=} \mathrm{argmin}_{(x'', y'') \in H} \mathrm{d}(x'', x)$. // We find the closest ball in H and return its label.
return h.

Let us apply powerball to the following problem. Our dataset is an extension of the Iris dataset, containing $N = 2000$ samples. There are, for each sample, 4 size-related attributes expressed in centimeters : sepal length, sepal width, petal length, petal width. The fifth attribute is the kind of iris, which is Setosa, Versicolor or Virginica. There are approximately the same number of samples for each kind in the dataset.

We want to predict the kind of an iris from its petal and sepal dimensions. We preprocess the dataset by standardizing the 4 size-related attributes (i.e. 0 mean, variance 1).

▷ **Q1** : Would you say that powerball implements the ERM [1] induction principle ? *(1 point(s))*

Yes, somehow, since it relies on the dataset and tries to give the correct answer for the samples it sees. Nevertheless, if some samples are inside the ball around the samples used for setting up H, their label is not considered, so it is not minimizing strictly the empirical risk. Some student argued that powerball implements some transductive learning. This is not true, since once selected, the samples are embedded. A transductive learning would require for each prediction the full dataset. The support of SVMs are also embedded, as for powerball, but SVMs are not a case of transductive learning. Last, some student said that as it is transductive, this is not ERM. It is a nonsense. Transductive vs Inductive is the way you learn, ERM is what you learn. If you learn to fit a dataset with a transductive method, this is ERM. Some student said that powerball is a supervized algorithm, so it implements ERM. This is wrong. Supervized only means that we use labels for learning. It was actually ask to check that ERM is applied by our supervized algorithm. For example, SVMs do not stricly implement ERM, due to the regularization.

1. Empirical Risk Minimization

▷ **Q2** : Would it be relevant to use bagging with powerball ? If yes, say how to implement this, if no, justify why it would be a nonsense. *(1 point(s))*

Yes it is, since there is some randomness in powerball. A bad choice (i.e. a sample whose label is not representative of its neighborhood) can have dramatic effects on the performance. Applying several times powerball on the dataset, and then merge the results, will provide a much better predictor.

We set $\rho = 2$, and apply the powerball algorithm 1 to get a predictor from our dataset.

▷ **Q3** : We measure an empirical risk equals to 0.05. What does this value actually mean ? Is it a good fitting of the data ? *(1 point(s))*

We have a classification problem, so we may have used the binary loss. In this case, the risk is the ratio of badly labelled samples, i.e. 5% here. As the dataset is balanced*, this is a very good fitting.*

▷ **Q4** : What can you say about the real risk of the computed predictor ? *(1 point(s))*

The $\rho = 2$ setting is a large value, since the dimensions are standardized. Powerball can hardly overfit. The empirical risk may reflect the real risk in this case. Some students argued that we have a lot of samples, so the empirical risk becomes reliable. This is a false reasonning, many algorithms have the ability to overfit on large datasets.

▷ **Q5** : Why can't powerball be used as a weak predictor learner in the Adaboost [2] boosting process ? *(1 point(s))*

Some of the students have argued that the current problem is not bi-class, as opposed to what is presented for adaboost in the lecture. Yes, indeed... but the main issue for using Adaboost is the following. The weak predictor has to be computed from a weighted empirical risk minimization. Here, considering weights cannot be done directly.

▷ **Q6** : Propose an adaptation of powerball enabling its usage in Adaboost *(1 point(s))*

Instead of choosing randomly the samples for building H, we can bias the selection with the samples weightings. It may help powerball to make less errors on highly weighted samples. This enables to set up a bi-class boosted classifier from powerset. Then, we can use one-versus-one for multi-class.

▷ **Q7** : How would you find the best value for parameter ρ ? How will you estimate the real risk of the predictor learnt by powerball with this parameter ? *(3 point(s))*

I would keep a subset V of the samples apart, for validation. Then, with the remaining samples, I will perform a grid-search for tuning ρ, using a cross-validation in order to estimate the real risk. Once the best ρ is found, I will use its empirical risk of some corresponding h measured on V in order to estimate the real risk. Using several h may be a good idea, since the training process is not deterministic.

▷ **Q8** : Do you think that the optimal value of ρ is related to the way the inputs are distributed in $\mathbb{R}^4$? Is it related to the way the labels are associated to the inputs ? *(1 point(s))*

Both indeed. If inputs with the same labels are well clustered, a ρ close to this cluster size my be good. This argument is not true if there is no clustering and if labels overlap.

2 Taxi driver (6 point(s))

Let us consider a car doted with a GPS and a compass. The GPS provides 2 angles, the longitude and the latitude, and the compass the angle relative to the magnetic north. All angles are given in radians.

The car belongs to a taxi driver who works in the city of Metz exclusively. She records the 3 angles in a dataset every minute of the whole day, during a whole year. She wants to use vector quantization in order to analyse her activity from the dataset she has collected.

▷ **Q9** : Why could it be relevant to have recorded the compass information ? *(1 point(s))*

2. This is the boosting process presented in class.

There may have streets that she uses only in one of the two ways.

▷ **Q10** : As vector quantization algorithms rely on a distance, propose a suitable distance for comparing two samples in the dataset. Nota : with angles, 2π is very close to 0. *(1 point(s))*

The GPS angles (longitude and latitude (l, λ)), as the driver stays in Metz, can be considered as usual Euclidian 2D points. The cycling problem occurs with orientation θ. Moreover, the distance has to be weighted (with β), in order to compare orientation differences with position differences. So we can define $\mathrm{d}^2((l,\lambda,\theta)\,,\,(l',\lambda',\theta')) \stackrel{\text{def}}{=} (l-l')^2 + (\lambda-\lambda')^2 + \beta\left\|\mathrm{e}^{\mathrm{i}\theta} - \mathrm{e}^{\mathrm{i}\theta'}\right\|^2$.

Everywhere in the following, when we refer to k-means, **it is the online k-means seen in class**.

▷ **Q11** : Explicit the learning rule of the online k-means, taking into account the distance you have proposed. *(1 point(s))*

If (l, λ, θ) is the weight and (l', λ', θ') the input, we have to learn as an interpolation bewtween inout and weght. We consider the cycling for θ, leading to $(l,\lambda,\theta) \leftarrow \Big((1-\alpha)\lambda + \alpha\lambda', (1-\alpha)l + \alpha l', \arg\Big((1-\alpha)\mathrm{e}^{\mathrm{i}\theta} + \alpha\mathrm{e}^{\mathrm{i}\theta'}\Big)\Big)$.

▷ **Q12** : We apply a k-means, with $k = 100$, on the dataset. What problem will occur and how could you fix it ? *(1 point(s))*

When the driver do not work, the car is parked. This provides a lot of data at the places where the car is parked during the night, week-ends or holidays. Many of the prototypes will be concentrated on those areas. This can be fixed by removing those points from the dataset, since the driver can identify them.

We apply a k-means with $k = 1000$ on a dataset where the previous problem has been solved. Then, for each sample, we determine the two closest of the k prototypes, and we link them with an edge if they were not connected yet by some previous step.

▷ **Q13** : What is the effect of the distance you have proposed on that graph construction ? *(1 point(s))*

The importance of angles relatively to position is crucial. If it is two small, all positions in a street will be connected as a chain, regardless the orientation, while a suitable ratio may lead to the building of two chains, one for each driving direction. Moreover, if to points are close but not easily reachable by car (need to cross a river for example), they will still be linked in the graph.

▷ **Q14** : The driver would like to visualize the dataset on a screen. Which vector quantization method would you use for that, and what will you display for each prototype ? What will the whole display look like ? *(1 point(s))*

Use a 2D Kohonen map. Each prototype is the position and orientation of the car in the city, arranged so that close situations of the car are close on the map (i.e. close on the screen).

3 Analyzing the density of plankton from environmental measures (5 point(s))

In this exercice, we are interested in an ecological problem of predicting the density of phytoplankton (small organisms floating in the sea) from environmental measures. For solving this problem, we consider the SEANOE Long term ecological database from the LTER-Italy Northern Adriatic Sea site as described in the paper (Acri et al 2019).

The training data contains 1509 recordings, each with 13 columns :

— Longitude (degrees)
— Latitude (degrees)
— Depth (m) of the measure
— Temperature (°C)
— Salinity (dimensionless)
— pH (pH units)
— Dissolved oxygen concentration (ml/l)
— Ammonia (microMol)
— Nitrite (microMol)
— Phosphate (microMol)
— Silicate (microMol)
— Chlorophyll a concentration (ug/l)
— total concentration of phytoplankton (cell/ml) : Value to be predicted

The first samples of the dataset are shown in table 1. In the following, we denote by x_i the i-th sample. Note that the very last column is the density to predict. Considering only the first 12 columns, some statistics of the dataset are given in table 2.

Long	**Lat**	**Depth**	**Temp**	**Sal**	**pH**	**Oxyg**	**NH3**	**NO2**	**PO4**	**Si**	**Chla**	**Phyto**
12.48	45.3	1.0	15.8	33.7	8.30	5.83	3.64	0.23	0.26	5.29	2.60	205.
12.48	45.3	5.0	14.9	33.5	8.30	6.00	5.17	0.28	0.20	5.98	3.18	146.
12.48	45.3	10.	11.3	37.6	8.27	5.87	1.42	0.10	0.64	3.39	1.12	101.
12.67	45.3	0.5	17.3	33.8	8.22	6.13	3.60	0.25	0.23	2.16	1.38	679.
12.67	45.3	5.0	15.6	36.2	8.32	6.26	1.95	0.16	0.11	4.29	0.61	192.

TABLE 1 – Some samples of the SEANOE dataset.

	Long	**Lat**	**Depth**	**Temp**	**Sal**	**pH**	**Oxyg**	**NH3**	**NO2**	**PO4**	**Si**	**Chla**
mean	12.81	45.13	11.44	15.82	36.16	8.22	6.88	1.21	0.28	0.15	5.59	1.91
min	12.31	44.82	0.00	5.16	1.97	7.44	0.14	0.00	0.00	0.00	0.00	0.00
max	13.35	45.58	41.00	29.90	38.81	8.78	89.99	31.66	5.32	17.14	109.49	111.81

TABLE 2 – Some statistics of the SEANOE dataset over the first 12 columns. The mean, minimum and maximum are computed individually for every feature.

We first begin by exploring the data and perform a PCA.

▷ **Q15** : When performing a PCA, one computes the eigenvalues and eigenvectors of a matrix : which is this matrix ? what are its dimensions ? Can the computed eigenvalues be negative ? *(1 point(s))*

If we stack the centered vectors, ***excluding the variable to predict****, in the columns of a matrix* X*. This matrix has* 12 *rows and* 1509 *columns. The PCA computes the eigevenvalue decomposition of the* $X.X^T$ *matrix which is* 12×12*. Its eigenvalues are positive since,* $\forall x, x^T X X^T x = (X^T x)^T (X^T x) = \|X^T x\|^2 \geq 0$

Performing the PCA and projecting the data by keeping only the two most important components (associated with the two largest eigenvalues), we get the scatterplot depicted on figure 1. The components of the projection vectors are given in table 3

	Long	**Lat**	**Depth**	**Temp**	**Sal**	**pH**	**Oxyg**	**NH3**	**NO2**	**PO4**	**Si**	**Chla**
1st	0.003	-0.002	0.887	-0.194	0.14	-0.003	0.388	-0.009	0.	-0.003	0.018	-0.074
2nd	-0.006	-0.002	-0.271	0.042	-0.211	-0.001	0.714	0.077	0.018	0.013	0.591	0.125

TABLE 3 – The components of the projection vectors.

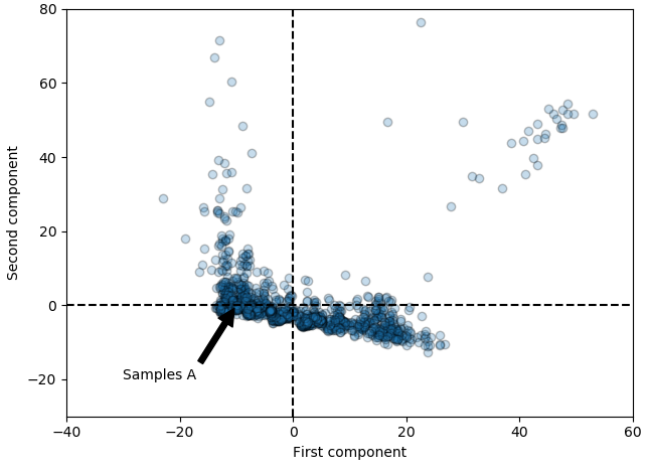

FIGURE 1 – Representation of the SEANOE dataset in the space projected by the PCA keeping the 2 most important components, associated with the 2 largest eigenvalues of the mystery matrix you must have found to answer one of the previous questions.

▷ **Q16** : Reasonning feature by feature (in univariate way), what does it mean for a sample to get projected with a negative first principal component ? Can you do the same reasonning for a null second principal component ? Can you then figure out which samples get projected around $(-10, 0)$ indicated as "Samples A" on the figure ? *(2 point(s))*

Samples A get a negative first component and a null second component. If we reason in a univariate way, projecting negativaly on the first component could be a :
- *depth lower than the mean depth,*
- *a temperature higher than the mean temperature,*
- *a salinity lower than the mean salinity,*
- *an oyxgen level lower than the mean oxygen*
- *whatever along the other dimensions which are associated with a low value component*

For the second component, being null mean we are on average for the Depth, Salinity, Oxygen, Si, and Chla. Therefore, these samples probably correspond to :
- *arbitrary distribution of latitude, longitude,*
- *an average or lower than the mean depth, ie around* 11.44 *or below*
- *a temperature higher than the mean, i.e. higher than* 15.82*,*
- *an average or lower than the mean salinity, i.e. around* 36.16 *or below*
- *an arbitrary pH,*
- *a lower than the mean oxygen, i.e. lower than* 6.88*,*
- *an arbitrary NH3, NO2, PO4,*
- *an average Si, i.e. around* 5.59
- *an average Chla, i.e. around* 1.91

Obviously, the things are in practice more complicated because positive contributions along some features might be compensated by negatrive contributions for other features. For example, a null second component can be obtained with a larger than the mean Oxygen and a larger than the mean Depth.

We now consider the question of solving the machine learning problem, i.e. finding a predictor for the concentration of phytoplankton.

▷ **Q17** : Which dimensionality reduction algorithm, of the embedded family, would you propose for performing feature selection on this problem ? Which hyperparameter(s) of this algorithm do you have to set in order to adjust the number of features involved in the decision ? *(1 point(s))*

For regression problems, some algorithms belonging to the embedded family are LASSO (linear regression with L1 penalty) and decision trees. For LASSO, the number of features can be adjusted with the L1 regularization coefficient. For trees, the number of features can be adjusted by either limiting the depth of the tree or limiting the total number of features involved in forming the decision.

▷ **Q18** : Describe with sufficient details a complete pipeline, **involving the feature selection algorithm**

proposed in the previous question, for solving our machine learning problem. By "with sufficient details", we mean you must be listing the learnable parameters. You must also specify the hyperparameters and the way you would operate for selecting the best set of hyperparameters. *(1 point(s))*

We proposed as feature selection algorithm a decision tree. Being an embedded approach, there is no need to follow the feature selection algorithm with another algorithm for the predictor; the optimized tree is our final predictor. The learnable parameters are the variables for the split at each node as well as the threshold for the decision. The hyperparameter to tune can be either the depth or the maximum number of variables to consider and these can be optimized by cross validation on the training data.
The risks can be evaluated with the RMSE.