(4)

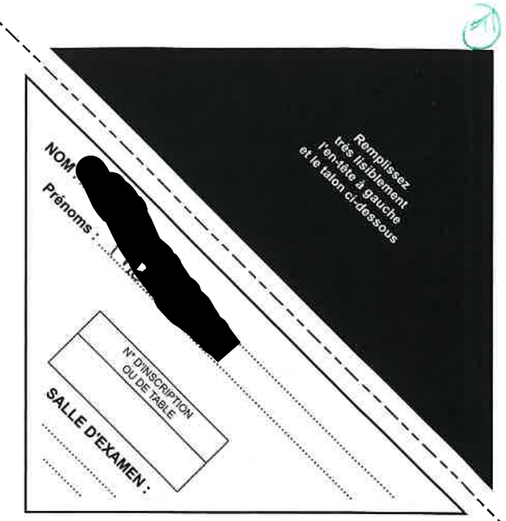
Perhaps we might see some cluster of positions. ~~ces~~
We can also ask for expert opinion to cluster the labels

ANNÉE : .....

UNITÉ D'ENSEIGNEMENT : .....

ÉPREUVE DE : .....

NOTE 18,25 /20	Coefficient	Note affectée du coefficient



Si votre composition comporte plusieurs feuilles, numérotez-les .../4

4,25

4) 4.1)

a) Goal: predict whether a mushroom is edible or poisonous to eat

b) Task: supervised, classification

c) Data structure: tabular data.
For example; the properties cap shape, color, ... as columns, one row = 1 mushroom. Target = to edible or poisonous

0,75

d) Algorithms: perceptron, decision tree, SVM

e) Unbalanced classes

4-2 :

a) Goal: predict whether team 1 or 2 ~~of~~ Pokemon will win a fight from their attributes

b) Task: supervised, ~~either~~ classification (win vs lose)

Notez avec exactitude votre numéro de table ou d'inscription.
Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

PARTIE À RABATTRE

4-2 :

c) Data structure : tabular. Attributes are the columns (Name, Hit Points, ...)
~~1row = 1 fb~~ For the input: 2 sets of features for Team 1 and Team 2.
Target : winner = Team 1 or winner = Team 2

d) Algorithm :
Decision Tree, Perceptron, SVM

1

e) We should shuffle data so that the same Pokemon on Team 1 also appear as Team 2. Otherwise, there may be bias. Also, composition of team should be different while splitting train and validation

4-3 :

a) Goal: predict future temperature for cities, from past record

b) Task :
Supervised, regression

1,25
good

c) Data structure
Tabular, time series
Columns: time, ~~fb~~ coordinates of city as input
Target : temperature

d) Algorithm
linear Regression, Polynomial regression

e) We need to be careful about splitting train and validation. Validation data should be posterior in time than training data
We can add features about the date, ~~expressing~~ expressing them in month of year for example, and season of year

4-4

a) Goal : ~~P~~
Predict a playing position of footballer from their attributes

b) Task :
Supervised, classification

1,25
≈
ok.

c) Data structure :
Tabular, attributes : weight, height, speed, ... as columns. ~~On~~
One row = 1 player
Target : playing positions

d) Algorithms :
Decision Tree, SVM, perceptron

e) We can start by using a dimensionality reduction technique and visualize the playing position (PCA or t SNE)

[7/75]

8

2 - 1

Linear Regression

Input : N datapoints, D - 1 dimension

We note $X = \{\vec{x_n}\} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_{D-1} \end{pmatrix}_n$ set of all input, with $\vec{x_n}$ the input at index n, augmented by 1.

$Y = \{y_n\}$, the regression target

$W = \{\vec{w}\}$, the model parameters of size $D = \begin{pmatrix} w_0 \\ w_2 \\ \vdots \\ w_{D-1} \end{pmatrix}$

the prediction of our model:

$\hat{y} = f_w(x) = W \cdot X = w_0 \cdot 1 + w_1 x_1 + \dots w_{D-1} x_{D-1}$

loss function least squares:

$J = \frac{1}{2N} \sum_n^N (\hat{y}_n - y_n)^2 = \frac{1}{2N} \sum_n^N (\vec{w} \cdot \vec{x_n} - y_n)^2$

2

From input X, we want to predict a scalar value Y

we train by minimizing this loss function.

$\nabla_w J = \frac{1}{N} \sum_n^N (\hat{y}_n - y_n) \nabla_w (\vec{w} \cdot \vec{x_n} - y_n)$

$= \frac{1}{N} \sum_n^N (\hat{y}_n - y_n) \vec{x_n}$

Then, we update the parameter with gradient descent:

$w \leftarrow w - \eta \vec{\nabla_w} J$

while choosing an appropriate step size η

5

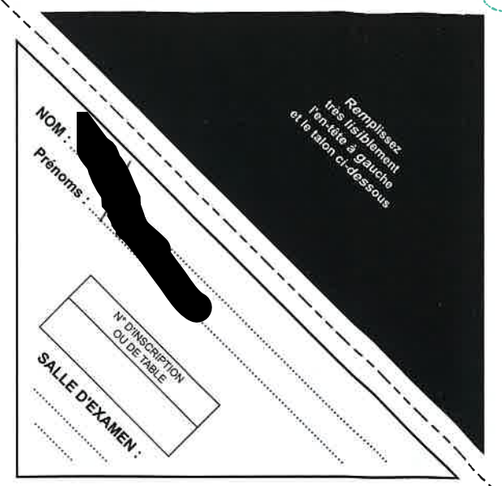
université PARIS-SACLAY

ANNÉE : .....

UNITÉ D'ENSEIGNEMENT : .....

ÉPREUVE DE : .....

NOTE /20	Coefficient	Note affectée du coefficient



Si votre composition comporte plusieurs feuilles, numérotez-les .../4

5,5

1 . 1

a) The left model is richer and more expressive. With small λ, or little regularization, there is a big gap between train and validation score. This is a sign of overfitting

1,5

As λ increases, the impact of regularization becomes heavier. The model is less prone to overfitting and generalizes better. That is the reason for the growth in validation score.

b) The left model is better. ~~The right model seems to make no pro~~ we can fight overfitting with regularization, and the validation is slightly higher.

ok

Meanwhile, ~~with~~ a too simple model will underfit and fail to learn the pattern in the data. ~~We have~~ only The only solution is to raise

PARTIE À RABATTRE

model complexity here.

1-2 :

Left case c : complex with high regularization.
The prediction curve is curvy at some point and almost flat at others => flat regions are the effects of regularization.

0,5

Right case d : the prediction curve is regularly curvy, so no regularization

1-3

1

a) Validation and train error converges, training seems to be going well. There is no overfitting

yes!

b) Due to inherent noise and bias in data, training and validation error is not 0.

c) Test errors should be similarly low, if the train and validation is performed correctly

1-4

a) Solid line : separator, decision surface
hyperplane

b) Dashed line : their margin delimiter

c) OK : top middle, top right
Overfitting : bottom middle, bottom right
Bottom left is between OK and overfitting
Top left is underfitting

d) Top middle is a good choice. It makes some prediction error, but should generalize well.
Top right is also good. It makes fewer error than top middle, but may generalize a slightly less.

1,75

1-5

0,5

Benji may start to overfit the test set. He keeps optimizing ~~the parameters~~ for this same test set

1-6 :

0,25

A simple solution is to simply remove the biased ~~in~~ * elements in the training data.
A harder solution is to diversify and increase the fairness of the data.
→ How ?

0,25

c) Mini batch also ~~leads~~ reduce computation time when training size is large, hence faster training iterative process.

2-4

a)

$$J = \frac{1}{4}\sum_n^N (w \cdot x_n - y_n)^4 + \frac{1}{4}\lambda \|w\|^4$$

$$\|w\|^4 = \left[(w_1^2 + w_2^2 + \dots + w_d^2)^{1/2}\right]^4 = (w_1^2 + w_2^2 + \dots + w_d^2)^2$$

$$\nabla_w \|w\|^4 = 2\|w\|_2^2 \nabla_w (w_1^2 + w_2^2 + \dots + w_d^2)$$

$$= 2\|w\|_2^2 \, 2\vec{w} = 4\|w\|^2 \vec{w}$$

$$\nabla_w \Rightarrow \nabla_w \frac{1}{4}\lambda\|w\|^4 = \frac{1}{4}\lambda 4\|w\|^2 \vec{w} = \lambda\|w\|^2 \vec{w}$$

$$\nabla_w \frac{1}{4}\sum_n^N (w \cdot x_n - y_n)^4 = 4\frac{1}{4}\sum_n^N (w \cdot x_n - y_n)^3 \nabla_w (w \cdot x_n - y_n)$$

$$= \sum_n^N (w \cdot x_n - y_n)^3 \vec{x_n}$$

$$\nabla_w J = \sum_n^N (w \cdot x_n - y_n)^3 \vec{x_n} + \lambda\|w\|^2 \vec{w}$$

1

Gradient Descent:

$$w \leftarrow w - \eta \nabla_w J$$

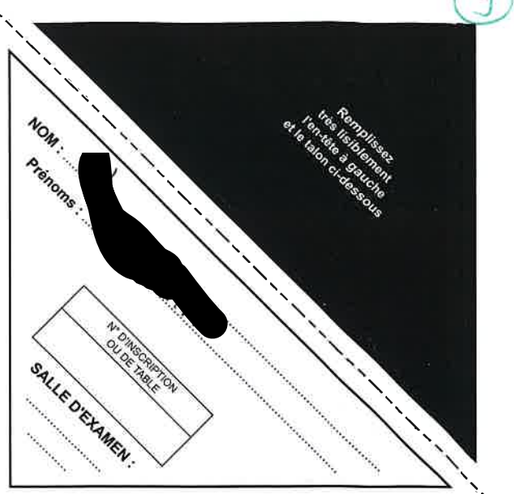
université PARIS-SACLAY

ANNÉE : .....

UNITÉ D'ENSEIGNEMENT : .....

ÉPREUVE DE : .....

NOTE /20	Coefficient	Note affectée du coefficient



Si votre composition comporte plusieurs feuilles, numérotez-les .../4

7,75

2.1 :

```
function fit (X, Y, η, niter):
    W ← initialize_weight(X.shape)

    for i in 1..niter:
        ŷ ← matmul (W, X)        (vector)
        error ← ŷ - Y            (vector)

        w ← w - η((error * X).mean())

    return w

function predict (X, w):
    X ← augment (X)   (add 1 at the top)
    return matmul (w, X)
```

Notez avec exactitude votre numéro de table ou d'inscription.
Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

PARTIE A — À RABATTRE

2-2.

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \end{bmatrix}$$

$w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\quad \hat{Y} = w^T X$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \end{bmatrix}$$

$$[1\ 1] \begin{bmatrix} 2 & 2 & 3 & 3 \end{bmatrix} = \hat{Y}$$

1/5

$$\hat{Y} - Y = [2\ 2\ 3\ 3] - [1\ 2\ 2\ 3]$$
$$= [1\ 0\ 1\ 0]$$

$$\nabla_w J = 1\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1\begin{bmatrix} 1 \\ 2 \end{bmatrix} + 0\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

✓

$$w \leftarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 1\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

2-3

0,5

We can transform the input with some feature map including the products of original feature

$x \rightarrow \phi(x)$

2-5.

a) We compute gradient for a minibatch instead of the whole data set:

$$\nabla_w J = \frac{1}{nmb} \sum_n^{nmb} (\hat{y}_n - y_n) \vec{x}_n$$

nmb = # of datapoints in a minibatch.

Then, we loop through minibatch inside the epoch loop: in the fit function

1

```
for i in 1... niter:
    for xmb, ymb in zip(XMB, YMB):
        ymb ← matmul(w, xmb)
        error
        w ← w - η * mean((ymb - ymb) * x)
```

XMB, YMB : all x and Y in batches
xmb, ymb : a batch.

b) Randomness of minibatch adds variations to the training process. Thus, it makes the model less prone to overfit, compared to full batch.

0,75

As the model sees more variations of input during training, it tends to generalize better

16

13

université PARIS-SACLAY

ANNÉE : .....

UNITÉ D'ENSEIGNEMENT : .....

ÉPREUVE DE : .....

NOTE /20	Coefficient	Note affectée du coefficient

Remplissez très lisiblement l'en-tête à gauche et le talon ci-dessous

NOM :

Prénoms :

N° D'INSCRIPTION OU DE TABLE :

SALLE D'EXAMEN :

.....

Si votre composition comporte plusieurs feuilles, numérotez-les .../4

2 - 4

b) fit function will be similar, except gradient step

0,25 for LinReg², the update step will need to implement this formula:

$$w \leftarrow w - \eta \nabla_w J$$

with $\nabla_w J = \sum_n^N (w \cdot x_n - y_n)^3 \vec{x_n} + \lambda \|w\|^2 \vec{w}$

predict function is the same.

c) With MQE, ~~error~~ penalty for error are heavier. Better performance means closer fit between model and data. 0,25 loss function for LinReg² has $\|w\|^4$ which force the weights to be very small => lot of constraints and little freedom. Therefore LinReg² will perform worse than LinReg

Notez avec exactitude votre numéro de table ou d'inscription.
Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

Notez avec exactitude votre numéro de table ou d'inscription.
Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

PARTIE À RABATTRE

0,75

3-2.

Each ~~tas~~ class k will have (k-1) separator.

Overall, we are counting each classifier twice (class k vs j and j vs k)

So, in total, # of classifier = $\frac{k(k-1)}{2}$ ✓

0,5

each classifier has D parameters.

=> Total # of parameters = $\frac{k(k-1)}{2} \times D$

3-1:

a) Class 1 : loss = $\sigma(\vec{w}_{12} \cdot \vec{x_n}) - (1-0)$

$= \sigma(\vec{w}_{12} \cdot \vec{x_n}) - 1$

Class 2 : loss = $\sigma(\vec{w}_{12} \cdot \vec{x_n}) - (0-1)$

$= \sigma(\vec{w}_{12} \cdot \vec{x_n}) + 1$ 0,25

=> loss = $\sigma(\vec{w}_{12} \cdot \vec{x_n}) - (2\,\mathbb{1}_{k=\bar{k}} - 1)$

$\mathbb{1}_{k=\bar{k}} = \begin{cases} 1 \text{ when } k = \bar{k} = k_{true} \\ 0 \text{ when } k \neq \bar{k} = k_{true} \end{cases}$

$t_{n,12} = -(2\,\mathbb{1}_{k=\bar{k}} - 1) = 1 - 2\,\mathbb{1}_{k=\bar{k}}$

Notez avec exactitude votre numéro de table ou d'inscription.
Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.