**INTRODUCTION MACHINE LEARNING**

**EXERCISE 3**

A red sign with white text

Description automatically generated

|  |
| --- |
| **TEACHER:** |
| Johannes Kiesel |

|  |
| --- |
| **GROUP:** |
| Group 16 |
|  |
| **SUBMITTED BY:** |
| Aaron Perez Herrera |
| Cesar Fernando Gamba Tiusaba |
| Chun Ting Lin |
| Olubunmi Emmanuel Ogunleye |

Exercice 1: Linear Models:

1. Name these concepts:

The point-wise loss

Global loss

 : objective function or error function

: Parameter vector or (p+1)-dimensional hypothesis.

* direction vector excluding wo or the p-dimensional direction vector.

1. How would the figure below change if wo is halved?

The separating line in the figure corresponds to the equation of the decision boundary:

When w0 halved, the intercept of the line on the x​ plane changes. Specifically:

W0 affects the position of the decision boundary but **not its orientation**, as the weights w1​​ determine the slope.

* The absolute value of X​ increases, meaning the intercept moves further away from the origin. The boundary shifts outward along the direction normal to w1​.
* Since w1​ remains unchanged, the slope of the decision boundary does not change. Thus, the boundary retains its orientation, and only its position is affected.
* The blue line shifts is the original equation, when W0 is halved, the orange line is the shifted equation, some of the points (red ones) keep the label, but some other points (black points) change of the label

1. What is the difference (if any) between decision boundaries for linear and logistic regression:

The key difference is the predicted output.

* Linear regression: predicts a continuous output and its decision boundary is a straight line.
* Logistic regression: predicts a probability (between 0 and 1) and uses a sigmoid function to map a linear combination of features to this probability. The decision boundary in logistic regression can take a nonlinear form.

1. The lecture notes slides state that a key difference between ridge  and lasso  regression is that, with lasso regression, parameters can be reduced to zero. Explain why.

This is because Lasso regression uses L1 regularization, which adds a penalty term proportional to the absolute value of the coefficients. As a result, some coefficients can be reduced to zero, effectively removing the corresponding features from the model

1. Why can the gradient descent method not be applied for ?

can not be expressed as a differentiable function, also because the loss function considered when applying 0/1 loss is typically non-convex.

Exercise 2: Pointwise Loss Functions

In the lecture notes, slide ML:III-63 on loss computation for logistic regression in detail, the rightmost plot "Loss over hyperplace distance" shows the pointwise logistic and 0/1 loss for a logistic regression model for ,that means, for examples with c = 1. In this exercise you will investigate the case of examples with c = 0.

1. Show that

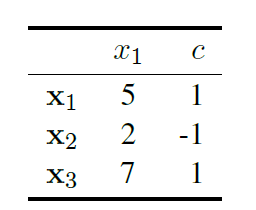
For c = 0

1. Draw the plot "Loss over hyperplane distance" for examples with c = 0, showing both logistic loss and 0/1 loss.

Exercise 3 : Gradient Descent (1.5+0.5+0+0.5+0.5+1+0+0=4 Points)

In this exercise you will be calculating one iteration of the LMS algorithm, slide ML:I-42.

The set D contains the following three examples of one-dimensional vectors with the two classes {−1, 1}:



Assume the weight vector w was randomly initialized to w := (0, 1)T and x1 was randomly selected for

the first iteration of the algorithm.

1. Plot the line defined by w and all examples from D into one coordinate system.

The decision boundary is defined by w^T x = 0.

For w = (0, 1)^T, the line equation becomes x\_2 = 0. The line is plotted along with the points x1, x2, and x3 on a 2D plane.

The examples in D are:

- w(random) = (0, 1) -> w0 = 0 and w1 = 1

- x1 = (5), c = 1

- x2 = (2), c = -1

- x3 = (7), c = 1

WT \* X1 = w0 + w1 \* 5 = 5

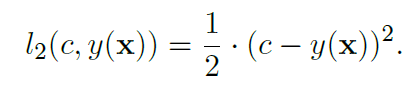
WT \* X2 = w0 + w1 \* 2 = 2

WT \* X3 = w0 + w1 \* 7 = 7





1. Compute the squared loss w.r.t. x1 and w. The squared loss is defined as



The squared loss is defined as:

L2 (c, y(x1) = ½ \* (1 – 5) ^2 = 8

1. Show that the loss gradient, (∂l2/∂w0, ∂l2/∂w1)^T , is indeed equal to −δ · x.

The gradient of the squared loss is derived as:

Gradient\_w l\_2 = -(c - w^T x) \* x.

This matches the formula for the LMS algorithm: Delta\_w = eta \* delta \* x.

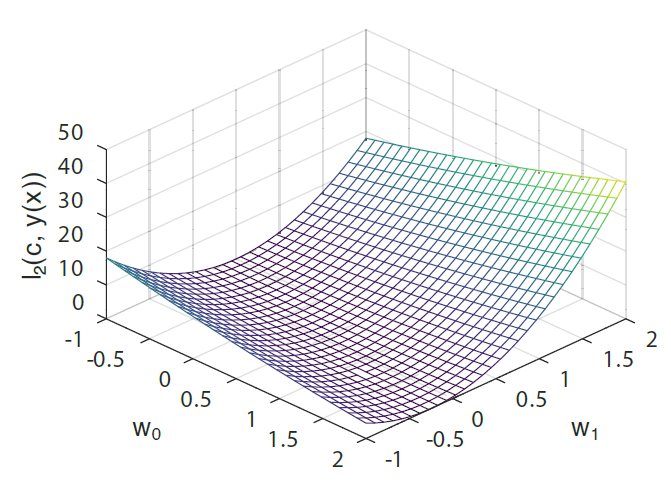
1. Derive the loss gradient for x1 and w.

Derivation using the chain rule:

1. Calculate Δw with a learning rate η = 0.01 for x1 and w.

The update rule is:

1. The following plot shows the loss landscape defined by l2 for x1. Mark the location of the model for w and for its update w + Δw.



The cost reduced significatively, also improved the values of “w”, as we can see in the graph we are getting closer to the “deep” part of exercise.

1. Compute the squared loss w.r.t. x1 and the updated w. Could it be possible that this loss is now larger than it was before the update?

w= w+∆w

w0=0-0.2= -0.2

w1=1-0.2= 0.8

L= 1/2\*(yi-(wo+w1\*x1))^2=3.92

No, the loss has been reduced. This is thanks to the regularization process that penalize huge values and search for a better solution with less cost

1. Repeat for x2, and x3.

For x2:

w= w+∆w

w0=0-0.04= -0.04

w1=1-0.04= 0.96

For x3:

w= w+∆w

w0=0-0.42= -0.42

w1=1-0.42= 0.58

Exercise 3: Overfitting and train-test leakage (1+1+0=2 Points)

1. What is the experimental setup of choice when trying to detect overfitting?

The experimental setup for detecting overfitting involves evaluating the model’s performance on separate datasets. This involves splitting the dataset into possibly three sets as explained in the lecture. These include:

* Training Set: This dataset is used to train the model and learn the parameters.
* Validation Set: This is used to evaluate or monitor overfitting during the training. If the losses due to validation increase and training loss decreases, there is an indication of overfitting.
* Test Set: It is used for the final evaluation of the model after it is completely trained and validated.

A comparison of the performance between training, validation and test datasets is done and if there is a large gap between training accuracy and validation/test accuracy, it implies there is an overfitting.

1. What are methods to mitigate overfitting?

* Increase the quality and quantity of the datasets.
* Data Augmentation to generate new training samples.
* Early stopping of the optimization/ refinement process.
* Regularization or smoothing of the model function using Lasso or Ridge Regression.

1. What must be paid attention to when performing a train-validation split on the following datasets in the given problems?

Detecting pneumonia from chest x-rays. Data includes 112,120 unique images from 30,805 unique patients.

1. Data Splitting: Ensure that all images from the same patient are in either the training or validation set, not to be able to predict for an unseen patient.
2. Stratified Sampling: use stratified sampling to maintain a proportion of pneumonia and non-pneumonia cases in training and validation data.
3. Avoid using datasets that reveal classification labels.

Given 1000 voice recordings (single sentences) of 100 people in total from 5 German cities. The model should be able to classify the dialects of arbitrary people into one of these cities.

1. Data Splitting: Ensure recording from the same person does not appear in both the training and validation sets.
2. Stratified Sampling: Maintain a balanced representation of recording from all cities in both sets.

Given 1000 voice recordings (single sentences) of 100 people in total from 5 German cities. The model should be able to rate the dialects of arbitrary people from all over Germany by intelligibility.

1. Data Splitting: Ensure diverse speakers from all cities are included in both train and validation set to capture a range of dialects.
2. Stratified Sampling: Ensure the validation dataset contains dialects from multiple cities and not one

Given 1000 voice recordings (single sentences) of 100 people in total from 5 German cities. The model should be able to classify the person that said a given sentence.

1. Data Splitting: Ensure no overlap of the same sentence spoken by different people in the training and validation set. Also, ensure that each speaker recording belongs entirely to the training or validation set
2. Stratified Sampling: Ensure the validation dataset contains diverse sentences to reflect real-world scenarios.

Exercise 5: Regularization (1+1=2 Points)

Suppose we are estimating the regression coefficients in a linear regression model by minimizing the objective function L.

L(w) = RSStr(w) + λwTw

The term RSStr(w) = ∑(xi,yi)∈Dtr (yi − wT xi)2 refers to the residual sum of squares computed on the set Dtr that is used for parameter estimation. Assume that we can also compute an RSStest on a separate set Dtest that we don’t use during training.

When we vary the hyperparameter λ, starting from 0 and gradually increase it, what will happen to the following quantities? Explain your answers.

1. The value of RSStr(w) will. . .

* remain constant. No
* steadily increase. Yes, because when the hyper parameter λ is increased, the regularization term gains more importance relative to the residual sum of squares, which forces the weight to shrink towards 0
* steadily decrease. No
* increase initially, then eventually start decreasing in an inverted U shape. No
* decrease initially, then eventually start increasing in a U shape. No

1. The value of RSStest(w) will. . .

* remain constant. No
* steadily increase. No
* steadily decrease. No
* increase initially, then eventually start decreasing in an inverted U shape. No
* decrease initially, then eventually start increasing in a U shape. Yes, because the regularization term reduces overfiting by improving generations, but continuous increase result in overly simplified model and underfitting the data leading to worse performance and RSStest(w) increasing again.

Exercise 6 : P Implementing Logistic Regression Classifier (1+1+2+2+1+1=8 Points)

In this exercise, you will implement a logistic regression model for predicting whether a given text was

written by a human or generated by a language model.

Download and use these files from Moodle:

• features-train.tsv: Feature vectors for each example in the training set.

• features-test.tsv: Feature vectors for each example in the test set.

• labels-train.tsv: Labels for each example in the training set indicating the class is\_human

(C = {True, False})

• programming\_exercise\_logistic\_model.py: Template program for writing your

implementation. It contains function stubs for each function mentioned below. Use the following

command to run the program:

python3 programming\_exercise\_logistic\_model.py

features-train.tsv labels-train.tsv

features-test.tsv predictions-test.tsv

• requirements.txt: Requirements file for the template; can be used to install dependencies.

(a) Implement two functions to load the dataset:

load\_feature\_vectors reads feature vectors from a features-\*.tsv and returns the

contained multiset of feature vectors X as an n-by-(p+1) matrix.1

load\_class\_values reads the is\_human labels from the labels-train.tsv as one list of 1s if

the example is human and 0s if it is machine-generated.

How many examples of each class are in the data set?

(b) Implement a function misclassification\_rate to measure the misclassification rate of the

model’s predictions.

What is the misclassification rate of a random classifier on the training set? Support your answer

with code.

(c) Implement a function logistic\_function to calculate the output of the logistic (sigmoid)

function w.r.t. input x parameterized by w and a function logistic\_prediction to predict the

class of x accordingly.

(d) Implement a function train\_logistic\_regression\_with\_bgd that fits a logistic regression

model using the Batch Gradient Descent (BGD) algorithm. A parameter of the function specifies the

fraction of training examples to not use for training but for validation. The function returns the

trained weights as p + 1-vector and three lists containing the training loss, misclassification rate on

the training examples, and the misclassification rate on the validation examples after each iteration.

(e) Plot the training loss, misclassification rate on the training examples, and the misclassification rate

on the validation examples after each iteration.

Are loss and misclassification rate correlated?

(f) Use the trained model to predict the labels for each example in the test set (features-test.tsv).

Write the prediction as one column to a file predictions-test.tsv and submit that along with

your other solutions.

Check the files in the “FinalProgram” folder, there you will find the code and also the predictions-test.tsv