Supervised Learning (COMP0078) – Coursework 2

Due : 03 January 2024.

Submission

You may work in groups of up to two. You should produce a report (this should be in .pdf format) about your results. You will not only be assessed on the **correctness/quality** of your answers but also on **clarity of presentation**. Additionally make sure that your code is *well commented*. Please submit on moodle i) your report as well as a ii) zip file with your source code. Finally, please ensure that if you are working in a group both of your stud[illegible]ries, you should implement regression using m[illegible]ines such as for example cross-validation shoul[illegible]

Note. Each coursework part [illegible]ot able to prove an individual sub-part (e.g. 1.[illegible]se cases, you are allowed to assume the results i[illegible]e to prove them.

1 PART I [20%]

Rademacher Complexi[illegible]

In this problem we will find a[illegible]ce of hypotheses that depends only logarithmica[illegible]prove the bound on the generalization error see[illegible]

We will first show an interme[illegible]andom variables (namely $\mathbb{E}X_i = 0$ for all $i = 1,$[illegible]

[illegible]

1.1 [2 marks]. Let $\bar{X} = \max_i X_i$. Show that for any $\lambda > 0$

$$\mathbb{E}\bar{X} \leq \frac{1}{\lambda} \log \mathbb{E} e^{\lambda \bar{X}}$$

1.2 [5 marks]. Show that

$$\frac{1}{\lambda} \log \mathbb{E} e^{\lambda \bar{X}} \leq \frac{1}{\lambda} \log m + \lambda \frac{(b-a)^2}{8}$$

*Hint: use **Hoeffding's Lemma:** for any random variable X such that $X - \mathbb{E}X \in [a, b]$ with $a, b \in \Re$, and for any $\lambda > 0$, we have*

$$\mathbb{E}\ e^{\lambda(X - \mathbb{E}X)} \leq e^{\lambda^2 (b-a)^2/8}$$

1.3 [3 marks]. Conclude that by choosing λ appropriately,

$$\mathbb{E} \max_{i=1,\ldots,m} X_i \leq \frac{b-a}{2} \sqrt{2 \log(m)}$$

as desired.

We are almost ready to provide the bound for the Rademacher complexity of a finite set of hypotheses. Let S a fomote set of points in $\Re^n$ with cardinality $|S| = m$. We can define the Rademacher complexity of S similarly to how we have done for the Rademacher complexity of a space of hypotheses:

$$\mathcal{R}(S) = \mathbb{E}_\sigma \max_{x \in S} \frac{1}{n} \sum_{j=1}^{n} \sigma_j x_j,$$

With $\sigma_1, \ldots, \sigma_n$ Rademacher variables (independent and uniformly sampled from $\{-1, 1\}$.

1.4 [3 marks]. Show that

$$\mathcal{R}(S) \leq \max_{x \in S} \|x\|_2 \frac{\sqrt{2 \log(m)}}{n}$$

with $\|\cdot\|_2$ denoting the Euclidean norm.

1.5 [7 marks]. Let $\mathcal{H}$ be a se ality $|\mathcal{H}| < +\infty$. Let $S = (x_i)_{i=1}^n$ be a set of poi upper bound for empirical Rademacher comple ally.

2 PART II [40%]

Bayes Decision Rule a

In (binary) classification proble ction $c : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{1, -1\}$. The quali ation error

assuming to sample an input-

2.1 [4 marks]. Let $\mathbf{1}_{y=y'}$ be . Show that the misclassification error cor

(Surrogate Approaches) Si ress the learning problem directly and in practice one usually looks for a real valued function $f : \mathcal{X} \to \Re$ solving a so-called *surrogate problem*

$$\mathcal{E}(f) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x), y) \; d\rho(x, y)$$

where $\ell : \Re \times \Re \to \Re$ is a "suitable" convex loss function that makes the surrogate learning problem more amenable to computations. Given a function $f : \mathcal{X} \to \Re$, a classification rule $c_f : \mathcal{X} \to \{-1, 1\}$ is given in terms of a "suitable" map $d : \Re \to \{-1, 1\}$ such that $c_f(x) = d(f(x))$ for all $x \in \mathcal{X}$. Here we will look at some surrogate frameworks.

A good surrogate method satisfies the following two properties:

(Fisher Consistency). Let $f_* : \mathcal{X} \to \Re$ denote the expected risk minimizer for $\mathcal{E}(f_*) = \inf_{f : \mathcal{X} \to \Re} \mathcal{E}(f)$, we say that the surrogate framework is Fisher consistent if

$$R(c_{f_*}) = \inf_{c : \mathcal{X} \to \{-1,1\}} R(c)$$

(Comparison Inequality). The surrogate framework satisfies as *comparison inequality* if for any $f : \mathcal{X} \to \Re$

$$R(c_f) - R(c_{f_*}) \leq \sqrt{\mathcal{E}(f) - \mathcal{E}(f_*)}$$

In particular, if we have an algorithm producing estimators f_n for the surrogate problem such that $\mathcal{E}(f_n) \to \mathcal{E}(f_*)$ for $n \to +\infty$, we automatically have $R(c_{f_n}) \to R(c_{f_*})$.

2.2 [4 marks]. (**Assuming to know** ρ), calculate the closed-form of the minimizer f_* of $\mathcal{E}(f)$ for the:

a) squared loss $\ell(f(x), y) = (f(x) - y)^2$

b) exponential loss $\ell(f(x), y) = \exp(-yf(x))$,

c) logistic loss $\ell(f(x), y) = \log(1 + \exp(-yf(x)))$,

d) hinge loss $\ell(f(x), y) = \max(0, 1 - yf(x))$.

(hint: recall that $\rho(x, y) = \rho(y|x)\rho_{\mathcal{X}}(x)$ with $\rho_{\mathcal{X}}$ the marginal distribution of ρ on $\mathcal{X}$ and $\rho(y|x)$ the corresponding conditional

$$\mathcal{E}(f) = \int$$

you can now solve the pro

2.3 [4 marks]. The minimiz ... is called *Bayes decision rule*. Write expli ... **priori**).

2.4 [4 marks]. Are the surro ... , can you find a map $d : \Re \to \{-1, 1\}$ such ... minimizer of the surrogate risk $\mathcal{E}$? If it is t

(Comparison Inequality fo ... minimizer of the expected risk for the surrogat ... (2.2). Let sign : $\Re \to \{-1, 1\}$ denote the "sign"

Prove the following compariso

$$0 \leq R(\text{sign}(f)) - R(\text{sign}(f_*)) \leq \sqrt{\mathcal{E}(f) - \mathcal{E}(f_*)},$$

by showing the following intermediate steps:

2.5.1 [8 marks]. $|R(\text{sign}(f)) - R(\text{sign}(f_*))| = \int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x)$,
Where $\mathcal{X}_f = \{x \in \mathcal{X} \mid \text{sign}(f(x)) \neq \text{sign}(f_*(x))\}$.

2.5.2 [8 marks]. $\int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x) \leq \int_{\mathcal{X}_f} |f_*(x) - f(x)| d\rho_{\mathcal{X}}(x) \leq \sqrt{\mathbb{E}(|f(x) - f_*(x)|^2)}$.
Where $\mathbb{E}$ denotes the expectation with respect to $\rho_{\mathcal{X}}$

2.5.3 [8 marks]. $\mathcal{E}(f) - \mathcal{E}(f_*) = \mathbb{E}(|f(x) - f_*(x)|^2)$.

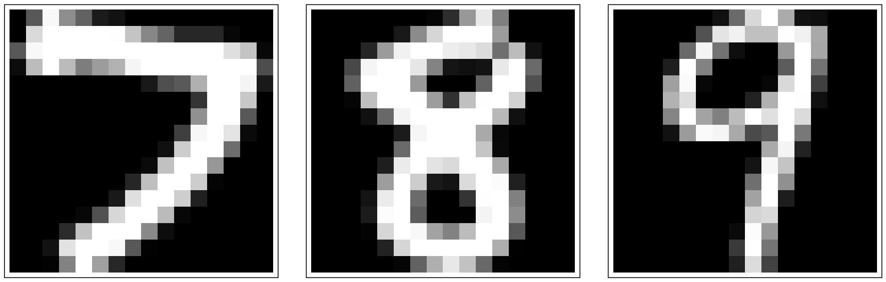

Figure 1: Scanned Digits

3 PART III [40%]

Kernel perceptron (Handwritten Digit Classification)

Introduction: In this exer [illegible] igits. The task is quasi-realistic and you will [illegible] ely large dataset which you would not encounte [illegible]

You may already be famili [illegible] on in two ways, first we generalize the percept [illegible] inear separating surface and second, we genera [illegible] that instead of separating only two classes we [illegible]
Adding a kernel: The *kern* [illegible] as we did with basis functions so that class o [illegible] We will consider a single type of kernel, the po [illegible] ositive integer d controlling the dimension of th [illegible]
Training and testing the k [illegible] orithms operate on a single example $(\boldsymbol{x}_t, y_t)$ [illegible] a single kernel function $K(\boldsymbol{x}_t, \cdot)$ is added for [illegible] line training we repeatedly cycle through the [illegible] classifier is no longer changing when we cycl [illegible] be the case for some datasets that the classif [illegible] r will *generalize* better if not trained to conver [illegible] w many epochs to train a particular classifier [illegible] erting an online algorithm to a batch algorithm [illegible] e table correctly describes training for a single [illegible] rect for multiple epochs, however, explicit nota [illegible] hru the data) is represented by repeating the dataset with the $\boldsymbol{x}_i$'s renumbered. I.e., suppose we have a 40 element training set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_{40}, y_{40})\}$ to model additional epochs simply extend the data by duplication, hence an m epoch dataset is

$$\underbrace{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{40}, y_{40})}_{\text{epoch 1}}, \underbrace{(\boldsymbol{x}_{41}, y_{41}), \ldots, (\boldsymbol{x}_{80}, y_{80})}_{\text{epoch 2}}, \ldots, \underbrace{(\boldsymbol{x}_{(m-1)\times 40+1}, y_{(m-1)\times 40+1}), \ldots, (\boldsymbol{x}_{(m-1)\times 40+40}, y_{(m-1)\times 40+40})}_{\text{epoch m}}$$

where $\boldsymbol{x}_1 = \boldsymbol{x}_{41} = \boldsymbol{x}_{81} = \ldots = \boldsymbol{x}_{(m-1)\times 40+1}$, etc. Testing is performed as follows, once we have trained a classifier $\boldsymbol{w}$ on the training set, we simply use the trained classifier with only the *prediction* step for each example in test set. It is a mistake when ever the prediction $\hat{y}_t$ does not match the desired output y_t, thus the test error is simply the number of mistakes divided by test set size. Remember in testing the *update* step is never performed.

	Two Class Kernel Perceptron (training)
Input:	$\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\Re^n, \{-1, +1\})^m$
Initialization:	$\boldsymbol{w}_1 = \boldsymbol{0}$ $(\alpha_0 = 0)$
Prediction:	Upon receiving the tth instance $\boldsymbol{x}_t$, predict $\hat{y}_t = \text{sign}(\boldsymbol{w}_t(\boldsymbol{x}_t)) = \text{sign}(\sum_{i=0}^{t-1} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_t))$
Update:	`if` $\hat{y}_t = y_t$ `then` $\alpha_t = 0$ `else` $\alpha_t = y_t$ $\mathbf{w}_{t+1}(\cdot) = \mathbf{w}_t(\cdot) + \alpha_t K(\boldsymbol{x}_t, \cdot)$

Generalizing to k classes: Design a method (or research a method) to generalise your two-class classifier to k classes. The method should return a vector $\boldsymbol{\kappa} \in \Re^k$ where κ_i is the "confidence" in label i; then you should predict either with a label that maximises confidence or alternately with a randomised scheme.

I'm providing you with mathematica code for a 3-classifier and a demonstration on a small subset of the data. First, however, my mathematica implementation is flawed and is relatively inefficient for large datasets. One aspect of your goals are to improve my code so that it can work on larger datasets. The mathematical logic of the algo[illegible] and/or the data structures will need to change [illegible] ment sufficiently fast code in Python (or the la[illegible]

Files: From `http://www0.`[illegible] files relevant to this [illegible]

`poorCodeDemoD`[illegible]
`dtrain123.dat` [illegible])
`dtest123.dat` [illegible]
`zipcombo.dat` [illegible]

each of the data files consists [illegible] he first value is the digit, the remaining 256 [illegible] ween -1 and 1. In attempting to understand [illegible] thematica code. However, remember the demo [illegible] ugh less efficient implementations are possible i[illegible] ire thought and observation of behaviour on th[illegible] ns for the kernel perceptron.

Experimental Protocol: Yo[illegible] (errors reported should be percentages not raw [illegible]

1. Basic Results: Perform [illegible] `combo` into 80% train and 20% test. Rep[illegible] dard deviations. Thus your data table, he[illegible]

2. Cross-validation: Perform 20 runs : when using the 80% training data split from within to perform 5-fold cross-validation to select the "best" parameter d^* then retrain on full 80% training set using d^* and then record the test errors on the remaining 20%. Thus you will find 20 d^* and 20 test errors. Your final result will consist of a mean test error±std and a mean d^* with std.

3. Confusion matrix: Perform 20 runs : when using the 80% training data split that further to perform 5-fold cross-validation to select the "best" parameter d^* retrain on the full "80%" training set using d^* and then produce a *confusion matrix*. Here the goal is to find "confusions" thus if the true label (on the test set) was "7" and "2" was predicted then a "error" should recorded for "(7,2)"; the final output will be a 10×10 matrix where each cell contains a confusion error *rate* and its standard deviation (here you will have averaged over the 20 runs). Note the diagonal will be 0. In computing the error rate for a cell use

$$\frac{\text{"Number of times digit } a \text{ was mistaken for digit } b \text{ (test set)"}}{\text{"Number of digit } a \text{ points (test set)"}}.$$

4. Within the dataset relative to your experiments there will be five hardest to predict correctly "pixelated images." Print out the visualisation of these five digits along with their labels. Is it surprising that these are hard to predict? Explain why in your opinion that is the case.

5. Repeat 1 and 2 (d^* is now c and $\{1, \ldots, 7\}$ is now S) above with a Gaussian kernel

$$K(\boldsymbol{p}, \boldsymbol{q}) = e^{-c\|\boldsymbol{p}-\boldsymbol{q}\|^2},$$

c the width of the kernel is now a parameter which must be optimised during cross-validation however, you will also need to perform some initial experiments to a decide a reasonable set S of values to cross-validate c over.

6. Choose (research) an alternate method to generalise the kernel perceptron to k-classes then repeat 1 and 2.

Assessment: In your report you will not only be assessed on the correctness/quality of your experiment (e.g., sound methods for choosing parameters, reasonable final test errors) but also on the clarity of presentation and the insightfulness of your observations. Thus the aim is that your report is sufficiently detailed so that the reader could largely ... port alone. The report should also contain the ...

- A discussion of any para...
- A discussion of the two ... classifiers.
- A discussion comparing ...
- A discussion of your im... discuss how the sum $\boldsymbol{w}(\cdot) = \sum_{i=0}^{m} \alpha_i K(\boldsymbol{x}$ ... are added to the sum during training.
- Any table produced in 1... e table.

Note: (further comments ... entage correct," rather it will be a qualitative ... t that is merely correct/good as a baseline can ... ve 32-40 points. Regarding page limits the exp... o more of three pages of text (this does not incl... the assignment). There is no strict limit, howev... s one page may be sufficient) and there are a ...