

# COMP9417 - Machine Learning

## Homework 2: Bias, Variance and an application of Gradient Descent

Introduc  
behavio  
different

characterizing the  
m for combining

Points A

- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu
- Qu

What to

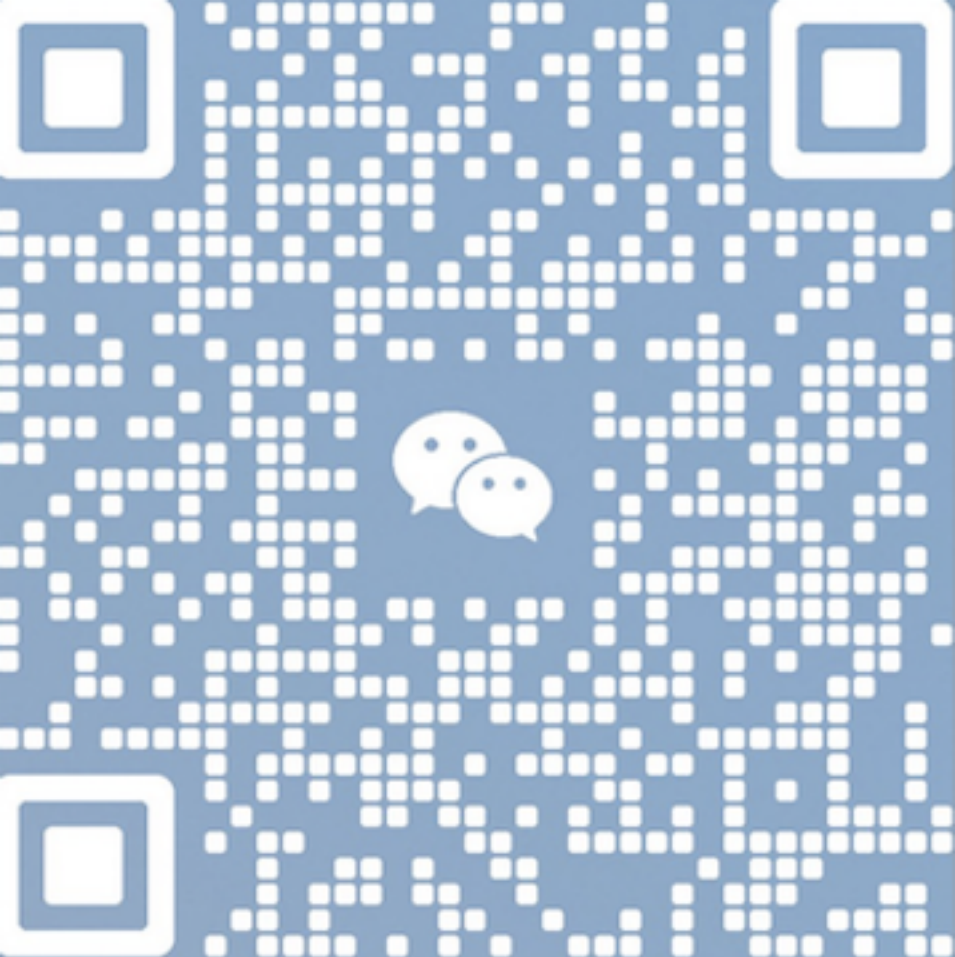
- A **single PDF** file which contains solutions to each question. For each question, provide your solution in the form of text and requested plots. For some questions you will be requested to provide screen shots of code used to generate your answer — only include these when they are explicitly asked for.
- **.py file(s) containing all code you used for the project, which should be provided in a separate .zip file.** This code must match the code provided in the report.

- You may be deducted points for not following these instructions.
- You may be deducted points for poorly presented/formatted work. Please be neat and make your solutions clear. Start each question on a new page if necessary.
- You **cannot** submit a Jupyter notebook; this will receive a mark of zero. This does not stop you from developing your code in a notebook and then copying it into a .py file though, or using a tool such as **nbconvert** or similar.
- We will set up a Moodle forum for questions about this homework. Please read the existing questions before posting new questions. Please do some basic research online before posting questions. Please only post clarification questions. Any questions deemed to be *fishing* for answers will be ignored and/or deleted.

- Ple  
an
- Ple  
cou  
ack  
zIL
- As  
tion
- You  
tion  
der

When ar

- Du  
mc
- Lat  
am  
80
- Sul



ility to check for

ner people in the  
own solution and  
their name(s) and

homework ques-  
isconduct.

derivation ques-  
You must do the

l not be actively

e grade. For ex-  
nal grade will be  
ro.

### Question 1. Bias of Estimators

Let  $\gamma > 0$  and suppose that  $X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} N(\gamma, \gamma^2)$ . We define:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i,$$

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2.$$

You may use the following two facts without proof:

(F1)  $\bar{X}$  and  $S^2$  are independent.

(F2)

*What must*

(a)

(b)

(c)

(d)

(e)

### Question

In the

tions

power

mini

The

algo

learn

Suppose

we have

data

$(x_i, y_i)$

for

$i = 1, \dots, n$ ,

which

we collect

into a

single

data set

$D_0$ . We

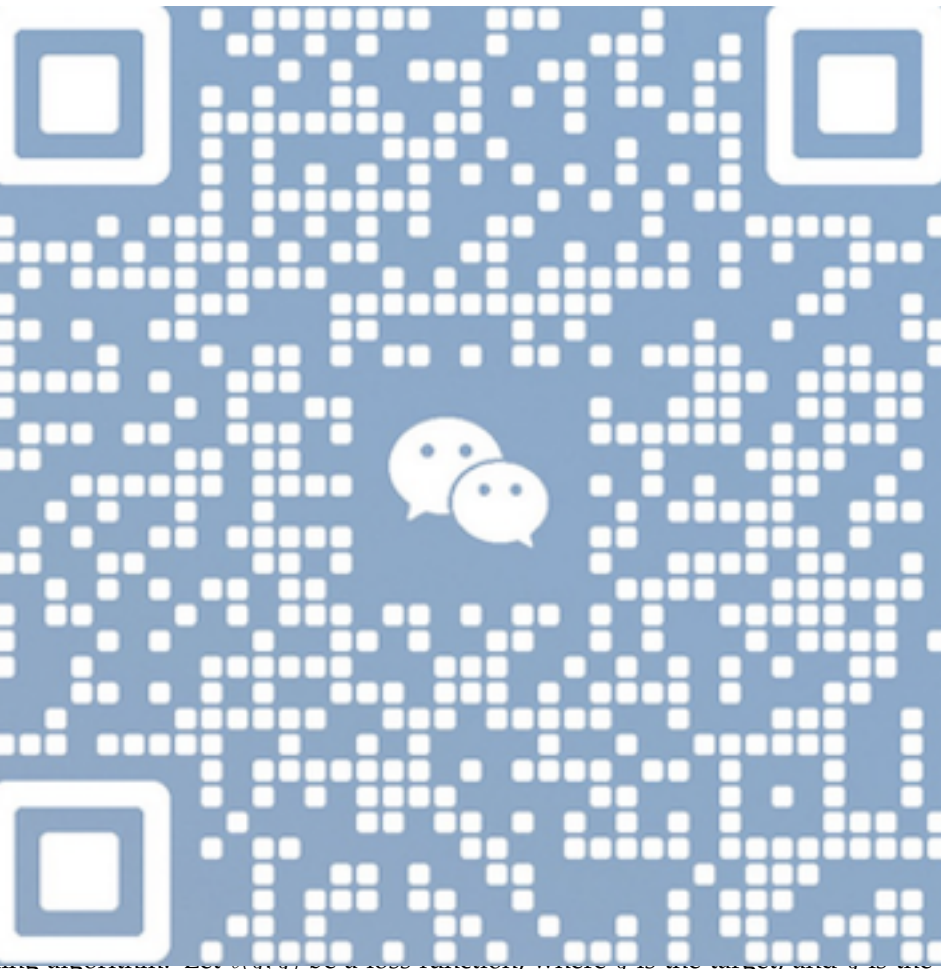
then set

the

number

of

desired



*For all parts, you*

derive an explicit

2. Find the con-  
stant  $a_2$  to minimize

is greater than or equal to the

learning combina-  
tion. A good idea is a very  
simple one beyond direct

of base learning  
up with a good

predicted value.<sup>3</sup>

<sup>1</sup>You do not need to worry about knowing or calculating  $c_*$  for this question, it is just some constant.

<sup>2</sup>For example, you could take  $\mathcal{F}$  to be the set of all regression models with a single feature, or alternatively the set of all regression models with 4 features, or the set of neural networks with 2 layers etc.

<sup>3</sup>Note that this set-up is general enough to include both regression and classification algorithms.

(I) Initialize  $f_0(x) = 0$  (i.e.  $f_0$  is the zero function.)

(II) For  $t = 1, 2, \dots, T$ :

(GC1) Compute:

$$r_{t,i} = -\frac{\partial}{\partial f(x_i)} \sum_{j=1}^n \ell(y_j, f(x_j)) \Big|_{f(x_j)=f_{t-1}(x_j), j=1,\dots,n}$$

for  $i = 1, \dots, n$ . We refer to  $r_{t,i}$  as the  $i$ -th pseudo-residual at iteration  $t$ .

(GC2) Construct a new *pseudo* data set,  $D_t$ , consisting of pairs:  $(x_i, r_{t,i})$  for  $i = 1, \dots, n$ .

(GC3) Fit a model to  $D_t$  using our base class  $\mathcal{F}$ . That is, we solve

(G

(G

(III)

We c

in (C

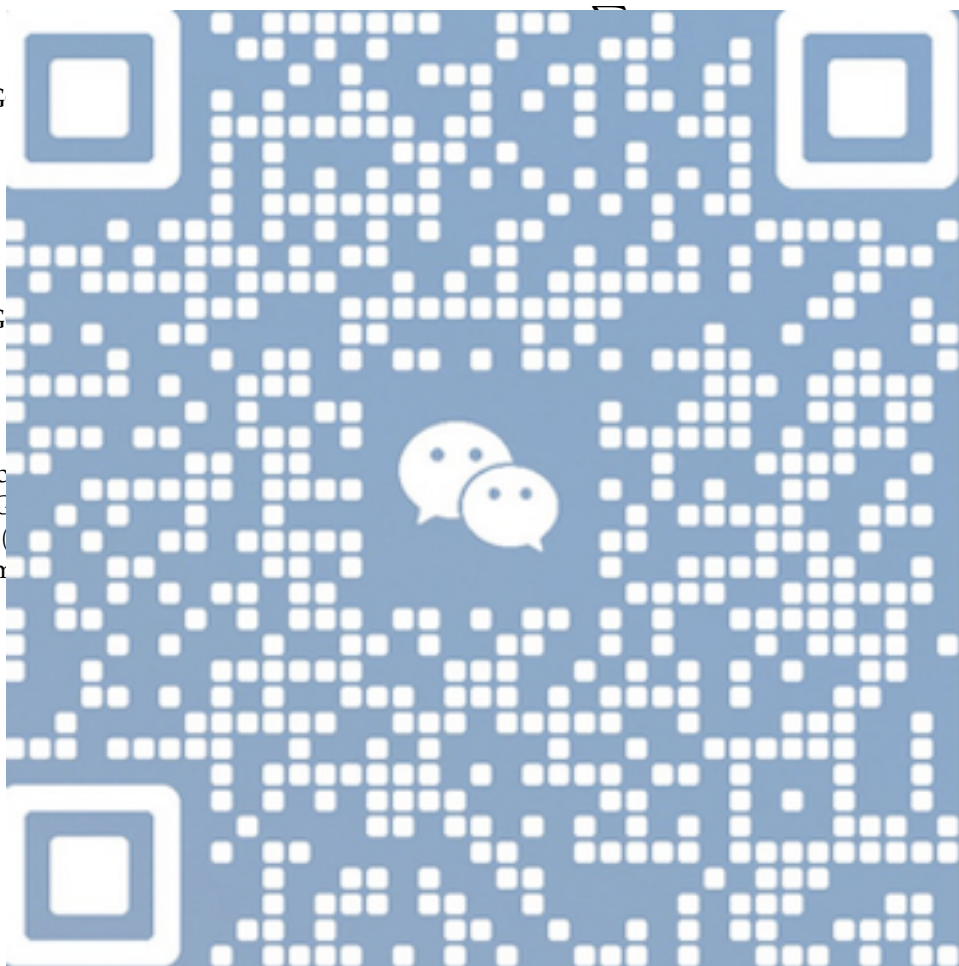
of  $f$

exam

(a)

(b)

(c) We will now implement the gradient-combination algorithm on a toy dataset from scratch, and we will use the class of decision stumps (depth 1 decision trees) as our base class ( $\mathcal{F}$ ), and squared error loss as in the previous parts.<sup>4</sup>. The following code generates the data and demonstrates plotting the predictions of a fitted decision tree (more details in q1.py):



class  $\mathcal{F}$ . Note that

et all occurrences

$y_j$ ), for all  $j$ . For

be real numbers.

arithm, show that

m in step (GC3)

according to the

<sup>4</sup>In your implementation, you may make use of `sklearn.tree.DecisionTreeRegressor`, but all other code must be your own. You may use `NumPy` and `matplotlib`, but do not use an existing implementation of the algorithm if you happen to find one.



```

1  np.random.seed(123)
2  X, y = f_sampler(f, 160, sigma=0.2)
3  X = X.reshape(-1,1)
4
5  fig = plt.figure(figsize=(7,7))
6  dt = DecisionTreeRegressor(max_depth=2).fit(X,y) # example model
7  xx = np.linspace(0,1,1000)
8  plt.plot(xx, f(xx), alpha=0.5, color='red', label='truth')
9  plt.scatter(X,y, marker='x', color='blue', label='observed')
10 plt.plot(xx, dt.predict(xx.reshape(-1,1)), color='green', label='dt') # plotting
    example model
11 plt.legend()
12 plt.show()
13

```



learners is increased? You should do this two times (two 5x2 plots), once with the adaptive step size, and the other with the step-size taken to be  $\alpha = 0.1$  fixed throughout. There is no need to split into train and test data here. Comment on the differences between your fixed and adaptive step-size implementations. How does your model perform on the different x-ranges of the data?

*What to submit: two 5 x 2 plots, one for adaptive and one for fixed step size, some commentary, and a screen shot of your code and a copy of your code in your .py file.*