CalPolyPomona

CS 4210 – Assignment #3

Maximum Points: 100 pts.

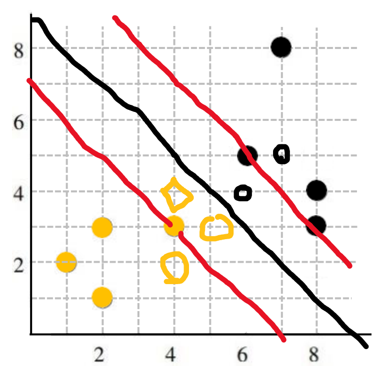
Bronco ID: 013 789 022

Last Name: Nash

First Name: Mason

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [20 points] Say you are given the training dataset shown below. This is a binary classification task in which the instances are described by two integer-valued attributes.



Dataset

a. [2 points] Draw the decision boundary and its parallel hyperplanes for a linear SVM with maximum margin (hard margin formulation) and identify the support vectors.

The support vectors are the points at (4, 3) and (6, 5)

b. [2 points] If a black circle is added as a training sample in the position (7,5), does this affect the previously learned decision boundary? Explain why.

No, this does not affect the decision boundary. The point at (7,5) is beyond the previously learned marigins, and the data is still linearly seperable with it, so they would not affect the position of the hyperplane.

c. [2 points] If a yellow circle is added as a training sample in the position (4,2), does this affect the previously learned decision boundary? Explain why.

No, this does not affect the decision boundary. The point at (7,5) is beyond the

previously learned marigins, and the data is still linearly seperable with it, so they would not affect the position of the hyperplane.

d. [2 points] If a black circle is added as a test sample in the position (7,5), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
Yes. The black circle support vector is on the same side of the hyperplane as that test black circle, so the test black circle would be classified correctly

e. [2 points] If a black circle is added as a test sample in the position (6,4), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
Yes. The black circle support vector is on the same side of the hyperplane as that test black circle, so the test black circle would be classified correctly.

f. [2 points] If a yellow circle is added as a test sample in the position (4,2), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
Yes. The yellow circle support vector is on the same side of the hyperplane as that test yellow circle, so the test black circle would be classified correctly.

g. [2 points] If a yellow circle is added as a test sample in the position (5,3), will this sample be classified correctly according to the previously learned decision boundary? Explain
Yes. The yellow circle support vector is on the same side of the hyperplane as that test yellow circle, so the test black circle would be classified correctly.

h. [2 points] If a black circle is added as a test sample in the position (5,3), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
No. The black circle support vector is on a differnt side of the hyperplane then that test black circle, so the test black circle would be classified incorrectly.

i. [2 points] If a yellow circle is added as a test sample in the position (6,4), will this sample be classified correctly according to the previously learned decision boundary? Explain why.
No. The yellow circle support vector is on a differnt side of the hyperplane then that test yellow circle, so the test yellow circle would be classified incorrectly.

j. [2 points] If a black circle is added as a training sample in the position (4,4), how this will affect the decision boundary if C = 1 and C = ∞? Consider the soft margin formulation.
C = 1: The margin would be huge. Some instances would probably end up within the margin. I think the decision boundary would shift slightly, because that new point's slack variable would be considered in the minimizaiton function.
C = infinity: This would dirrastically shift the decision boundary because the point (4,4) would become a support vector. In addition, the margin would be narrower.

2. [20 points] Consider the following 1-dimensional data with two classes:

x	-3	0	1	2	3	4	5
Class	$-$	$-$	$+$	$+$	$+$	$+$	$+$

a. [5 points] Find the decision boundary of a linear SVM on this data (hard-margin formulation) and identify the support vectors (write the x coordinate to provide your answer).

The support vectors would be at x = 0 and x = 1. The decision boundary is x = 0.5

b. [5 points] Find the solution parameters w and b for this linear SVM and the width of the margin. Hint: place the identified support vectors (positive and negative) into the formula $y_i(w.x_i + b) = 1$ since you know this formula holds for them.

-1(w.0 + b) = 1 => b = -1
1(w.1 + b) = 1 => w + b = 1 => w = 2

The decision boundary is
(2.x + -1) = 0

c. [5 points] Show mathematically that the SVM classifications for the test data {-1.5, 1.5} are negative and positive respectively.

For -1.5:
(2.-1.5 + -1) = -4 => 4 < 0
(2.1.5+-1) = 2 => 2 > 0

d. [5 points] Suppose we remove the point (1,+) from this training set and train the SVM again. Find the new values of the solution parameters w and b and the width of the margin.

-1(W.0 + B) = 1 => **B = -1**
1 (W.2 + B) = 1 => W.2 = 1-B => W.2 = 2 => **w = 1**

The decision boundary is
(1.x + -1)=0

d = 2/||w|| = 2/1 = **2 = d**

3. [20 points] The quadratic kernel $K(x, y) = (x.y + 1)^2$ should be equivalent to mapping each x into a six-dimensional space where

$$\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

for the case where $x = (x_1, x_2)$. Demonstrate this equivalence by answering the following questions while using the data points: A = (1,2), B = (2,4).

a. [5 points] $\Phi(A)$ **= (1, 4, sqrt(2)2, sqrt(2), sqrt(2)2, 1)**
b. [5 points] $\Phi(B)$ **= (4, 16, sqrt(2)8, sqrt(2)2, sqrt(2)4, 1)**
c. [5 points] $\Phi(A)\Phi(B)$ **= 4 + 64 + 32 + 4 + 16 + 1 = 121**
d. [5 points] K(A, B) **= (A.B + 1)^2 = (2 + 8 + 1)^2 = 121**
Hint: your answers for (c) and (d) should be the same. By using the kernel function, SVM "cheats" and performs significantly fewer calculations (kernel trick).

4. [15 points] Complete the Python program (svm.py) that will read the file optdigits.tra (3,823 samples) that includes training instances of handwritten digits (optically recognized). Read the file optdigits.names to get detailed information about this dataset. Also, check the file optdigits-orig.tra and optdigits-orig.names to see the original format of this data, and how it was transformed to speed-up the learning process (pre-processing phase). Your goal is to build multiple SVM classifiers using this data. You will simulate a grid search, trying to find which combination of four SVM hyperparameters (c, degree, kernel, and decision_function_shape) leads you to the best prediction performance. To test the accuracy of those distinct models, you will use the file optdigits.tes (1,797 samples).

ML/blob/25a30fdbb322fe51ed5cc687171e74fd11905c81/Assignment%203/svm.py

5. [25 points] Consider the dataset below.

Outlook	Temperature	PlayTennis
Sunny	Hot	No
Overcast	Cool	Yes
Overcast	Hot	Yes
Rain	Cool	No
Overcast	Mild	Yes

We will apply *steady state* Genetic Algorithms (r = 0.5) to solve this classification problem. If the instance matches the rule pre-condition of the chromosome, you predict according to its post-condition, otherwise you predict the opposite class defined by the chromosome (there must be a prediction for each instance then). Follow the planning below to build your solution.

- Representation: single if-then rule by using bit strings (binary encoding).
    - Outlook $<Sunny, Overcast, Rain>$
    - Temperature $<Hot, Mild, Cool>$

    - Examples:
    - $\text{Outlook} = Overcast \rightarrow$ 010
    - $\text{Outlook} = Overcast \vee \text{Rain} \rightarrow$ 011
    - $\text{Outlook} = Sunny \vee Overcast \vee \text{Rain} \rightarrow$ 111
    - $\text{Outlook} = (Overcast \vee Rain) \wedge (\text{Temperature} = Hot) \rightarrow$ 011100
    - $\text{Outlook} = Sunny \wedge \text{Temperature} = Hot$ then $\text{PlayTennis} = yes \rightarrow$ 1001001
- Initial population (Chromosomes) : (C_1=1001001, C_2=0100101, C_3=1011000, C_4=1101100). Population size should remain the same (4 individuals) over time.
- Fitness function: accuracy
- Penalty criterion: no penalty.
- Selection method: The best two chromosomes are carried over to the next generation. The other two are selected for crossover by using the roulette wheel (simulation).
- Crossover strategy: single-point crossover with mask 1110000 in the 1st generation, two-point crossover with mask 0001100 in the 2nd generation. Use the following chromosomes to perform crossover (simulating the process of spinning the roulette wheel) according to the relative fitness (sectors of a roulette wheel), generating two offspring for each crossover:
    - (1st and 3rd) in the 1st generation
    - (1st and 2nd) in the 2nd generation
- Mutation: on the 6th bit of the chromosome(s) 1011000 selected/produced during the 2nd generation.
- Termination criteria: accuracy = 1.0. **Return the corresponding chromosome(s) – this will be your model.**

Solution:

Fitness(C_1) = 1/5
Fitness(C_2) = 3/5 to be kept
Fitness(C_3) = 4/5 to be kept
Fitness(C_4) = 2/5

1st generation (C_1 =1001001, C_2 = 0100101, C_3 = 1011000, C_4 = 1101100):

Pr(C_1) = .2 / (.6 + .8 + .4 + .2) = 0.1 (4^{th})
Pr(C_2) = .6 / (.6 + .8 + .4 + .2) = 0.3 (2^{nd})
Pr(C_3) = .8 / (.6 + .8 + .4 + .2) = 0.4 (1^{st}) selected
Pr(C_4) = .4 / (.6 + .8 + .4 + .2) = 0.2 (3^{rd}) selected

Mask: 1110000
C_3 = 1011000 →crossover→ C_5 = 1011100
C_4 = 1101100 →crossover→ C_6 = 1101000

Fitness(C_2) = .6
Fitness(C_3) = .8 to be kept
Fitness(C_5) = 4/5 = .8 to be kept
Fitness(C_6) = 3/5 =.6

2nd generation (C_2 = 0100101, C_3= 1011000, C_5 = 1011100, C_6 = 1101000)

Pr(C_2) = .6 / (.6 + .8 + .8 + .6) = 0.214 (3^{rd})
Pr(C_3) = .8 / (.6 + .8 + .8 + .6) = 0.285 (1st) selected
Pr(C_5) = .8 / (.6 + .8 + .8 + .6) = 0.285 (2^{nd}) selected
Pr(C_6) = .6 / (.6 + .8 + .8 + .6) = 0.214 (4^{th})

Mask: 0001100
C_3= 1011000 → C_7 = 1011000
C_5 = 1011100 → C_8 = 1011100

Applying mutation on 1011000 (C3 =1011000 → C9 = 1011010)

Fitness(C5) = .8
Fitness(C_7) = .8
Fitness(C_8) = .8
Fitness(C9) = 1 Final answer: c9 = 1011010

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!