Linear Optimization Assignment #2

Due: Sunday, April 15, 23:59:59

Instruction: Write a report and complete code. Download the code from ftp (10.13.71.168). Upload report and code to ftp (10.13.72.84).

- Please name the report as hw2_31xxxxxxxx.pdf and use pdf format; Please name the compressed file as hw2_31xxxxxxxx.zip or hw2_31xxxxxxxx.rar. And put your name into report.
- Upload:
  - Address: 10.13.72.84
  - Username: opt; Passwd: opt18; Port: 21
- Download:
  - Address: 10.13.71.168
  - Username: opt; Passwd: opt18; Port: 21

Problem 1 Short Answers

(a) What is the advantage of using cross-validation over splitting a dataset into dedicated training and test sets? When is that less important?

(b) Describe 3 optimization tricks for speeding up learning in multi-layer perceptron training for a fixed error function and network design.

(c) Describe the training process of RBF shortly.

Problem 2 Linear Separability

Consider the following two sets of points: $C_1 = \{(0,0), (-1,1), (1,1)\}$, $C_2 = \{(0,2), (-2,0), (2,0)\}$.

(a) Are these points linearly separable? Why or why not?

(b) Design a MLP that can separate them and plot its decision boundaries.

(c) Design a RBF net that can separate them and plot its decision boundaries.

Problem 3 Duality

(a) Show that the dual LP of $\min\{b^T y; A^T y = c, y \geq 0\}$ is $\max\{c^T x; Ax \leq b\}$.

(b) ***Lagrangian relaxation of Boolean LP*** A Boolean linear program is an optimization problem of the form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, n \end{aligned} \tag{1}$$

and is, in general, very difficult to solve. Relax $x_i \in \{0,1\}$ to $0 \leq x_i \leq 1$, we get LP relaxation of Boolean LP. Relax $x_i \in \{0,1\}$ to $x_i(1-x_i) = 0$ and find its lagrangian dual, we get lagrangian relaxation of this problem.

- Derive the dual of LP relaxation
- Derive the Lagrangian relaxation, *i.e.*, the dual of $x_i(1-x_i) = 0$ relaxation.
- (Bonus, you can choose to skip this question) prove the optimal value for LP relaxation and lagrangian relaxation are the same.

Hint for bonus:

- Derive and use the dual of LP relaxation, since LP satisfies strong duality.
- standard form convex problem is equivalent to its epigraph form, *i.e.*

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, n \\ & Ax = b \end{aligned} \tag{2}$$

  equivalent to

$$\begin{aligned} \min_{x,t} \quad & t \\ \text{s.t.} \quad & f(x) - t \leq 0 \\ & g_i(x) \leq 0, \quad i = 1, \ldots, n \\ & Ax = b \end{aligned} \tag{3}$$

- To minimize over multiple variables, we can first minimize one variable.

(c) l_2 **norm soft margin SVMs** If our data is not linearly separable, then we can modify our support vector machine algorithm by introducing an error margin that must be minimized. Specifically, the formulation we have looked at is known as the l_1 norm soft margin SVM. In this problem we will consider an alternative method, known as the l_2 norm soft margin

SVM. This new algorithm is given by the following optimization problem (notice that the slack penalties are now squared):

$$\begin{aligned} \min_{w,b,\varepsilon} \quad & 1/2||w||^2 + C/2\sum_{i=1}^{m} \varepsilon_2^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \quad i = 1, \ldots, m \end{aligned} \tag{4}$$

What is the Lagrangian of the l_2 soft margin SVM optimization problem? What is the dual of the l_2 soft margin SVM optimization problem?

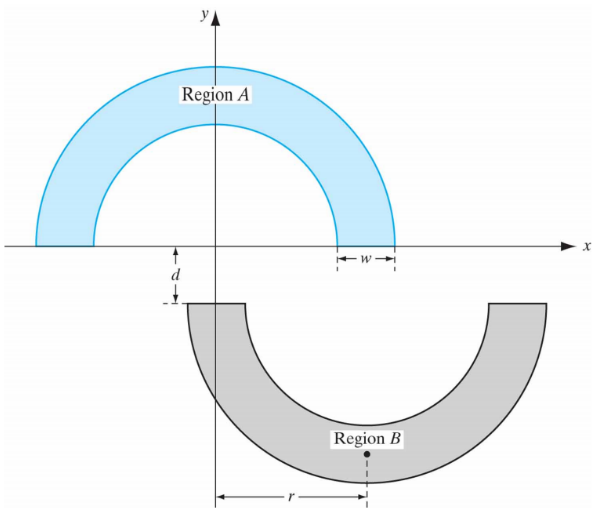

Figure 1: Double moon

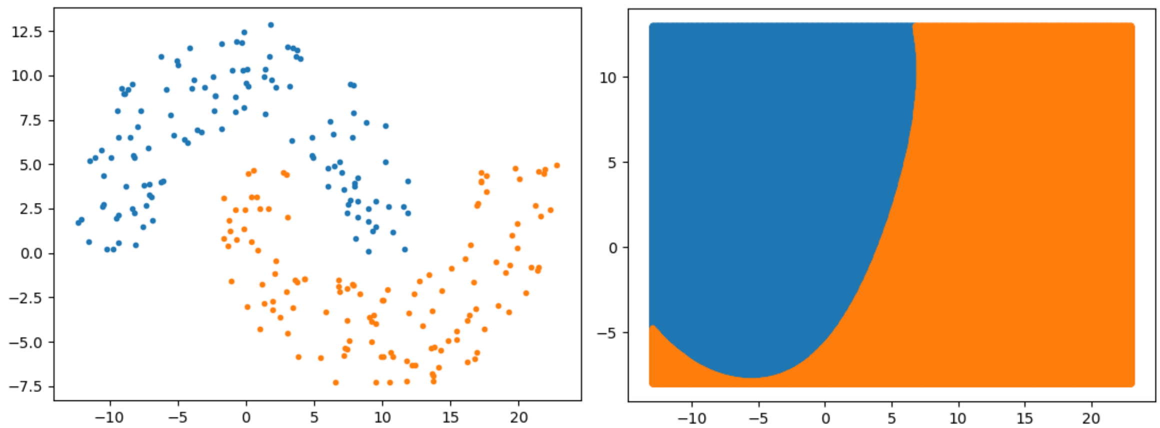

Figure 2: Results. Note that: the results are got from random seed = 16. **Left:** Training data. **Right:** Decision boundary.

Problem 4 RBF on Double Moon

Double Moon data is not linearly separable, ref to fig 1.

- Run 'main.py', the code is runnable and supposed to be bug free. If success, you can see fig 2. What you need to do is to answer the question and improve the code.
- Read the code, the code is not commented, you need to understand it by yourself. RBF in the code uses mean square loss and least square method to calculate weight and bias. Please improve code, *i.e.*, do one or more of the following:
  - add l_2 regularizetion
  - gradient based optimization method

  - use logistic regression to calculate weight and bias. (Thus you may have to implement gradient based optimization method)

- (Bonus: answer one or more questions shown below) Do something extra surrounding the topics in this assignment, using the code you developed. For example, is there some other interesting question we could have asked? Is there any insightful visualization you can plot?

  Explain the principle of function 'cal_distmat', Profile and compare with other potential implementation of 'cal_distmat'.

  Is the code robust to all exceptions and/or elegant with enough documents/comments? May comment for it and describe the training process of RBF.

  Using 'np.ndarray' maybe lengthy, may try 'np.matrix' instead?

  How to treat bias as weight by $[w^T, b]^T$ notation in gradient based optimization method?

  What would happen if 'train_pnts' is not shuffled?

  How about using other hyperparameter**s** (*e.g.* 'n_clusters')?

  Can you implement kmeans from scratch? *et. al.*