CS 542: Machine Learning **Summer 2020**

Problem Set 4

Lecturer: Prof. Peter Chin *Due: June 21, 2020*

- ◇ Please submit your solutions for the written questions (either type up your answers or scan your handwritten solution) in a single PDF on Gradescope under the assignment named "PS4" by 23:59PM on the due date. In your PDF submission for written questions, make sure you have a single PDF with FIVE pages in total as specified in the problem prompts below.
- ◇ Submit your programming assignment in a separate assignment named "PS3 Programming" by the same due date in a zip file.
- ◇ Late policy: there will be a penalty of 10% per day, up to TWO days late. After that no credit will be given. The written part and programming part are considered as a whole when calculating late penalties.

1. (50 points) Written Problems

(a) (10 points, Page 1) The linear regression error function (3.12) can be written

$$E_D(\mathbf{w}) = \frac{1}{2}\left(\mathbf{t} - \Phi\mathbf{w}\right)^{\mathrm{T}}\left(\mathbf{t} - \Phi\mathbf{w}\right).$$

Prove that $\nabla_{\mathbf{w}} E_D(\mathbf{w}) = -\Phi^{\mathrm{T}}\left(\mathbf{t} - \Phi\mathbf{w}\right)$.

(b) (10 points, Page 2) Bishop 6.2

(c) (10 points, Page 3) Bishop 7.3

(d) (10 points, Page 4) Bishop 7.4

(e) (10 points, Page 5) Bishop 7.5

2. (50 points) Support Vector Machine

We've attached a dataset, `MNIST_data.mat`, containing a sample of the famout MNIST benchmark[1]. Your report must provide summaries of each method's performance and some additional details of your implementation. Compare the relative strengths and weaknesses of the methods based on both the experimental results and your understanding of the algorithms.

You can load the data with `scipy.io.loadmat`, which will return a Python dictionary containing the test and train data and labels.

The purpose of this assignment is for you to implement the SVM. You are not allowed to import an SVM from, for instance, `scikit-learn`. You may, however, use a library (such as `scipy.optimize.minimize` or `cvxopt.solvers.qp`) for the optimization process.

[1] `http://yann.lecun/com/exdb/mnist`

Here are your instructions:

(a) Develop code for training an SVM for binary classification with nonlinear kernels. You'll need to accomodate non-overlapping class distributions. One way to implement this is to maximize (7.32) subject to (7.33) and (7.43). It may be helpful to redefine these as matrix operations. Let $\mathbb{1} \in \mathbb{R}^{N\times 1}$ be the vector whose entries are all 1's. Let $\mathbf{a} \in \mathbb{R}^{N\times 1}$ have entries a_i. Let $\mathbf{T} \in \mathbb{R}^{N\times N}$ be a diagonal matrix with $\mathbf{T}_{ii} = t_i$ on the diagonal. Then we can reformulate the objective to be

$$\text{maximize} \quad \tilde{L}(\mathbf{a}) = \mathbb{1}^{\mathrm{T}}\mathbf{a} - \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{TKTa}$$

subject to

$$\begin{aligned} \mathbb{1}^{\mathrm{T}}\mathbf{a} &\preceq C \\ \mathbb{1}^{\mathrm{T}}\mathbf{a} &\succeq 0 \\ \mathbf{a}^{\mathrm{T}}\mathbf{t} &= 0. \end{aligned}$$

The "$\preceq$" symbol here means element-wise comparison. This formulation is very close to what `cvxopt` expects.

(b) Develop code to predict the $\{-1, +1\}$ class for new data. To use the predictive model (7.13) you need to determine b, which can be done with (7.37).

(c) Using your implementation, compare multiclass classification performance of two different voting schemes:

i. one versus rest

ii. one versus one

(d) The parameter $C > 0$ controls the tradeoff between the size of the margin and the slack variable penalty. It is analogous to the inverse of a regularization coefficient. Include in your report a brief discussion of how you found an appropriate value.

(e) In addition to calculating percent accuracy, generate multiclass confusion matrices[2] as part of your analysis.

[2] `https://en.wikipedia.org/wiki/Confusion_matrix`