MA4270: Computational Exercise

There are three problems in this computational exercise, which comprises 10% of your final grade.

Detailed instructions are as follows:

1. This exercise is to be done individually.
2. It is due on **March 5, 2018** in class (Monday after recess week). Please print your report and turn it in in class.
3. You are allowed to use **any programming language**. From our experience, MATLAB, Python, or R will make your life easy. We have provided some skeleton code for guidance:
   - Python 3.x – `MA4270_CE_Python.ipynb`
     (Visit `www.anaconda.com/download/` to get Python and Jupyter for your platform)
   - MATLAB – `MA4270_CE_MATLAB.zip`
4. You are allowed to use **optimization software** such as the optimization toolbox in MATLAB (e.g., the `quadprog` function in MATLAB) or Python optimization libraries (e.g. `CVXOPT`, `quadprog`).
5. To ensure that this is an educational and meaningful exercise, please **do not** resort to using SVM packages (such as `scikit-learn`). Please turn in your code as an appendix to your report.
6. To report any bugs or questions, please contact **both** Vincent Tan (vtan@nus.edu.sg) and Timothy Wee (timothywee@u.nus.edu). We will endeavour to respond as quickly as possible.
7. This exercise will take some time, so please start on it as soon as possible.

Problem 1: Perceptron

You have a two-dimensional, linearly separable dataset, with twenty samples and corresponding labels (Problem1.csv). The samples have already been normalised such that the length of each sample is one. You are required to do the following.

1. Plot the dataset (using different colors/markers for the two different classes) to show it is linearly separable. You may find the information on these websites[1] useful for plotting.

2. Find θ^* (the optimal θ; please normalise it to be of length one) and corresponding
$$\gamma^* := \min_{t\in\{1,\dots,n\}} y_t\langle\theta^*, x_t\rangle$$
using a linearly separable (primal) SVM without offset and without slack variables. To do this, you can use the `quadprog` function in MATLAB or `quadprog` module in Python. Again, please be reminded not to use any SVM package.

3. Write your code for the standard perceptron algorithm, and

   (a) Using the zero vector as the starting point (i.e., $\theta^{(0)} = 0$), report the number of updates required to successfully classify the dataset, the converged solution θ (also normalise it to be of length one), and the corresponding
   $$\gamma := \min_{t\in\{1,\dots,n\}} y_t\langle\theta, x_t\rangle.$$
   Use the natural order, i.e., update your parameter vector by cycling through the points in the order $\{1, 2, 3, \dots\}$.

   (b) Run your code 10 times with different starting points and report your results. What do you observe about the results? In particular, are the parameter vectors obtained the same?

   (c) Suppose the starting point $\theta^{(0)}$ is drawn from the boundary of the unit circle. Prove that the number of iterations required to successfully classify the dataset in terms of the starting point $\theta^{(0)}$ is upper bounded as
   $$k \le \max\left\{-\frac{a}{\gamma^*}, \frac{1 - 2a\gamma^* + \sqrt{(2a\gamma^* - 1)^2 - 4(\gamma^*)^2(a^2 - 1)}}{2(\gamma^*)^2}\right\}.$$
   where $a = \langle\theta^*, \theta^{(0)}\rangle$ and γ^*, θ^* are defined as in part 2.
   Hint: The proof is very similar to that in lec2.pdf of the MIT (Jaakkola) lecture notes. Note that the starting point is not zero vector and you need to modify equations (3) and (8) in that proof.

   (d) Run your code 10000 times with starting points randomly drawn from the unit circle. You can refer to this website[2] to find out how to sample points uniformly at random from the unit sphere. Plot the results of the first 100 runs by comparing the number of updates required to successfully classify the dataset (i.e. achieve zero training error) and the corresponding theoretical bound. Report the average of the number of updates over the 10000 required to successfully classify the dataset and the average of the γ's. Compare this average with the γ^* derived from part 2.

4. Change the label of the first sample to -1 and change the label of the third sample to $+1$.

   - Plot the dataset to show it is not linearly separable.
   - Run part 2 again. Can we find a feasible solution from the SVM without offset and without slack variables?
   - Run the standard perceptron algorithm with any initial vector. Does the standard perceptron algorithm converge?

[1] MATLAB: https://www.mathworks.com/help/stats/kmeans.html
Python: https://matplotlib.org/devdocs/api/_as_gen/matplotlib.pyplot.plot.html
[2] http://mathoverflow.net/questions/24688/efficiently-sampling-points-uniformly-from-the-surface-of-an-n-sphere

Problem 2: Hard-SVM

You have an appropriately selected subset of the iris dataset (iris1.csv). It is a two-dimensional, linearly separable dataset, with 100 samples and corresponding labels.

1. Plot the dataset to show it is linearly separable;
2. Solve the *primal* problem of the SVM for this linearly separable dataset, and report the optimal θ, optimal θ_0 and the optimal objective function value $\frac{1}{2}\|\theta\|_2^2$;
3. Plot the line $\theta^T x + \theta_0 = 0$ and indicate all the support vectors;
4. Solve the *dual* problem of the SVM (without slack variables) for this linearly separable dataset. Report the non-zero (larger than 10^{-6}) entries of the optimal $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ vector and their indices and report the optimal objective function value for the dual problem, i.e.,

   $$\sum_{t=1}^{n} \alpha_t - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

   What do you observe? In particular, is the dual optimal objective value the same as the primal? Which problem (dual or primal) takes a shorter time to solve? Why?
5. Calculate

   $$\sum_{i=1}^{n} \alpha_i y_i x_i$$

   By arbitrarily picking a $j \in \{1, 2, \ldots, n\}$ which satisfies $\alpha_j > 0$, calculate as well

   $$y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j.$$

   What do you observe? In particular, what are the two terms in the displayed equations above equal to?

Problem 3: Soft-SVM and Cross-validation

You have another appropriately selected subset of the iris dataset (iris2.csv). It is a four-dimensional, not linearly separable dataset, with 100 samples and corresponding labels.

1. Solve the relaxed (primal) SVM problem by setting $C = 100$. Report the optimal θ, optimal θ_0 and the optimal objective function value

   $$\frac{1}{2}\|\theta\|_2^2 + C \sum_{t=1}^{n} \xi_t.$$

   Report the number of misclassified samples.
2. Now we are going to explore using cross-validation to find a good C. By 10-*fold cross-validation*, we mean that in the first run, we use the first 10 samples as validation set, and other 90 samples as training dataset; then we use the next 10 samples as validation set, and other 90 samples as training dataset and so on. Finally, we use the last 10 samples as validation set, and other 90 samples as training dataset. Then we average the resultant 10 test errors.

   There are three candidates for $C \in \{1, 100, 10000\}$. For each fixed C, use 10-fold cross validation and report the misclassified rate, i.e., the averages of the 10 test errors. Thus, find the best choice of C among these three values.