COM S 573: Machine Learning
Homework #2

---

1. Please put required code files and report into a compressed file "HW#_FirstName_LastName.zip"
2. Unlimited number of submissions are allowed on Canvas and the latest one will be graded.
3. No later submission is accepted.
4. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
5. All students are required to typeset their reports using latex. Overleaf (`https://www.overleaf.com/learn/latex/Tutorials`) can be a good start.

---

1. (20 points) Consider the toy data set $\{([0,0],-1),([2,2],-1),([2,0],+1)\}$. Set up the dual problem for the toy data set. Then, solve the dual problem and compute α^*, the optimal Lagrange multipliers. (Note that there will be three weights $\boldsymbol{w} = [w_0, w_1, w_2]$ by considering the bias.)

2. (15 points) In a separable case, when a multiplier $\alpha_i > 0$, its corresponding data point $(\boldsymbol{x}_i, y_i)$ is on the boundary of the optimal separating hyperplane with $y_i(\boldsymbol{w}^T\boldsymbol{x}_i) = 1$.

   Show that the inverse is not True. Namely, it is possible that $\alpha_i = 0$ and $(\boldsymbol{x}_i, y_i)$ is on the boundary satisfying $y_i(\boldsymbol{w}^T\boldsymbol{x}_i) = 1$.

   [Hint: Consider a toy data set with two positive examples at ([0,0], +1) and ([1, 0], +1), and one negative example at ([0, 1], -1).] (Note that there will be three weights $\boldsymbol{w} = [w_0, w_1, w_2]$ by considering the bias.)

3. (15 points) **Non-separable Case SVM:** In Lecture 8 (page 18), we compare the hard-margin SVM and soft-margin SVM. Prove that the dual problem of soft-margin SVM is almost identical to the hard-margin SVM, except that α_is are now bounded by C (tradeoff parameter).

4. (20 points) **Kernel Function:** A function K computes $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\boldsymbol{x_i}^T\boldsymbol{x}_j$. Is this function a valid kernel function for SVM? Prove or disprove it.

5. (15 points) **Support Vectors:** In the **linearly separable** case, prove that we need at most $d+1$ support vectors for a naive SVM, where d is the number of features. (Naive SVM means we use a hard-margin SVM with "linear" kernel.)

6. (15 points) **Support Vector Machine for Handwritten Digits Recognition**: You need to use the software package scikit-learn https://scikit-learn.org/stable/modules/svm.html to finish this assignment. We will use "svm.SVC()" to create a svm model. The handwritten digits files are in the "data" folder: train.txt and test.txt. The starting code is in the "code" folder. In the data file, each row is a data example. The first entry is the digit label ("1" or "5"), and the next 256 are grayscale values between -1 and 1. The 256 pixels correspond to a 16×16 image. You are expected to implement your solution based on the given codes. The only file you need to modify is the "solution.py" file. You can test your solution by running "main.py" file. Note that code is provided to compute a two-dimensional feature (symmetry and average intensity) from each digit image; that is, each digit image is represented by a two-dimensional vector. These features along with the corresponding labels should serve as inputs to your solution functions.

(a) (5 points) Complete the **svm_with_diff_c()** function. In this function, you are asked to try different values of cost parameter c.

(b) (5 points) Complete the **svm_with_diff_kernel()** function. In this function, you are asked to try different kernels (linear, polynomial and radial basis function kernels).

(c) (5 points) Summarize your observations from (a) and (b) into a short report. In your report, please report the accuracy result and total support vector number of each model. A briefly analysis based on the results is also needed. For example, how the number of support vectors changes as parameter value changes and why.2