1. **Vector Norms.** Recall the definitions of vector norms.

\[
\forall x \in \mathbb{R}^d, p \geq 1, ||x||_p := \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}.
\]

Show the following inequalities for any vector \(x \in \mathbb{R}^d\).

(a) \(||x||_2 \leq ||x||_1 \leq \sqrt{d}||x||_2\).

(b) \(||x||_\infty \leq ||x||_2 \leq \sqrt{d}||x||_\infty\).

(c) \(||x||_\infty \leq ||x||_1 \leq d||x||_\infty\).

(d) \(\forall p \geq 1, ||x||_\infty \leq ||x||_p \leq d^{1/p}||x||_\infty\).

(e) \(\lim_{p \to \infty} ||x||_p = ||x||_\infty\).

2. **Matrix Norms.** Show that the following hold for any matrices \(A\) and \(B\).

(a) Recall the *Frobenius* norm of a matrix \(A\):

\[
||A||_F := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2}.
\]

Show that,

\[
||AB||_F \leq ||A||_F \cdot ||B||_F,
\]

whenever, \(A\) and \(B\) are compatible.

(b) Recall the *spectral* norm of a matrix:

\[
||A||_2 := \sup_{||x||_2 \leq 1} ||Ax||_2.
\]

Show that,

\[
||A||_2 = \sqrt{\lambda_{\text{max}}(A^T A)},
\]

where \(\lambda_{\text{max}}(M)\) is the maximum eigenvalue of a square matrix \(M\).

(c) Let \(A \in \mathbb{R}^{n \times p}\). Show that

\[
||A||_F = \sqrt{tr(A^T A)} = \sqrt{\sum_{i=1}^{n} \sigma_i(A)^2},
\]

where \(tr(M) := \sum_{i=1}^{p} M_{ii}\) for any square matrix \(M \in \mathbb{R}^{n \times n}\) and \(\sigma_i(A) := \text{ith largest singular value of matrix } A\).

3. Let \(H\) be a hyperplane in \(\mathbb{R}^d\) whose equation is given by

\[
wx + b = 0,
\]

for some normal vector \(w \in \mathbb{R}^d\) and offset \(b \in \mathbb{R}\). Let \(d_p(x, H)\) denote the \(p-\text{norm}\) distance of \(x\) to the hyperplane \(H\), that is,

\[
d_p(x, H) = \inf_{x' \in H} ||x' - x||_p.
\]

Then, show that the following holds \(\forall p \geq 1\),

\[
d_p(x, H) = \frac{|w^T x + b|}{||w||_q},
\]

where \(q\) is the conjugate of \(p\), that is, \(\frac{1}{p} + \frac{1}{q} = 1\).

4. Assume \(h : \mathbb{R} \to \mathbb{R}\) and \(g : \mathbb{R}^d \to \mathbb{R}\) are twice differentiable functions and for all \(x \in \mathbb{R}^d\), define \(f(x) = h(g(x))\). Then show that the following implications are valid:

(a) \(h\) is convex and non-increasing, and \(g\) is concave \(\Rightarrow f\) is convex.

(b) \(h\) is concave and non-decreasing, and \(g\) is concave \(\Rightarrow f\) is concave.

(c) \(h\) is concave and non-increasing, and \(g\) is convex \(\Rightarrow f\) is concave.

5. **Support Vector Machines.** Consider the following training data \((x_1, x_2) \in \mathbb{R}^2\),
(a) Plot these six training points. Are the classes \{+, −\} linearly separable?

(b) Construct the weight vector of the maximum margin hyperplane by inspection and identify the support vectors.

(c) If you remove one of the support vectors does the size of the optimal margin decrease, stay the same, or increase?

6. **Non-Linearly Separable Data – Soft Margin SVM**

Recall, in class we discussed the primal form of SVMs with slack variables,

$$\min_{w, b, \xi} \frac{1}{2}||w||^2 + C \sum_{i=1}^{m} \xi_i$$

\[
\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m,
\]

where \(C \geq 0\) is a constant chosen to trade off the simultaneous goals of maximizing the margin and minimizing the margin violation.

(a) If \(C = 0\), What is the solution of the above optimization problem and what is its optimal value?

(b) How can we set \(C\) to make this problem essentially equivalent to that with the linearly separable data without slack variables?

(c) Write the Lagrangian of this optimization problem for SVMs with slack variables. Use \(\{\alpha_i\}_{i=1}^{m}\) to denote Lagrangian multipliers and derive the dual. Explicitly state how do you derive the dual.

(d) Compute \(w\) and \(b\) using the dual formulation.

(e) As you just showed, we can compute the optimal separating hyperplane using the solution to the dual problem. Only a subset of the training examples \(\{x_i\}_{i=1}^{m}\) are actually used to compute \(w\) and \(b\). Find those examples in terms of the Lagrange multipliers \(\{\alpha_i\}_{i=1}^{m}\) and rewrite the expressions of \(w\) and \(b\) using those. [Hint. These examples are called Support Vectors.]

(f) Where do these examples – found in part (d) – lie with respect to the separating hyperplane and two margin hyperplanes, and what can you state about the values of their corresponding slack variables \(\{\xi_i\}_{i=1}^{m}\) and Lagrange multipliers \(\{\alpha_i\}_{i=1}^{m}\)? What can you say about the remaining examples in the training set? [Hint. If possible, break your answers for \(\alpha_i = 0, 0 < \alpha_i < C, \alpha_i = 1\) and \(\alpha_i > C\) respectively]

(g) If we test our model \((w, b)\) on the training examples \(\{x_i\}_{i=1}^{m}\) itself, which all examples will be correctly classified and which all will be misclassified? [Hint. Part (f)]

7. **Programming Exercise: Soft Margin SVM.**

**Dataset:** The [https://github.com/TeachingReps/Machine-Learning/tree/master/Datasets/Spambase](https://github.com/TeachingReps/Machine-Learning/tree/master/Datasets/Spambase) folder contains a spam classification data set drawn from the UCI Machine Learning repository (http://archive.ics.uci.edu/ml/datasets/Spambase) and normalized to have each feature value in the range \([0, 1]\). In the data files provided, each row is a separate training example; the first 57 columns correspond to features, while the last column is the label \((+1/−1)\). There are two files train.txt and test.txt containing 250 training instances and 4351 test instances respectively. The CrossValidation folder contains the train and test data for
each fold of the 5-fold cross-validation procedure on the training set train.txt. In each fold, there are two files cv-train.txt and cv-test.txt containing 200 training instances and 50 test instances respectively.

**Code:** You have been given a PYTHON code https://github.com/TeachingReps/Machine-Learning/blob/master/codes/SVMcode.py to learn an SVM classifier. You need to edit in the PYTHON code ”SVMcode.py” as directed.

**Instructions:**

1. Your program should read the training data, test data, parameter $C$ and print the percentage error of your trained SVM model.

2. Run your implementation of the SVM learning algorithm selecting $C$ from the range \{1, 10, 10^2, 10^3, 10^4\} through 5-fold cross-validation on the training set. Report the average cross-validation error (over 5 folds) for each value of $C$ in this range. [*Note: Use files from CrossValidation folder here.*]

3. Take the best value of $C$ (i.e., the one which achieves the least average cross-validation error) and run your SVM implementation with this choice of $C$. Report the training and test errors achieved by this value of $C$. [*Note: Use train.txt and test.txt files here.*]

4. You need to submit a zip file ”Assignment_1_yourfullname.zip” containing
   - Your edited version of ”SVMcode.py” renamed as ”SVMcode_yourfullname.py”.
   - A text file ”SVMcode_yourfullname.txt” reporting the results obtained in parts 2 and 3.