CSCI 416/516 Homework #2

DUE: March 27, 2026, at 11:59 pm

Submission: For all the problems excluding the multiple choice problem(s), you need to **show all your works, steps, and calculations** if applicable, or **your justification/expalantion to the answer(s) you provide**. You should submit a PDF to Blackboard with your answers that are recognizable/intelligible. Preferably, you should use LaTeX.

- **Problem 1 [2pts]: Support Vector Machine.**
We want to maximize the margin between the cluster of positive samples and the cluster of negative samples using SVM. Suppose the support vectors in the cluster of positive samples fall on the vertical line $\boldsymbol{\theta}^\top \boldsymbol{x}_+ = 1$, and the support vectors in the cluster of negative samples fall on the vertical line $\boldsymbol{\theta}^\top \boldsymbol{x}_- = -1$, what is the formula, expressed in terms of $\boldsymbol{\theta}$, that we want to maximize? Show the steps on how you reached the conclusion.

- **Problem 2 [2pts]: Support Vector Machine.**
The optimization objective of SVM is given as

$$\min_{\boldsymbol{\theta}} C \sum_{i=1}^{N} [y_i \text{cost}_1(\boldsymbol{\theta}^\top \boldsymbol{x}_i) + (1 - y_i)\text{cost}_0(\boldsymbol{\theta}^\top \boldsymbol{x}_i)] + \frac{1}{2} \sum_{j=1}^{d} \theta_j^2 \tag{1}$$

where cost_0 and cost_1 are defined using the hinge loss. Explain the difference between the scenario in which the tunable hyperparameter C is large and the scenario in which C is small - what are we favoring, by making C large or small?

- **Problem 3 [2pts]: Support Vector Machine.**
How did we reach (for that $\sum_i a_j y_j = 0$, s.t. $\alpha_i \geq 0, \forall i$) Equation 2 from Equation 3?

$$\mathcal{J}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \tag{2}$$

$$\frac{1}{2} \sum_{j=1}^{d} \theta_j^2 - \sum_{i=1}^{n} \alpha_i (y_i (\boldsymbol{\theta}^\top x_i + b) - 1) \tag{3}$$

- **Problem 4 [2pts]: Kernels.**
Which of the following is not true about the kernel trick in SVMs?

- A. The kernel function computes the dot product of two vectors in a transformed feature space.
- B. The kernel function always transforms the data to a higher-dimensional space.
- C. The choice of kernel affects the computational complexity of training an SVM.
- D. The kernel function allows SVM to operate in a transformed feature space without explicitly calculating the transformation.

- **Problem 5 [2pts]: Lagrangian Multiplier.**
  Suppose a sample in the training dataset has a Lagrangian multiplier being 0. What does this say about this sample?