CAAM/STAT 31310, Autumn 2024, U Chicago

STAT/CAAM 37781: Homework 1

Due Feb 5, '25 (11:59 pm ET) on Gradescope

Cite any sources and collaborators; do not copy. See syllabus for policy.

In this homework, we implement kernel SVMs for binary classification and do some numerical experiments to understand the effect of the margin.

First recall the setup. We have inputs from a domain $\mathcal{X} \subseteq \mathbb{R}^d$ and a binary function, taking values in $\{-1, 1\}$ defined on $\mathcal{X}$. We know its values at m points. That is, we are given input examples, (x_i, y_i) $x_1, \cdots, x_m \in \mathbb{R}^d$ that are $1 \leq i \leq m$, where $y_i = \{-1, 1\}$ are the binary labels associated with x_i. We can assume (x_i, y_i) are independent and identically distributed (iid) according to a distribution $\mathcal{D}$. Hyperplane classifiers are of the form, $h(x; w, b) = \mathrm{sgn}(w^\top x + b)$, where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learnable parameters. Our task is to learn hyperplane parameters $(w, b) \in \mathbb{R}^{d+1}$ such that $h(x; w, b) \approx y$, the true label of a point x, on average with respect to the unknown distribution $\mathcal{D}$. Or more precisely, the generalization error, defined as $\mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(w, b; x, y)$ is as small as possible for a loss function, $\ell : \mathbb{R}^{d+1} \times (\mathcal{X} \times \{-1, 1\}) \to \mathbb{R}^+$. Since $\mathcal{D}$ is unknown, we try to minimize instead an *empirical risk*, $\sum_{i=1}^m \ell(w, b; x_i, y_i)$, wherein the expectation is approximated with a sample average.

- Now, in the class, we derived the classical SVM problem, under the condition of linear separability, based on the margin maximization principle, and the constraints $y_i(w \cdot x_i + b) \geq 1$. What was the margin in this formulation, in terms of the solutions of the classical SVM, w, b and the inputs? (1 point)
- When points are not linearly separable, the idea of geometric margin that we discussed in class does not apply as such. We can still take the minimum distance only over correctly classified points. Now set $\ell(w, b; x, y) = \max(0, y\ h(x; w, b))$ and write down the corresponding empirical risk minimization problem with the constraint that the geometric margin (defined over correctly classified points) is a fixed $\rho \in \mathbb{R}^+$. (2 points)
- Convexify the constraint from the previous part (1 point), check that the overall optimization for w, b is now convex, and write down the KKT conditions (2 points) and the dual problem (2 points). Hint: notice that this is indeed the same as the soft-SVM.
- Now we will implement the above algorithms on the MNIST dataset (you can work with the USPS dataset if you do not have access to large enough memory/compute). Use $y_i = 1$ if x_i is an image of a number less than or equal to 4, and $y_i = -1$ otherwise. You should see roughly 50% of the data have $y_i = 1$. Split the imported dataset into train and test subsets (feel free to use mnist-kernel-classification.py from class).

Solve the convex optimization you derive above (or soft SVM) using a package for convex optimization, such as cvxpy. You do not need to submit the code. Answer the following questions:

  - How do you choose the maximum number of iterations? (1 points)
  - How did you choose the margin ρ above? (2 points) This is usually chosen with cross-validation. In your answer, you can specify what trend you observe in the test error (sample average of ℓ over test data) with ρ.

- Now implement the soft kernel SVM algorithm with a Gaussian kernel. Again you do not need to submit the code, but answer the following questions:
  - Refer to Chapter 2 of the book by Smola and Scholkopf to understand their kernel SVM implementation. How do you choose the hyperparameter (width/-variance) of the kernel? (2 points)
  - Plot the test error as a function of margin (4 points) and as a function of the width/variance (3 points); explain your plots.

Reminder about the use of language models: feel free to take help for the code or to get ideas for the answers, but do not copy the answers directly. Finally what you submit must be your own words.