CMU 10-715: Homework 2

Soft Support Vector Machine Theory and Implementation

Released: Sept. 14, 2022.

Due: Sept. 23, 2022, 11:59 PM.

Instructions:

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books, again after you have thought about the problems on your own. Please don't search for answers on the web, previous years' homeworks, etc. (please ask the TAs if you are not sure if you can use a particular reference). There are two requirements: first, cite your collaborators fully and completely (e.g., "Alice explained to me what is asked in Question 4.3"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.

- **Submitting your work:** On Gradescope:

  - Submit your completed python file, soft_svm.py, to the assignment titled "HW2 (code) Soft Support Vector Machine Theory and Implementation".
  - Submit your PDF report file, named [your andrew id].pdf , to the assignment titled "HW2 (report) Soft Support Vector Machine Theory and Implementation". Upon submission, please follow Gradescope instructions to match the question numbers with the page numbers of your report.

  There is no limit on the number of submissions to Gradescope.

- **Gradescope access:** The link to this course is `https://www.gradescope.com/courses/428731/`. You should already have access (from HW1).

- **Auto-grader:** Your code will be evaluated with Gradescope auto-grader. We encourage you to start early and use the auto-grader to check that your implementation is correct.

- **Late days:** For each homework you get three late days to be used only when anything urgent comes up. No points will be deducted for using these late days. We will consider an honor system where we will rely on you to use the late days appropriately.

- **Skeleton code:** The python files, data.py and soft_svm.py, can be found at the attachment section of the Diderot post that announces the release of this homework.

1 Soft Support Vector Machine Theory [40 points]

Consider the primal problem for the soft support vector machine (soft SVM). Where $y_i \in \{-1, 1\}$ are the labels, $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$ are the features (features already include the bias term), $\xi_i \in \mathbb{R}^+$ are the slack variables.

$$\begin{aligned} \underset{\mathbf{w}, \xi_i}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n} \xi_i & \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1 - \xi_i & i = 1, \ldots, n \\ & \xi_i \geq 0 & i = 1, \ldots, n \end{aligned} \tag{1}$$

(a) (25 points) Show that the soft SVM problem can be written as a regularized Hinge Loss problem:

$$\underset{\mathbf{w}}{\text{minimize}} \ \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i)\right) \tag{2}$$

(b) (15 points) Find an expression for the subgradient of the regularized Hinge Loss problem from equation (2).

2 Implementation of the Soft SVM [60 points]

Algorithm 1: Gradient Descent

Input: Number of steps T, learning rate α
Initialize parameter $\mathbf{w}_0 = 0$ (weights which already implicitly include the bias)
for $t \in 0, \ldots, T-1$ **do**
  Compute the subgradient $\nabla f(\mathbf{w}_t)$ according to (1.b).
  Update the parameters $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla f(\mathbf{w}_t)$;
end

(a) (5 points) Complete the code of the Soft SVM Loss method. This method computes the regularized Hinge Loss shown in questions (1.a) using all examples. The Hinge Loss term is **summed** across examples, not averaged. Expected input and output sizes are given in code comments.

(b) (10 points) Complete the code of the subgradient method. This method computes the subgradient of the **regularized** hinge loss problem—the quantity that you derived in question (1.b)—using all examples. Expected input and output sizes are given in code comments.

(c) (5 points) Complete the code of the predict method, which predicts labels in $\{-1, 1\}$ for all examples. Expected input and output sizes are given in code comments.

(d) (15 points) Complete the train method of the soft SVM using the gradient descent algorithm. Expected input sizes are given in code comments.

(e) Train your soft SVM classifiers on the data from data.py. You can use get_data to get the data. Notably, this function will automatically append a constant 1 to the start of each x_i, allowing for the inclusion of the bias in the weight vector.

Report the following in your pdf for each of C=0.1, C=1, C=50 (keep random_seed at 1, use 10,000 training iterations, and learning rate 1e-5):

(i) (5 points) Final train and test losses.

(ii) (5 points) Final train and test accuracies.

(iii) (10 points) After training, randomly sample without replacement 10% of all **training** data points and plot them in a 2D scatterplot, using colors and/or shapes to distinguish between:

- Examples with label +1 which *are not* support vectors.
- Examples with label +1 which *are* support vectors.
- Examples with label -1 which *are not* support vectors.
- Examples with label -1 which *are* support vectors.

Provide a legend to describe which type of point is which.

Also plot the final learned decision boundary.

NOTE: Keep in mind that you should plot the 2 actual dimensions of each datapoint, and exclude the constant 1 attached for convenience in the prepare_data function in data.py.

(f) (5 points) Which C gives the least number of support vectors in the sampled subset? Why do you think this is? Explain in 1-2 sentences (you don't need to rigorously prove it).