Assignment #3
Due: 11:59pm March 29, 2019

Homework 3: Max-Margin, Ethics, Clustering

Introduction

This homework assignment will have you work with max-margin methods and clustering, as well as an ethics assignment. The aim of the assignment is (1) to further develop your geometrical intuition behind margin-based classification and decision boundaries, (2) try coding a simple K-means classifier, and (3) to have you reflect on the ethics lecture and to address the scenario discussed in class in more depth by considering the labor market dynamically.

We encourage you to first read the Bishop textbook coverage of these topics, particularly: Section 7.1 (Max-Margin and SVMs) and Section 9.1 (Clustering). Chapters 5 and 6 of the student textbook are also relevant.

There is a mathematical component and a programming component to this homework. Please submit your PDF, tex, and Python files to Canvas, and push all of your work to your GitHub repository. If a question requires you to make any plots, like Problem 2, please include those in the writeup.

Problem 1 (Fitting an SVM by hand, 7pts)

For this problem you will solve an SVM without the help of a computer, relying instead on principled rules and properties of these classifiers.

Consider a dataset with the following 7 data points each with $x \in \mathbb{R}$:

$$\{(x_i, y_i)\}_i = \{(-3, +1), (-2, +1), (-1, -1), (0, +1), (1, -1), (2, +1), (-3, +1)\}$$

Consider mapping these points to 2 dimensions using the feature vector $\phi(x) = (x, -\frac{8}{3}x^2 + \frac{2}{3}x^4)$. The hard margin classifier training problem is:

$$\min_{\mathbf{w}, w_0} \|\mathbf{w}\|_2^2$$
$$\text{s.t.} \quad y_i(\mathbf{w}^\top \phi(x_i) + w_0) \geq 1, \ \forall i \in \{1, \ldots, n\}$$

The exercise has been broken down into a series of questions, each providing a part of the solution. Make sure to follow the logical structure of the exercise when composing your answer and to justify each step.

1. Plot the training data in $\mathbb{R}^2$ and draw the decision boundary of the max margin classifer.

2. What is the value of the margin achieved by the optimal decision boundary?

3. What is a vector that is orthogonal to the decision boundary?

4. Considering discriminant $h(\phi(x); \mathbf{w}, w_0) = \mathbf{w}^\top \phi(x) + w_0$, give an expression for *all possible* $(\mathbf{w}, w_0)$ that define the optimal decision boundary. Justify your answer.

5. Consider now the training problem. Using your answers so far, what particular solution to $\mathbf{w}$ will be optimal for this optimization problem?

6. Now solve for the corresponding value of w_0, using your general expression from part (4.) for the optimal decision boundary. Write down the discriminant function $h(\phi(x); \mathbf{w}, w_0)$.

7. What are the support vectors of the classifier? Confirm that the solution in part (6.) makes the constraints above binding for support vectors.

Solution

Problem 2 (K-Means, 10pts)

For this problem you will implement K-Means clustering from scratch. Using `numpy` is fine, but don't use a third-party machine learning implementation like `scikit-learn`. You will then apply this approach to clustering of image data.

We have provided you with the MNIST dataset, a collection of handwritten digits used as a benchmark of image recogntion (you can learn more about the data set at http://yann.lecun.com/exdb/mnist/). The MNIST task is widely used in supervised learning, and modern algorithms with neural networks do very well on this task.

Here we will apply unsupervised learning to MNIST. You have been given representations of 6000 MNIST images, each of which are 28×28 greyscale handwritten digits. Your job is to implement K-means clustering on MNIST, and to test whether this relatively simple algorithm can cluster similar-looking images together.

The given code loads the images into your environment as a 6000x28x28 array. In your code, you may use the ℓ_2 norm as your distance metric. (You should feel free to explore other metrics than the ℓ_2 norm, but this is strictly optional.)

- Starting at a random initialization and some choice of K, plot the K-means objective function as a function of iteration and verify that it never increases.

- Run the K-means algorithm from several different restarts for different values of K. Plot the final K-means objective as a function of K with errorbars over the random restarts. How does the objective and the variance of the objective change with K?

- For $K = 10$, for a couple of random restarts, show the mean images for each cluster. To render an image, use the pyplot `imshow` function.

- Now, before running K-means, standardize the data. That is, center the data first so that each pixel has mean 0 and variance 1 (except for any pixels that have zero variance). For $K = 10$, for a couple of random restarts, show the mean images for each cluster. Compare them to the previous part.

As in past problem sets, please include your plots in this document. (There may be several plots for this problem, so feel free to take up multiple pages.)

Solution

Solution

- Name:

- Email:

- Collaborators:

- Approximately how long did this homework take you to complete (in hours):