Your Name
email@fas.harvard.edu
CS181-S17

Assignment #3
Due: 5:00pm March 24, 2016

Collaborators: John Doe, Fred Doe

Homework 3: Max-Margin and SVM

Introduction

This homework assignment will have you work with max-margin methods and SVM classification. The aim of the assignment is (1) to further develop your geometrical intuition behind margin-based classification and decision boundaries, (2) to explore the properties of kernels and how they provide a different form of feature development from basis functions, and finally (3) to implement a basic Kernel based classifier.

There is a mathematical component and a programming component to this homework. Please submit your PDF 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 3, 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 $\boldsymbol{\phi}(x) = (x, x^2)$. The max-margin classifier objective is given by:

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

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. Write down a vector that is parallel to the optimal vector $\mathbf{w}$. Justify your answer.
2. What is the value of the margin achieved by $\mathbf{w}$? Justify your answer.
3. Solve for $\mathbf{w}$ using your answers to the two previous questions.
4. Solve for w_0. Justify your answer.
5. Write down the discriminant, $h(x; \mathbf{w}, w_0)$, as an explicit function of x.

Solution

Problem 2 (Composing Kernel Functions , 10pts)

A key benefit of SVM training is the ability to use kernel functions $K(\mathbf{x}, \mathbf{x}')$ as opposed to explicit basis functions $\boldsymbol{\phi}(\mathbf{x})$. Kernels make it possible to implicitly express large or even infinite dimensional basis features. We do this by computing $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ directly, without ever computing $\boldsymbol{\phi}(\mathbf{x})$.

When training SVMs, we begin by computing the kernel matrix $\mathbf{K}$, over our training data $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The kernel matrix, defined as $K_{i,i'} = K(\mathbf{x}_i, \mathbf{x}_{i'})$, expresses the kernel function applied between all pairs of training points.

In class, we saw Mercer's theorem, which tells us that any function K that yields a positive semi-definite kernel matrix forms a valid kernel, i.e. corresponds to a matrix of dot-products under *some* basis $\boldsymbol{\phi}$. Therefore instead of using an explicit basis, we can build kernel functions directly that fulfill this property.

A particularly nice benefit of this theorem is that it allows us to build more expressive kernels by composition. In this problem, you are tasked with using Mercer's theorem and the definition of a kernel matrix to prove that the following compositions are valid kernels, assuming $K^{(1)}$ and $K^{(2)}$ are valid kernels. Recall that a positive semi-definite matrix $\mathbf{K}$ requires $\mathbf{z}^\top \mathbf{K} \mathbf{z} \geq 0, \ \forall \ \mathbf{z} \in \mathbb{R}^n$.

1. $K(\mathbf{x}, \mathbf{x}') = c\, K^{(1)}(\mathbf{x}, \mathbf{x}') \quad \text{for } c > 0$
2. $K(\mathbf{x}, \mathbf{x}') = K^{(1)}(\mathbf{x}, \mathbf{x}') + K^{(2)}(\mathbf{x}, \mathbf{x}')$
3. $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})\, K^{(1)}(\mathbf{x}, \mathbf{x}')\, f(\mathbf{x}') \quad \text{where } f \text{ is any function from } \mathbb{R}^m \text{ to } \mathbb{R}$
4. $K(\mathbf{x}, \mathbf{x}') = K^{(1)}(\mathbf{x}, \mathbf{x}')\, K^{(2)}(\mathbf{x}, \mathbf{x}')$

   [Hint: Use the property that for any $\boldsymbol{\phi}'(\mathbf{x})$, $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}'(\mathbf{x})^\top \boldsymbol{\phi}'(\mathbf{x}')$ forms a positive semi-definite kernel matrix.]
5. (a) The exp function can be written as,

   $$\exp\{x\} = \lim_{i \to \infty} \left(1 + x + \cdots + \frac{x^i}{i!} \right).$$

   Use this to show that $\exp\{xx'\}$ can be written as $\boldsymbol{\phi}'(x)^\top \boldsymbol{\phi}'(x')$ for some $\boldsymbol{\phi}'(x)$. Explain why this $\boldsymbol{\phi}'$ be hard to use as a basis in standard logistic regression.

   (b) Using the previous identities, show that $K(\mathbf{x}, \mathbf{x}') = \exp\{K^{(1)}(\mathbf{x}, \mathbf{x}')\}$ is a valid kernel.
6. Finally use only these identities to prove the validity of the Gaussian kernel:

   $$K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||_2^2}{2\sigma^2} \right)$$

Solution

Problem 3 (Scaling up your SVM solver, 10pts (+3pts with extra credit))

In the previous homework, you studied a simple data set of fruit measurements. Here we would like you to code up a few simple SVM solvers to classify lemons from apples.

- First read the paper at http://www.jmlr.org/papers/volume6/bordes05a/bordes05a.pdf and implement the Kernel Perceptron algorithm and the Budget Kernel Perceptron algorithm. Aim to make the optimization as fast as possible.
- Additionally, we would like you to do some experimentation with the hyperparameters for each of these models. Try seeing if you can identify some patterns by changing β, N (maximum number of support vectors), or the number of random samples you take. Note the training time, accuracy, types of hyperplanes, and number of support vectors for various setups.

Hint: For this problem, efficiency will be an issue. Instead of directly implementing this algorithm using numpy matrices, you should utilize Python dictionaries to represent sparse matrices. This will be necessary to have the algorithm run in a reasonable amount of time.

We are intentionally leaving this open-ended to allow for experimentation, and so we will be looking for your thought process and not a rigid graph this time. That being said, any visualizations that you want us to grade and refer to in your descriptions should be included in this writeup. You can use the trivial $K(x_1, x_2) = x_1^\top x_2$ kernel for this problem, though you are welcome to experiment with more interesting kernels too.

Lastly, compare the classification to the naive SVM imported from scikit-learn. For extra credit (+3 pts), implement the SMO algorithm and implement the LASVM process and do the same as above.

Answer the following reading questions in one or two sentences.

1. In one short sentence, state the main purpose of the paper.
2. Identify each of the parameters in Eq. 1
3. State one guarantee for the Kernel perceptron algorithm described in the paper.
4. What is the main way the budget kernel perceptron algorithm tries to improve on the perceptron algorithm?
5. In simple words, what is the theoretical guarantee of LASVM algorithm? How does it compare to its practical performance?

Solution

Calibration [1pt]

Approximately how long did this homework take you to complete?