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 have you implement a basic Kernel-based classifier and get some experience in implementing a model/algorithm from an academic paper in the field, 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.

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 hard margin classifier training problem is:

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

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 $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0) = \mathbf{w}^\top \boldsymbol{\phi}(x) + w_0$, 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 (1). 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 $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0)$.
7. What are the support vectors of the classifier? Confirm that the solution in part (6.) makes the constraints in (1) binding for support vectors.

Solution 1

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

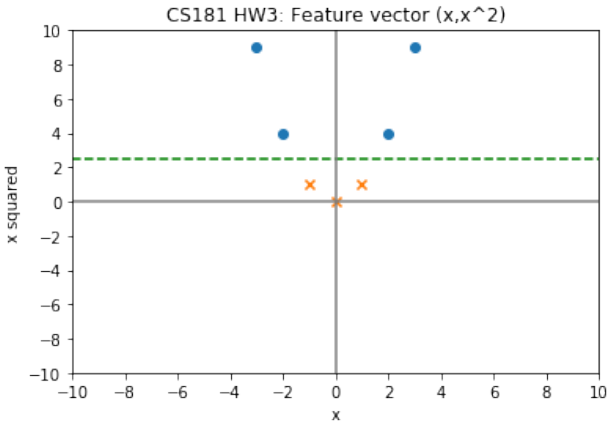

Figure 1

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

   Minimum distance from point to boundary is 1.5

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

   Any vertical vector is. e.g. (0,10). Abstractly, w.

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

   Our decision boundary by visual inspection is at $x^2 = 2.5$. Our decision boundary is $h = \mathbf{w}^\top \boldsymbol{\phi}(x) + w_0$.

$$\mathbf{w}^T \begin{bmatrix} x \\ x^2 \end{bmatrix} + w_0 = 0$$

$$\begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} x \\ x^2 \end{bmatrix} + w_0 = 0$$

$$w_1 x + w_2 x^2 + w_0 = 0$$

Our decision boundary by visual inspection is $x^2 = 2.5$.

Thus we get that

$$w_1 x = 0$$

$$x^2 = -\frac{w_0}{w_2} = 2.5$$

Rewriting in terms of w_2 we get

$$w_0 = -2.5 * w_2$$

For final answer

$$w_2 x^2 - 2.5 \cdot w_2 = 0$$

$$w_0 = -2.5 \cdot w_2$$

Equivalently in matrix form,

$$\begin{bmatrix} 0 & w_2 \end{bmatrix} \begin{bmatrix} x \\ x^2 \end{bmatrix} - 2.5 \cdot w_2 = 0$$

5. Consider now the training problem (1). Using your answers so far, what particular solution to $\mathbf{w}$ will be optimal for this optimization problem?

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

If we plug in one of the points we can see by visual inspection is a support vector (-2,1) and taking the optimal solution to be where the above expression $y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) = 1, \ \forall i$, we get that

$$\begin{aligned} 1(w_2(-2)^2 - w_2\frac{5}{2} &= 1 \\ 4w_2 - 2.5w_2 &= 1 \\ w_2 = \frac{1}{1.5} &= \frac{2}{3} \end{aligned}$$

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 $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0)$.

Given that $w_2 = 2.5$, we get that

$$\begin{aligned} w_2 &= \frac{2}{3} \\ w_0 &= -\frac{5}{3} \\ h(x) &= (2/3)x^2 - 5/3 \end{aligned}$$

7. What are the support vectors of the classifier? Confirm that the solution in part (6.) makes the constraints in (1) binding for support vectors.

Plugging in the points directly, we get that

$$\begin{aligned} h(-3) &= (2/3) \cdot 9 - (5/3) = 13/3 \\ h(-2) &= (8/3) - (5/3) = 1 \\ h(-1) &= (2/3) - (5/3) = -1 \\ h(0) &= (0) - (5/3) = -5/3 \\ h(1) &= (2/3) - (5/3) = -1 \\ h(2) &= (8/3) - (5/3) = 1 \\ h(3) &= (18/3) - (5/3) = 13/3 \end{aligned}$$

Our support vectors are (-2,1), (1,-1), (-1,1), and (2,1) in terms of (x,y). These points optimize our constraint that h(x) $\geq$ 1.

Problem 2 (Scaling up your SVM solver, 10pts (+opportunity for extra credit))

For this problem you will build a simple SVM classifier for a binary classification problem. We have provided you two files for experimentation: training *data.csv* and validation *val.csv*.

- 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. Implement this algorithm in *problem2.py*.

  [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.]

- Next experiment with the hyperparameters for each of these models. Try seeing if you can identify some patterns by changing β, N (the maximum number of support vectors), or the number of random training samples taken during the Randomized Search procedure (Section 4.3). Note the training time, training and validation accuracy, and number of support vectors for various setups.

- Lastly, compare the classification to the naive SVM imported from scikit-learn by reporting accuracy on the provided validation data. *For extra credit, implement the SMO algorithm and implement the LASVM process and do the same as above.*[a]

We are intentionally leaving this problem open-ended to allow for experimentation, and so we will be looking for your thought process and not a particular graph. Visualizations should be generated using the provided code. You can use the trivial $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$ kernel for this problem, though you are welcome to experiment with more interesting kernels too.

In addition, provide answers the following reading questions **in one or two sentences for each**.

1. In one short sentence, state the main purpose of the paper.
2. Describe each of the parameters in Eq. 1 in the paper
3. State, informally, one guarantee about 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. (*if you did the extra credit*) In simple words, what is the theoretical guarantee of LASVM algorithm? How does it compare to its practical performance?

[a] Extra credit only makes a difference to your grade at the end of the semester if you are on a grade boundary.

Solution 2

1. Next experiment with the hyperparameters for each of these models. Try seeing if you can identify some patterns by changing β, N (the maximum number of support vectors), or the number of random training samples taken during the Randomized Search procedure (Section 4.3). Note the training time, training and validation accuracy, and number of support vectors for various setups.

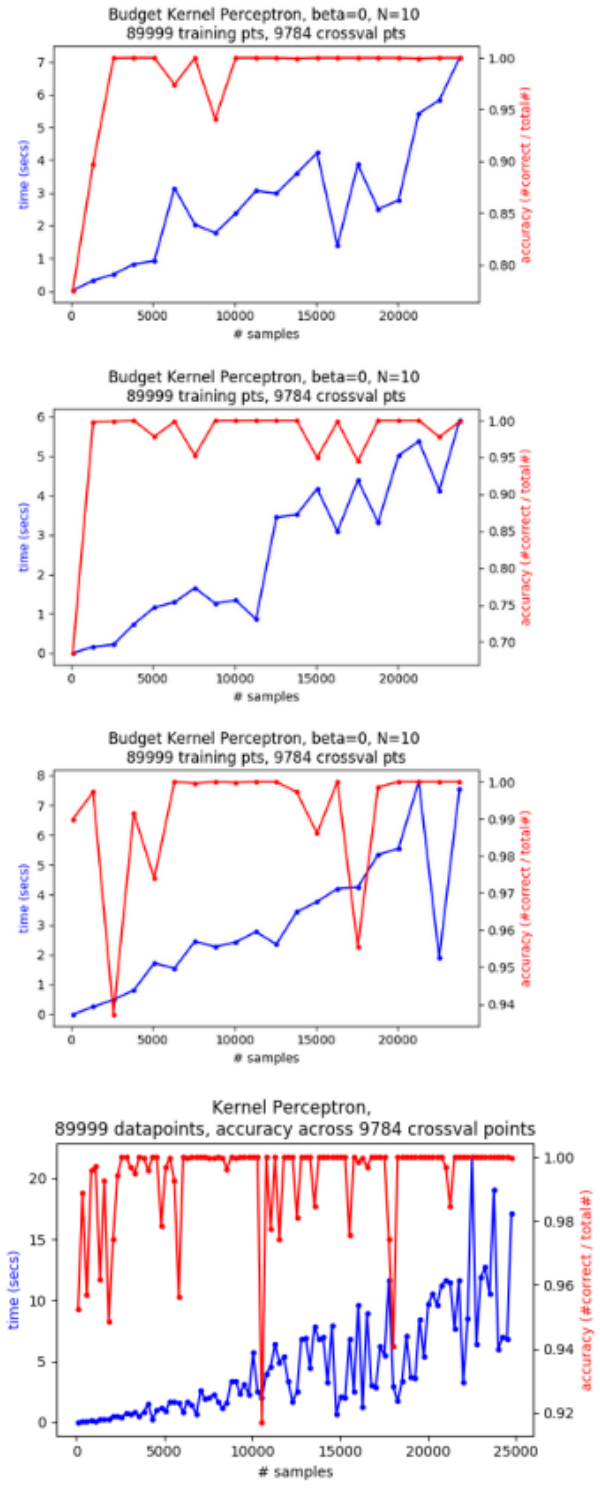

Figure 2: Samples vs Time. We can see that (a) the SVM is very sensitive to the initial randomly selected points (the three top graphs are all different) and (b) the budget perceptron does improve on the time required to train – at 15000 samples, it takes 3 seconds instead of 5 seconds. The data is very noisy, perhaps in part due to being run inside a virtual machine while other processes are running, but in general we see a trend toward higher accuracy (against the cross validation set) as number of samples increases. In fact it seems like we only need 2500 or so data points, not 20k, out of the original set of 90k, in order to potentially have high accuracy.

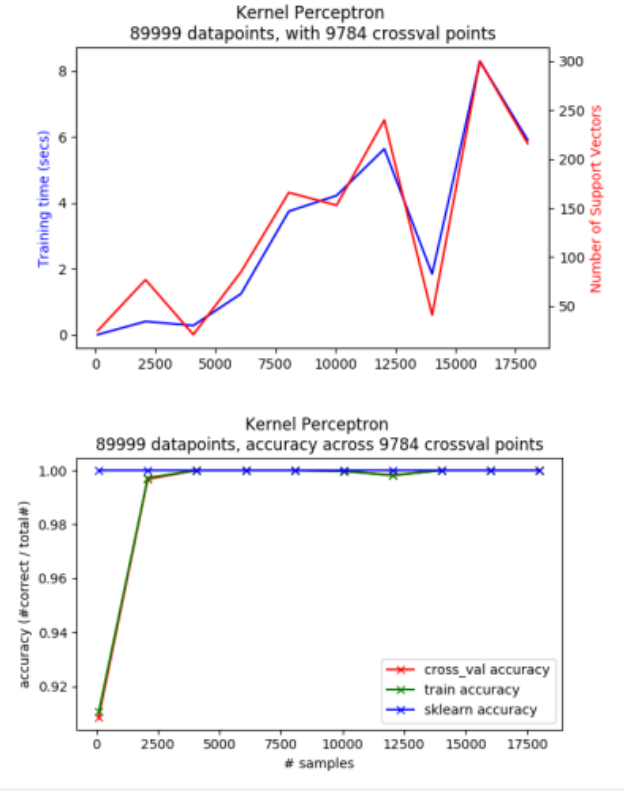

Figure 3: Kernel perceptron, run using a varying number of randomly drawn samples. We can see again that around 2500 samples we already have high accuracy. The number of support vectors and time to train both increase, as expected. The sklearn SVM (with linear kernel) again seems to have perfect accuracy, I suspect there is a bug in my code.

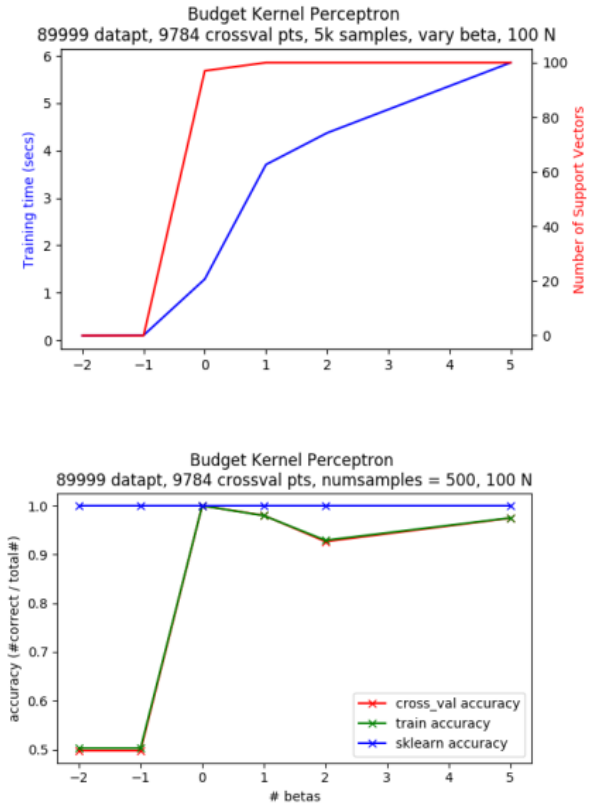

Figure 4: Budget version, vs changes in beta. We see that we want beta to be at least zero, or else all examples are misclassified. Zero also appears to be sufficient for accuracy purposes. Not that zero does mean that we have fewer than our max number of support vectors, vs when beta is 1 or higher.

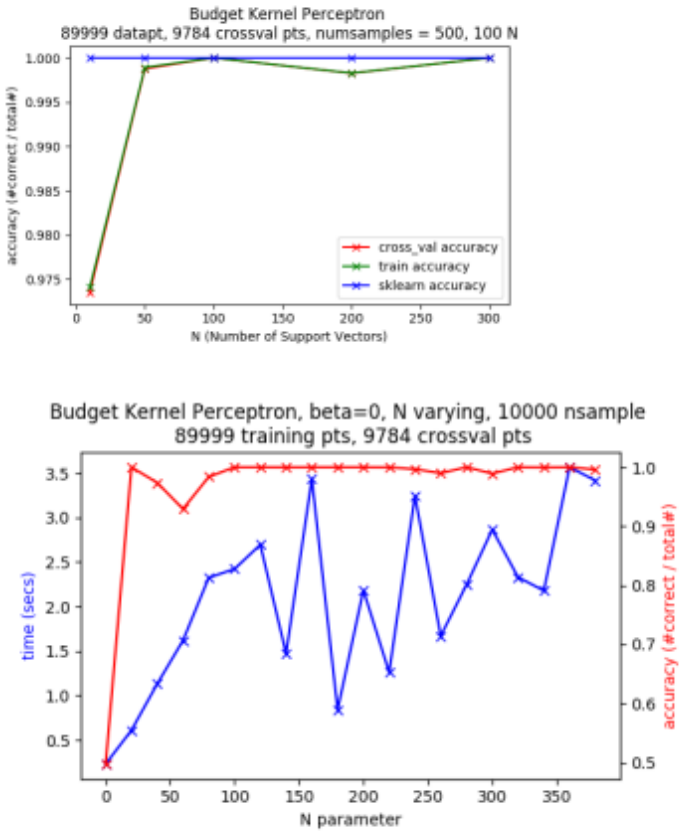

Figure 5: Budget version, vs changes in N (number of support vectors). At 50 or so support vectors we already have very good accuracy. As number of support vectors allowed goes up, the time to train goes up as well.

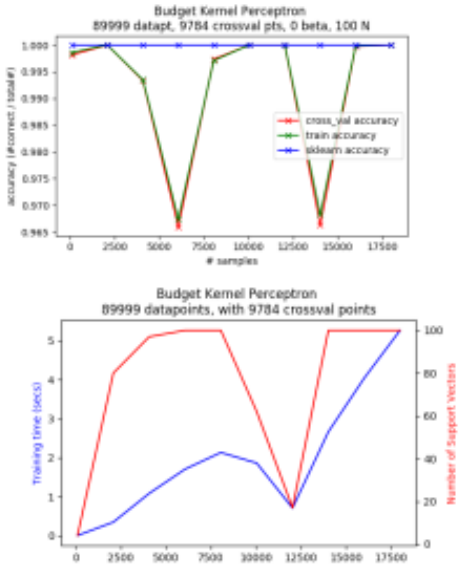

Figure 6: Budget version, vs number of randomly drawn training samples, There is a stray datapoint due to a copy paste error, but overall we can see that again at 2500 samples we have good accuracy, and 100 support vectors is sufficient. The training time remains significantly below that of the kernel.

In conclusion: beta = 0, N = 100, and number of samples = 5000 appear to be good (somewhat optimal) choices. The budget version performs just as well with less training time. We note that the SVM appears to perform very well on the cross validated data when fitted against the test data – this may be to a bug in the hand-written accuracy calculation scoring function.

2. Lastly, compare the classification to the naive SVM imported from scikit-learn by reporting accuracy on the provided validation data. *For extra credit, implement the SMO algorithm and implement the LASVM process and do the same as above.*[1]

3. In one short sentence, state the main purpose of the paper.

   This paper describes ways to improve the computational efficiency of SVM classifiers when dealing with large and potentially noisy datasets, including random sampling, removing support vectors and selectively picking training samples.

4. Describe each of the parameters in Eq. 1 in the paper

$$\hat{y}(x) = w'\mathbf{\Phi}(x) + b$$

   $\hat{y}$ is our estimate of the true class, either -1 or 1.

   $\mathbf{\Phi}(x)$ is our feature function, which transforms our data into a space where they are (hopefully) linearly separable. It is often hand-chosen, for instance in the problem set problem above, we were given $f(x) = (x, x^2)$.

   w and b are parameters we find by learning on a set of training examples for which we have the true class (or think we do).

[1] Extra credit only makes a difference to your grade at the end of the semester if you are on a grade boundary.

5. State, informally, one guarantee about the Kernel perceptron algorithm described in the paper.

   If a solution exists, the kernel perceptron will converge to it after a finite number of iterations.

6. What is the main way the budget kernel perceptron algorithm tries to improve on the perceptron algorithm?

   The budget versions maintains a limit on the number of support vectors, discarding the ones that are the furthest away from the margin as needed. This allows it to maintain sparsity (of support vectors, v.s total number of training examples) and to run on noisy data in a computationally efficient manner.

7. (*if you did the extra credit*) In simple words, what is the theoretical guarantee of LASVM algorithm? How does it compare to its practical performance?

   LASVM will exactly reach the SVM solution after a sufficient number of epochs. Additionally, the authors found that after just one epoch, the LASVM error rate closely approached LIBSVM accuracy.

Problem 3 (Ethics Assignment, 10pts)

Recall our class activity:

> Hiring at Abercrombie and Fitch. Abercrombie and Fitch have hired a new computer science team to design an algorithm to predict the success of various job applicants to sales positions at Abercrombie and Fitch. As you go through the data and design the algorithm, you notice that African-American sales representatives have significantly fewer average sales than white sales representatives. The algorithm's output recommends hiring far fewer African-Americans than white applicants, when the percentage of applications from people of various races are adjusted for.

In class, we thought about the problem *statically*: given historical data, such as data about sales performance, who should Abercrombie and Fitch hire right now?

In this follow-up assignment, I want you to think about consumer behavior and firm hiring practice dynamically. Looking at features of the labor market dynamically allows you more, or different, degrees of freedom in your model. For example, in class, you probably took consumer preference about the race of their sales representative as given. What would happen if you allowed consumer preference to vary (say, on the basis of changing racial demographics in the sales force)?

Heres the new case:

> The US Secretary of Labor has heard about your team's success with Abercrombie and Fitch and comes to you with a request. The Department of Labor wants to reduce disparate impact discrimination in hiring. They want you to come up with a model of fair hiring practices in the labor market that will reduce disparate impact while also producing good outcomes for companies.

Write two or three paragraphs that address the following:

- What are the relevant socially good outcomes, for both workers and companies?
- What are some properties of your algorithm that might produce those socially good results?
  - Think about constraints that you might build in, such as the fairness constraints that we discussed in class, or how you might specify the prediction task that we are asking the machine to optimize.
- Are there tradeoffs that your algorithm has to balance? [optional]
- Are there any features of data collection, algorithm implementation, or the social world that make you wary of using machine learning in this case? [optional]

We expect that:

- You focus on one or two points of discussion for each question.
  - For example, for question 2, pick a single fairness criterion. item Depth over breadth here!
- You provide reasons in support of your answers (i.e., explain why you chose your answer).
  - For example, for the first question, you might choose the socially good outcome of increased profit for companies, and give reasons why profit is the right social goal.
- You are clear and concise - stick to plain, unadorned language.
- You do not do any outside research.
- You demonstrate a thoughtful engagement with the questions.

Solution

Calibration [1pt]

Approximately how long did this homework take you to complete?

25-40 hrs (as usual depending on if you count reading time)

Name, Email, and Collaborators

Name: Nao Ouyang

Email: nouyang@g.harvard.edu

Collaborators: Eric Wilson Sharon Buse David