Homework 3: Support Vector Machines

Writeup due 23:59 on Friday 27 March 2015

You will do this assignment individually and submit your answers as a PDF via the Canvas course website. There is a mathematical component and a programming component to this homework. Do not submit code.

1. Fitting an SVM by hand

Consider a dataset with the following 6 points in $1D$:

$$\{(x_1, y_1)\} = \{(-3, +1), (-2, +1), (-1, -1), (1, -1), (2, +1), (3, +1)\}$$

Consider mapping these points to 2 dimensions using the feature vector $\phi : x \mapsto (x, x^2)$. The max-margin classifier objective is given by:

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

Note: the purpose of this exercise is to solve the SVM without the help of a computer, relying instead on principled rules and properties of these classifiers. 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 as an explicit function of x.

Solution

1. Geometrically, the maximum margin decision boundary is given by the line of equation $y = c$ where c is a constant. Any other line would yield a smaller margin. The optimal vector w is perpendicular to the decision boundary. Hence, a vector which is parallel to w is $(0, 1)$.

Check 1.1 You specified that the optimal decision boundary is given by a line parallel to the x-axis.

Check 1.2 You specified that the optimal vector is perpendicular to the decision boundary.

Check 1.3 You suggested any vector of the form $(0, c)$ where $c \in \mathbb{R}$

2. The margin is the distance from each support vector to the decision boundary. This can be read off the plot: $d = \frac{3}{2}$. You can also get full points if you took the margin to be the distance between the two separating hyperplanes (i.e twice the definition above). Note that this will require you to be consistent with the following question.

Check 1.4 You defined the margin or identified it on the plot.

Check 1.5 You gave either $\frac{3}{2}$ or 3 as your answer, consistently with the definition you offered.

3. Assuming the scaling on w and b where $\min_n w^T\phi(x_n) + b = 1$,we have that $d = \frac{1}{\|w\|}$. Note that if you took to the margin to be the distance between the two hyperplanes, you should have $d = \frac{2}{\|w\|}$. Together with the previous answer, we have that the optimal w is $(0, \frac{2}{3})$.

Check 1.6 You noted one of the two relations between margin and the norm of the optimal vector, consistently with your definition of margin.

Check 1.7 You answered that $w^* = (0, \frac{2}{3})$.

4. The support vectors will be on the decision boundary, so the inequalities will be tight. The points $(2, +1)$ is a support vector, such that: $4\frac{2}{3} + w_0 = 1 \implies w_0 = -\frac{5}{3}$

Check 1.8 You correctly substituted a support vector to the equation $w^T\phi(x) + w_0 = \pm 1$

Check 1.9 You gave $w_0 = -\frac{5}{3}$ as your answer.

5. The discriminant is $f(x) = \frac{2}{3}x^2 - \frac{5}{3}$.

Check 1.10 You gave the correct form for the discriminant.

2. Composing Kernel Functions [4pts]

Prove that

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-||\boldsymbol{x} - \boldsymbol{x}'||_2^2\},$$

where $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$ is a valid kernel, using only the following properties. If $K_1(\cdot,\cdot)$ and $K_2(\cdot,\cdot)$ are valid kernels, then the following are also valid kernels:

$$\begin{aligned}
K(\boldsymbol{x},\boldsymbol{x}') &= c\,K_1(\boldsymbol{x},\boldsymbol{x}') \quad \text{for } c > 0 \\
K(\boldsymbol{x},\boldsymbol{x}') &= K_1(\boldsymbol{x},\boldsymbol{x}') + K_2(\boldsymbol{x},\boldsymbol{x}') \\
K(\boldsymbol{x},\boldsymbol{x}') &= K_1(\boldsymbol{x},\boldsymbol{x}')\,K_2(\boldsymbol{x},\boldsymbol{x}') \\
K(\boldsymbol{x},\boldsymbol{x}') &= \exp\{K_1(\boldsymbol{x},\boldsymbol{x}')\} \\
K(\boldsymbol{x},\boldsymbol{x}') &= f(\boldsymbol{x})\,K_1(\boldsymbol{x},\boldsymbol{x}')\,f(\boldsymbol{x}') \quad \text{where } f \text{ is any function from } \mathbb{R}^D \text{ to } \mathbb{R}
\end{aligned}$$

Solution

- $K_0(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}'$ is a valid kernel by definition of the kernel (it is the inner product of $\boldsymbol{x}$ and $\boldsymbol{x}'$).
- Thus $K_1(\boldsymbol{x},\boldsymbol{x}') = \exp(2\boldsymbol{x}^T\boldsymbol{x}')$ is also a valid kernel by property 2 and 4.
- Note that $K(\boldsymbol{x},\boldsymbol{x}') = \exp(-\boldsymbol{x}^T\boldsymbol{x})\exp(2\boldsymbol{x}^T\boldsymbol{x}')\exp(-\boldsymbol{x}'^T\boldsymbol{x}') = f(\boldsymbol{x})K_1 f(\boldsymbol{x}')$, where $f(\boldsymbol{x}) = \exp(-\boldsymbol{x}^T\boldsymbol{x})$.
- Hence using property 5 in the last step, we proved $K(\boldsymbol{x},\boldsymbol{x}')$ is a kernel.

Note: a common mistake is saying $\exp(-\boldsymbol{x}^T\boldsymbol{x})$ is a kernel. It is not.

> **Check 2.1** You specified a series of steps using the various properties to show that it is a valid kernel.

3. Scaling up your SVM Solver

In the previous homework, you studied a simple data set of fruit measurements. We would like you to code up a few simple SVM solvers to classify lemons from apples. To do this, 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. 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. 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 $K(x_1, x_2) = x_1^T x_2$ kernel for this problem, though you can welcome to experiment with more interesting kernels too.

Lastly, compare the classification to the naive SVM imported from scikit-learn. For extra credit (+7 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?

Reading questions

1. The goal of the paper is to achieve approximations of the SVM QP solution at a lower computational cost to handle larger datasets.

   Check 3.1 Your answer closely matches the one above.

2. The parameters are:

   - $\hat{Y}$: value to be thresholded to get prediction.
   - $\phi(X)$: feature vector for the data.
   - W: weight vector. Is a parameter of the model.
   - B: bias term. Is a parameter of the model.

   Check 3.2 Your answer identifies each of the terms above.

3. One guarantee of the Kernel perceptron algorithm is that it converges after a finite number of misclassifications/insertions of support vectors (cf. Novikoff's Theorem) if a solution exists (the mapping is linearly separable).

   Check 3.3 Your answer closely matches the one above.

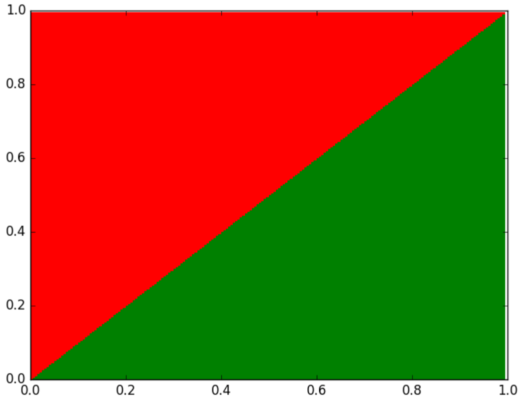
4. The main way the budget kernel perceptron improves on the perceptron algorithm is to allow for the removal of support vectors in order to reduce overfitting while increasing the margin.

   Check 3.4 Your answer closely matches the one above.

5. The theoretical guarantee of the LaSVM algorithm is that it converges to the SVM QP solution after sufficiently many passes over the data. In practice, the LaSVM algorithm outputs a good approximation after 1 pass over the data.

   Check 3.5 Your answer closely matches the one above.

Programming questions Your image for the correct implementation of KernelPerceptron and BudgetKernelPerceptron should look like



Check 3.6 Your image looks like the above..

Next, you were asked to some experimentation to see how changing the parameters affects the outcome. You should have done at least two of the following:

- adjust β
- adjust N
- adjust number of samples
- changed the Kernel

Your analysis should focus on (at least) changes to accuracy and speed. $\beta = 0$ would produce a valid decision boundary. Positive values of β would lead to adjustments if $y * \hat{y}$ for a point was negative (meaning the point was misclassified) or within β of 0 (meaning not correctly classified by enough). Similarly, negative β means that you are okay with occasionally being wrong about a classification, as long as $\hat{y}$ is within β of 0. These should, respectively, have slowed down and sped up your algorithm, relative to $\beta = 0$.

Changing N changes the maximum number of support vectors. You will likely find that very low values of N like 10 are quite fast per epoch (and overall), but not as highly performant. You likely will not get a correct decision boundary with this. With larger N, you get less speed performance, since you have more support vectors, but better accuracy.

Increasing the number of samples likely slowed down your algorithm since it has to iterate through more samples. However, having too few will decrease accuracy since you may not have enough training data (you may not have enough support vectors).

Changing the Kernel could have varied behavior, and you are expected to explain what happened and why you think it did.

Check 3.7 You adjusted at least one of the above parameters and noted what happened, which should be close to what is described above.

Check 3.8 You adjusted at least two of the above parameters and noted what happened, which should be close to what is described above. Note that if you got Check 3.7, you should also state that you got Check 3.8.