CS 536 : Perceptrons and SVMs

16:198:536

Perceptrons

As usual, we want to generate a data set to fit a perceptron onto. Recall that in the analysis of perceptrons in the notes, we assumed that all the data was contained within the unit sphere of whatever the underlying dimension was. Consider generating data points *on* the unit sphere in the following way: let $\underline{Z}$ be a k-dimensional vector, where each entry is an i.i.d. standard normal, mean 0 and variance 1. Then define $\underline{X} = \underline{Z}/||\underline{Z}||$ (taking the norm as the 2-norm or Euclidean norm), so that $\underline{X}$ is a random vector of length 1, lying exactly on the k-dimensional unit sphere. Note that because of the spherical symmetry of the $\underline{Z}$, the $\underline{X}$ will be uniformly distributed over the surface of the sphere.

For a given value of $1 > \epsilon > 0$, discard any points where $|X_k| < \epsilon$ - this will create two hemispheres of data points separated by a gap along the equator (in this high dimensional space). Let us classify any remaining point where $X_k \geq \epsilon$ with $Y = +1$, and any remaining point with $X_k \leq -\epsilon$ with $Y = -1$. Note that for a data set defined in this way, we have a very natural perceptron that can be applied:

$$\text{classify}(\underline{x}) = \begin{cases} +1 & \text{if } 0x_1 + 0x_2 + \ldots + 0x_{k-1} + 1x_k > 0 \\ -1 & \text{if } 0x_1 + 0x_2 + \ldots + 0x_{k-1} + 1x_k < 0, \end{cases} \tag{1}$$

with a linear separator given by $x_k = 0$. This is by no means the only feasible perceptron, but it does show that the data is linearly separable.

Additionally, note that the margin of separation between the positive and negative data classes is *at least* ϵ.

We proved in class that the convergence of the Perceptron Learning Algorithm is bound independently of the dimension of the data and the number of data points, and can be bound entirely in terms of the maximum margin between the two data classes. I'd like you to try to verify this experimentally. For a given value of k and ϵ, and a desired value of m, consider repeatedly generating points $\underline{x}$ but discarding them if they do not satisfy $|x_k| \geq \epsilon$ - repeat this process until m data points have been generated.

1) For $k = 5, \epsilon = 0.1$, for a range of possible m values, repeatedly generate data sets of size m and fit a perceptron to them. Plot, as a function of m, the average number of steps needed for the Perceptron Learning Algorithm to converge. Do your results make sense? Do you think looking at larger and larger m values outside the range you plot will produce anything different?

2) For $m = 100, \epsilon = 0.05$, for a range of possible k values, repeatedly generate data sets of dimension k, and fit a perceptron to them. Plot, as a function of k, the average number of steps needed for the Perceptron Learning Algorithm to converge. Do your results make sense? Do you think looking at larger and larger k values outside the range you plot will produce anything different?

3) For $k = 5, m = 100$, for a range of possible ϵ values in $[0, 1]$, repeatedly generate data sets with an ϵ-threshold cutoff and fit a perceptron to them. Plot, as a function of ϵ, the average number of steps needed for the Perceptron Learning Algorithm to converge. Do your results make sense? Do you think looking at different k, m values will produce anything different?

Bonus: Let $\underline{w}^*, b^*$ be the weight vector and bias specifying the perceptron I initially gave for this data set. We might be interested in how close the 'typical' perceptron generated is to this 'ideal' perceptron. For each of the three problems above, repeat the calculations for the specified parameters, but average the weights and biases of the perceptrons generated to give an 'average' perceptron weight vector and bias $\underline{w}, b$. Then plot, as a function of the indicated parameter, the value of $||\underline{w}^* - \underline{w}||^2 + (b^* - b)^2$. Analyze your results.

SVMs

1) Suppose you had a data set in two dimensions that satisfied the following: the positive class all lay within a certain radius of a point, the negative class all lay outside that radius.

   - Show that under the feature map $\phi(x_1, x_2) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ (or equivalently, with the kernel $K(\underline{x}, \underline{y}) = (1 + \underline{x}.\underline{y})^2$), a linear separator can always be found in this embedded space, *regardless of radius and where the data is centered.*
   - In fact show that if there is an ellipsoidal separator, regardless of center, width, orientation (and dimension!), a separator can be found in the quadratic feature space using this kernel.

2) As an extension of the previous problem, suppose that the two dimensional data set satisfied the following: the positive class lay within one of two (disjoint) ellipsoidal regions, and the negative class was everywhere else. Argue that the kernel $K(\underline{x}, \underline{y}) = (1 + \underline{x}.\underline{y})^4$ will recover a separator.

3) Suppose that the two dimensional data set is distributed like the following: the positive class lays in a circle centered at some point, the negative class lies in a circular band surrounding it of some radius, and then additional positive points lie outside that radius. Argue that the kernel $K(\underline{x}, \underline{y}) = (1 + \underline{x}.\underline{y})^4$ will recover a separator.

4) Consider the XOR data (located at $(\pm 1, \pm 1)$). Express the dual SVM problem and find a separator using

   - $K(\underline{x}, \underline{y}) = (1 + \underline{x}.\underline{y})^2$
   - $K(\underline{x}, \underline{y}) = \exp(-||\underline{x} - \underline{y}||^2)$.

   For each, determine the regions of (x_1, x_2) space where points will be classified as positive or negative. Given that each produces a distinct separator, how might you decide which of the two was preferred?