CS 189 Spring 2016: Introduction to Machine Learning - Discussion 8

1. Perceptron and SVM Kernel

   We are given a set of n training points $\{\mathbf{x}^{(i)}\}$ and associated training labels $\{y^{(i)}\}$. On this data, we train a perceptron with a Gaussian kernel,

$$K(\mathbf{u}, \mathbf{v}) = e^{\frac{-(\mathbf{u}-\mathbf{v})^2}{2\sigma^2}}$$

   using the dual perceptron algorithm. This gives us a set of dual weights $\{\alpha_i\}$.

   a) What is the interpretation of the dual weights $\{\alpha_i\}$?

   Solution: α_i reflects how many times we have added $\epsilon * y_i * x_i$ to w during the training phase.

   b) How do we classify a new test point $\mathbf{x}$?

   Solution: Let Φ be the (infinite-dimensional) feature mapping for a Gaussian kernel, $\Phi(\mathbf{u})^\top \Phi(\mathbf{v}) = K(\mathbf{u}, \mathbf{v})$. Recall $\mathbf{w} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}^{(i)})$.

$$\begin{aligned}
\hat{y} &= \text{sign}(\mathbf{w}^T \Phi(\mathbf{x})) \\
&= \text{sign}\Big(\sum_{i=1}^n \alpha_i \Phi(\mathbf{x}^{(i)})^\top \Phi(\mathbf{x})\Big) \\
&= \text{sign}\Big(\sum_{i=1}^n \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x})\Big) \\
&= \text{sign}\Big(\sum_{i=1}^n \alpha_i e^{\frac{-(\mathbf{x}^{(i)}-\mathbf{x})^2}{2\sigma^2}}\Big)
\end{aligned}$$

   c) Suppose we are training an SVM instead of a perceptron. What changes about the answer to b)?

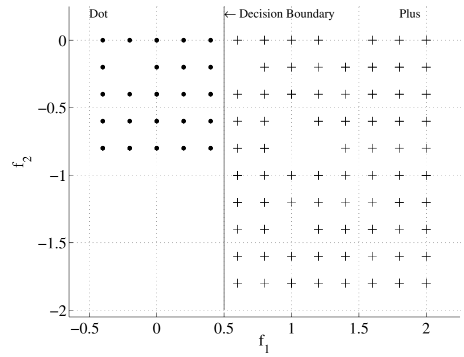
   Solution: For an SVM, the weight vector w is a linear combination of only the support vectors. Therefore, if we instead consider $\{\mathbf{x}^{(i)}\}$ to be the set of support vectors with associated training labels $\{y^{(i)}\}$, our previous answer still holds.

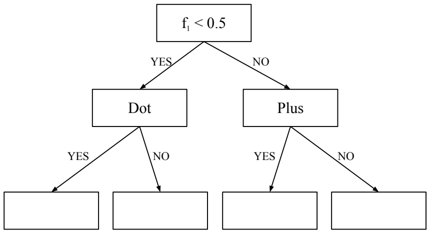
2. Decision Trees

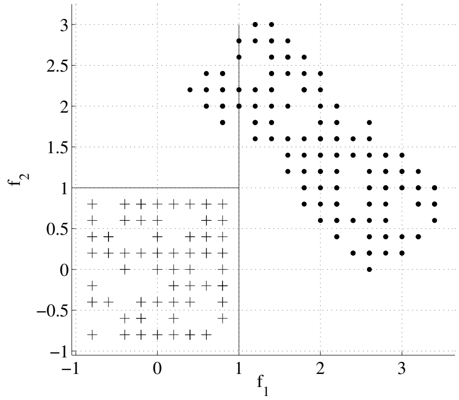
   You are given points from 2 classes, shown as +'s and ·'s. For each of the following sets of points,

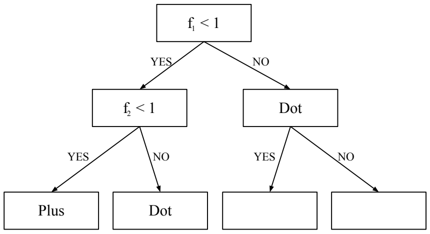
   1. Draw the decision tree of depth at most 2 that can separate the given data completely, by filling in binary predicates (which only involve thresholding of a *single* variable) in the boxes for the decision trees below. If the data is already separated when you hit a box, simply write the class, and leave the sub-tree hanging from that box empty.
   2. Draw the corresponding decision boundaries on the scatter plot, and write the class labels for each of the resulting bins somewhere inside the resulting bins.

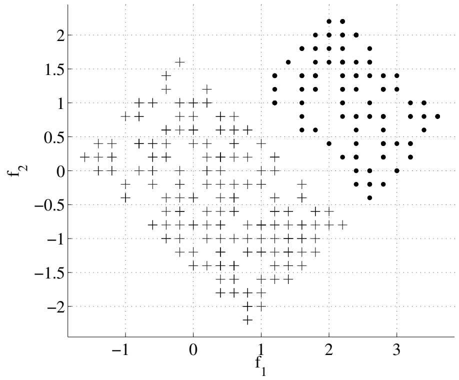
   If the data cannot be separated completely by a depth 2 decision tree, simply cross out the tree template. We solve the first part as an example.

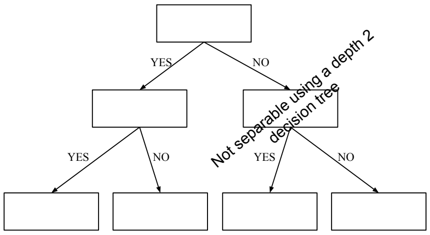

Dot
← Decision Boundary
Plus
0
−0.5
−1
−1.5
−2
f_2
−0.5
0
0.5
1
1.5
2
f_1

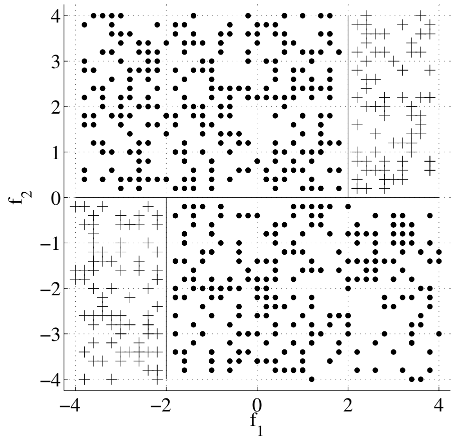

f_1 < 0.5
YES
NO
Dot
Plus
YES
NO
YES
NO

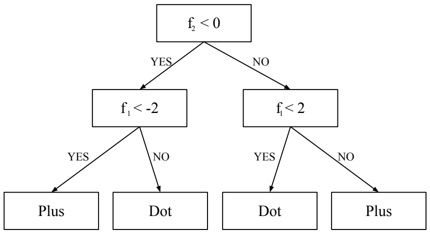

3
2.5
2
1.5
1
0.5
0
−0.5
−1
f_2
−1
0
1
2
3
f_1


f_1 < 1
YES
NO
f_2 < 1
Dot
YES
NO
YES
NO
Plus
Dot


2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
f_2
−1
0
1
2
3
f_1


YES
NO
Not separable using a depth 2 decision tree
YES
NO
YES
NO


4
3
2
1
0
−1
−2
−3
−4
f_2
−4
−2
0
2
4
f_1


f_2 < 0
YES
NO
f_1 < -2
f_1 < 2
YES
NO
YES
NO
Plus
Dot
Dot
Plus

3. Curse of Dimensionality

   We have a training set: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})$, with $\mathbf{x}^{(i)} \in \mathbb{R}^d$. Our 1-nearest neighbor classifier for a test point $\mathbf{x}$ is:

$$\text{class}(\mathbf{x}) = y^{(i^*)} \quad \text{where } \mathbf{x}^{(i^*)} \text{ is the nearest neighbor of } \mathbf{x}.$$

   Assume any data point $\mathbf{x}$ is inside the Euclidean ball of radius 1, i.e. $\|\mathbf{x}\|_2 \leq 1$. To be confident in our prediction, we want the distance between $\mathbf{x}$ and its nearest neighbor to be small, within some positive ϵ:

$$\|\mathbf{x} - \mathbf{x}^{(i^*)}\|_2 \leq \epsilon \quad \text{for all } \|\mathbf{x}\|_2 \leq 1. \tag{1}$$

   For this condition to hold, at least how many data points should be in the training set? How does this lower bound depend on the dimension d?

   Hint: Think about the volumes of the hyperspheres, and use the union bound:

$$\text{vol}(\bigcup_{j=1}^{k} S_j) \leq \sum_{j=1}^{k} \text{vol}(S_j), \text{ where } S_j \text{ is a hypersphere.}$$

Solution: Let B_0 be the ball centered at the origin, having radius 1 (inside which we assume our data lies). Let $B_i(\epsilon)$ be the ball centered at $\mathbf{x}^{(i)}$, having radius ϵ. For inequality (1) to hold, for any point $\mathbf{x} \in B_0$, there must be at least one index i such that $\mathbf{x} \in B_i(\epsilon)$. This is equivalent to saying that the union of $B_1(\epsilon), \ldots, B_n(\epsilon)$ covers the ball B_0. Let $\text{vol}(\cdot)$ indicate the volume of an object. Then

$$\text{vol}(B_0) \leq \text{vol}(\bigcup_{i=1}^{n} B_i(\epsilon)) \leq \sum_{i=1}^{n} \text{vol}(B_i(\epsilon)) = n\text{vol}(B_1(\epsilon)).$$

This implies

$$n \geq \frac{\text{vol}(B_0)}{\text{vol}(B_1(\epsilon))} = \frac{c(1^d)}{c\epsilon^d} = \frac{1}{\epsilon^d}$$

where the constant c is dependent on the formula for the volume of a hypersphere in d dimensions. This lower bound suggests that to make an accurate prediction on high-dimensional input, we need exponentially many samples in the training set. This exponential dependence is sometimes called the *curse of dimensionality.* It highlights the difficulty of using non-parametric methods for solving high-dimensional problems.