01/25/2016

Classifiers

- You are given a set of n samples, each with d features.
- Some samples belong to a certain class $\mathcal{O}$; some do not.
- Example: sample are bank loans, features are income and age (d=2). Some are in class defaulted; some are not. Goal: Predict whether future borrowers will default based on their income and age.
- Represent each sample as a point in a d-dimensional space, called a feature vector (aka predictors, independent variables).
- Decision boundary: the boundary chosen by our classifier to separate $\mathcal{O}$ from not $\mathcal{O}$.
- Some (not all) classifiers work by computing a predictor function: A function $f(x)$ that maps sample point x to a scalar such that,

$$f(x) > 0 \; if \; x \; \in class \; \mathcal{O}$$
$$f(x) \leq 0 \; if \; x \; \notin \; class \; \mathcal{O}$$

  (aka decision function, or discriminant function).
- For these classifiers, the decision boundary is,

$$\{x \in \mathbb{R}^d : f(x) = 0\}$$

  That is the set of all points where the prediction function is zero. Usually this set is a $(d-1)$-dimensional surface in $\mathbb{R}^d$.
- $\{x : f(x) = 0\}$ is also called an isosurface (aka isocontours) of f for the isovalue 0.
- Linear classifier: The decision boundary is a hyperplane. Usually uses a linear predictor function.
- Overfitting: when sinuous (having many curves and turns) decision boundary fits sample data so well that it doesn't classify future (test set) items well.

Math Review

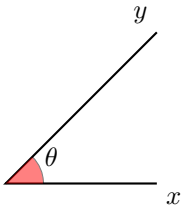
- Vectors:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T$$

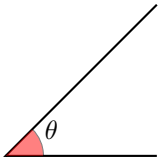
  Think of x as a point in $\mathbb{R}^d$.
- Conventions (often, but not always):
  - Uppercase roman = matrix.
  - Lowercase roman = vector.
  - Greek = scalar.
  - Other scalars: n = number of samples, d = number of features or dimension of sample, i, j, and k = indices.

- Functions (often scalars): $f()$, $s()$, etc.

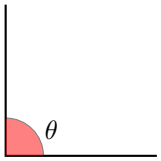
- Inner products (aka dot products)
  - $x \cdot y = x_1 y_1 + x_2 y_2 + \cdots + x_d y_d$
  - Also written $x^T y$.
  - Clearly, $f(x) = w \cdot x + \alpha$ is a linear function in x.
- Eucledian norms: $||x|| = \sqrt{x \cdot x} = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2}$
  - $||x||$ is the length (aka Eucledian length) of a vector x.
  - Given a vector x, $\frac{x}{||x||}$ is a unit vector (length 1).
  - "Normalize" a vector x: replace x with $\frac{x}{||x||}$.
- Use dot product to compute angles:



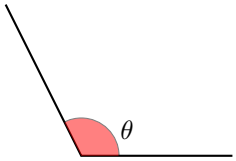
$$cos\ \theta = \frac{x \cdot y}{||x|| \cdot ||y||} = \frac{x}{||x||} \cdot \frac{y}{||y||}$$



$$acute,\ cos\ \theta > 0$$



$$right,\ cos\ \theta = 0$$



$$obtuse,\ cos\ \theta < 0$$

- Given a linear predictor function $f(x) = w \cdot x + \alpha$, decision boundary is

$$H = \{x : w \cdot x = -\alpha\}$$

  - The set H is called a hyperplane (A line in 2D, a plane in 3D).
- Theorem: Let $\vec{xy}$ be a vector that lies in H. Then $w \cdot (y - x) = 0$.
  Proof: x and y lie on the hyperplane H. $\therefore w \cdot (y - x) = -\alpha - (-\alpha) = 0$.
- w is called the normal vector of H. w is normal (perpendicular) or orthogonal to H.
- If w is a unit vector, $w \cdot x + \alpha$ is called the signed distance from x to H i.e. its distance, but positive on one side of H; negative on the other. Moreover the distance from H to the origin is α. Hence $\alpha = 0$ if and only if H contains the origin.

- The coefficients in w, plus α are called weights or regression coefficients. Goal of many Machine Learning algorithms is to find what the weights should be.
- The input data is linearly separable if $\exists$ a hyperplane that separates all samples $\in \mathcal{O}$ from all samples $\notin \mathcal{O}$.

Perceptron algorithm

- (Frank Rosenblatt, 1957) Slow, but correct for linearly separable samples. Uses a numerical optimization algorithm: gradient descent.
- Consider n sample vectors $x_1, x_2, \ldots x_n$.
- For each sample, let

$$y_i = \begin{cases} 1 & \text{if } x_i \in \mathcal{O} \\ -1 & \text{if } x_i \notin \mathcal{O} \end{cases}$$

- Goal: find weights w such that

$$\begin{aligned} x_i \cdot w &\geq 0 \quad \text{if } y_i = 1 \\ x_i \cdot w &\leq 0 \quad \text{if } y_i = -1 \end{aligned}$$

- Equivalently: $y_i x_i \cdot w \geq 0$. Inequality is called a constraint.
- Idea: We define a risk function R that is positive if some constraint is violated. Then we use optimization to choose w that minimizes R.
- Define the loss function

$$L(y, y_i) = \begin{cases} 0 & \text{if } y_i y \geq 0 \\ -y_i y & \text{otherwise} \end{cases}$$

- Define the risk function (aka objective function or cost function)

$$R(w) = \sum_{i=1}^{n} L(x_i \cdot w, y_i) = \sum_{i \in V} -y_i \cdot x_i \cdot w$$

  where $x_i \cdot w$ is our prediction, y_i is the correct classification and v is the set of indices i for which $y_i x_i \cdot w < 0$.
- If w classifies $X_1, \ldots, X_n$ correctly, then $R(w) = 0$. Otherwise, $R(w) > 0$; we want to find a better w.
- **Goal: Solve this optimization problem; Find w that minimizes $R(w)$.**