k-Nearest-Neighbor Algorithm

- Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it.
- The training tuples are described by *n* attributes. Each tuple represents a point in an *n*-dimensional space. In this way, all of the training tuples are stored in an *n*-dimensional pattern space. When given an unknown tuple, a ***k***-nearest-neighbor classifier searches the pattern space for the *k* training tuples that are closest to the unknown tuple. These *k* training tuples are the *k* nearest neighbors of the unknown tuple.
- Closeness is defined in terms of a distance metric, such as Euclidean distance.
- The Euclidean distance between two points or tuples, say, ***X***1 = (*x*11, *x*12, … , *x*1*n*) and ***X***2 = (*x*21, *x*22, … ,*x*2*n*), is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple ***X*1**and in tuple ***X*2**, square this difference, and accumulate it. The square root is taken of the total accumulated distance count. Min-Max normalization can be used to transforma value *v* of a numeric attribute *A* to *v*0 in the range [0, 1] by computing

$$v' = \frac{v - min_A}{max_A - min_A},$$

Where *minA* and *maxA* are the minimum and maximum values of attribute *A*

For *k*-nearest-neighbor classification, the unknown tuple is assigned the mostcommon class among its *k* nearest neighbors.

When *k* = 1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.

Nearest neighbor classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple.

In this case, the classifier returns the average value of the real-valued labels associated with the *k* nearest neighbors of the unknown tuple.

SVM—SUPPORT VECTOR MACHINES

- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyper plane (i.e., “decision boundary”)
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyper plane
- SVM finds this hyper plane using support vectors (“essential” training tuples) and margins (defined by the support vectors)
- Features: training can be slow but accuracy is high owing to their ability to model complex nonlinear decision boundaries (margin maximization)
- Used both for classification and prediction

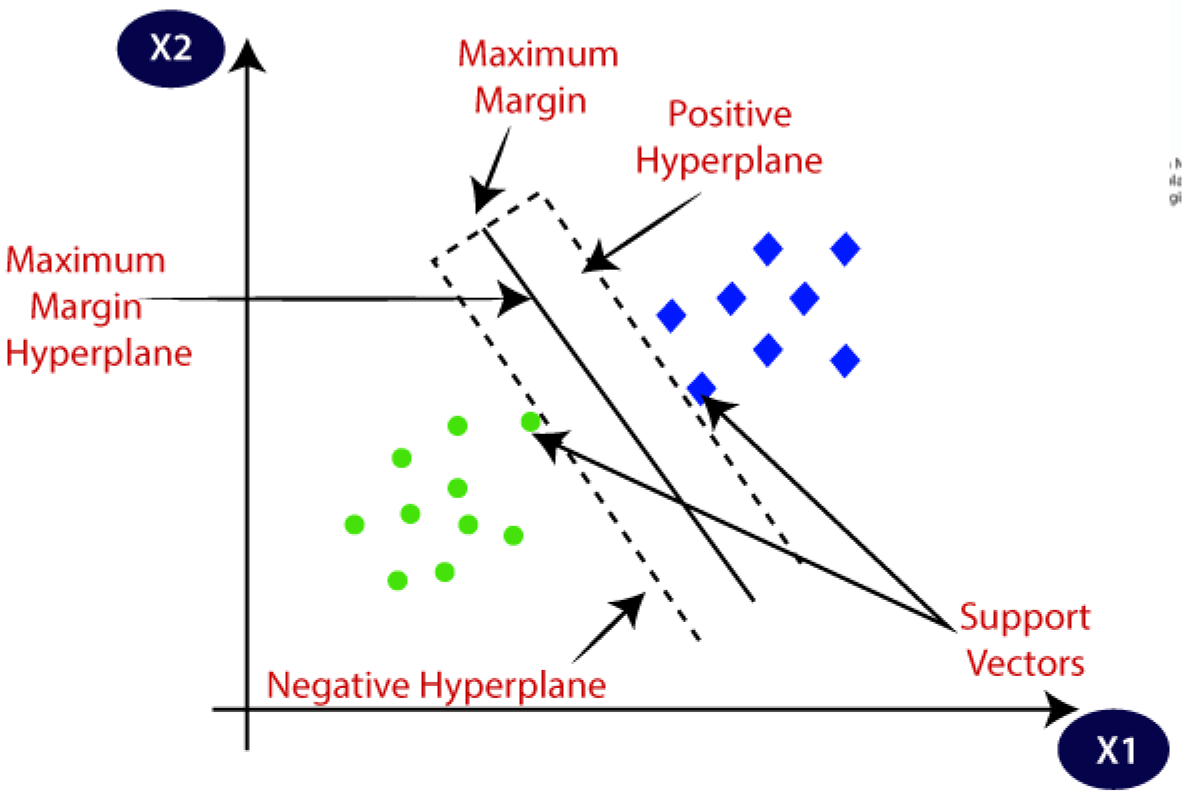
Applications:

Handwritten digit recognition,

Object recognition

Speaker identification,

Benchmarking time-series prediction tests



d

A separating hyper plane can be written as

$$W \bullet X + b = 0$$

Where W={w1, w2, …, wn} is a weight vector and b a scalar (bias)

For 2-D it can be written as

$w_0 + w_1 x_1 + w_2 x_2 = 0$

The hyper plane defining the sides of the margin:

H1: $w_0 + w_1 x_1 + w_2 x_2 \geq 1$ for $y_i = +1$, and

H2: $w_0 + w_1 x_1 + w_2 x_2 \leq -1$ for $y_i = -1$

Any training tuples that fall on hyper planes H1 or H2 (i.e., the sides defining the margin) are support vectors

□ This becomes a constrained (convex) quadratic optimization problem: Quadratic objective function and linear constraints □ Quadratic Programming (QP) □ Lagrangian multipliers

LINEAR REGRESSION

Linear regression: involves a response variable y and a single predictor variable x

y = w0 + w1 x

Where w0 (y-intercept) and w1 (slope) are regression coefficients

 Method of least squares: estimates the best-fitting straight line

 Multiple linear regression: involves more than one predictor variable

 Training data is of the form (X1, y1), (X2, y2),…, (X|D|, y|D|)

 Ex. For 2-D data, we may have: y = w0 + w1 x1+ w2 x2

 Solvable by extension of least square method or using SAS, S-Plus

 Many nonlinear functions can be transformed into the above

Nonlinear Regression

Some nonlinear models can be modeled by a polynomial function

A polynomial regression model can be transformed into linear regression model. For example,

y = w0 + w1 x + w2 x2 + w3 x3

Convertible to linear with new variables: x2 = x2, x3= x3

y = w0 + w1 x + w2 x2 + w3 x3

 Other functions, such as power function, can also be transformed to linear model

 Some models are intractable nonlinear (e.g., sum of exponential terms)

Possible to obtain least square estimates through extensive calculation on more complex formulae