Support Vector Machine

Tuan Nguyen

November 22, 2022

Maximum Margin Classifiers

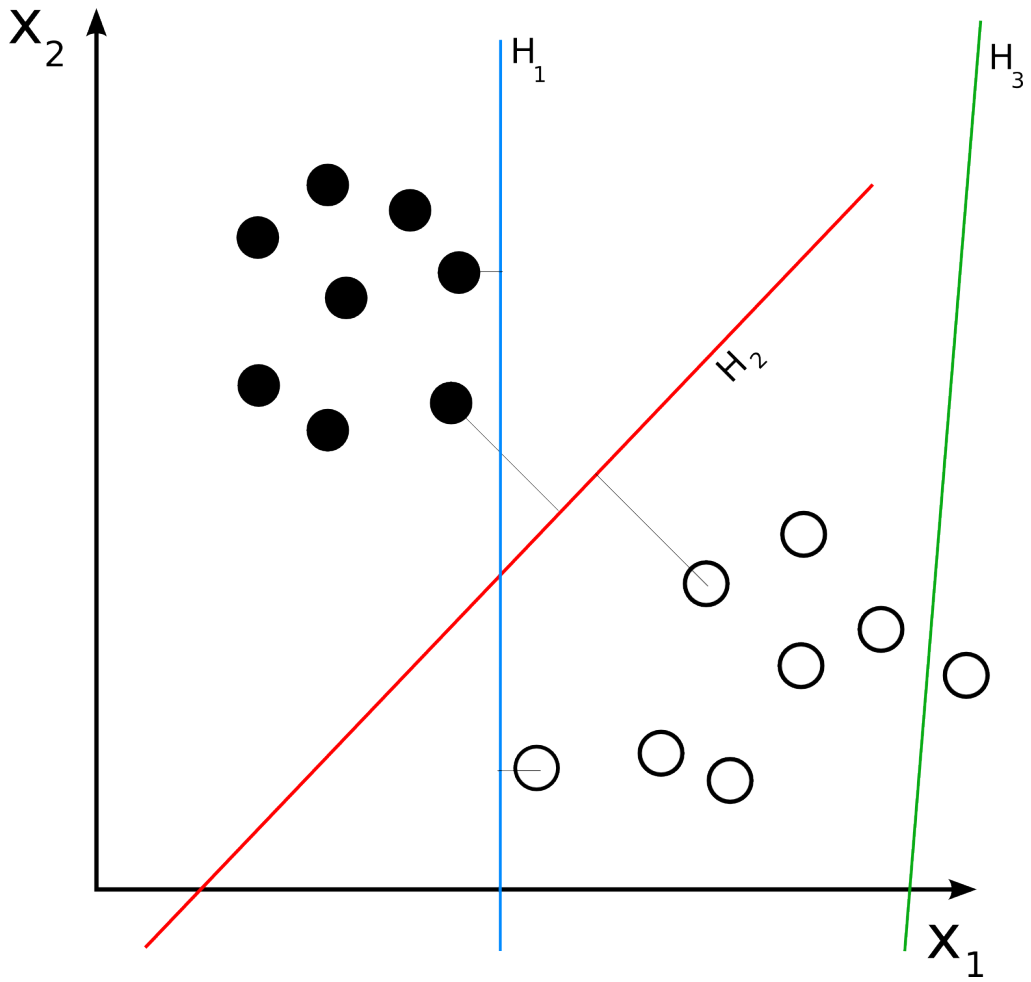

Figure 1: Which Separating Hyperplane?

Maximum Margin Classifiers (cont.)

The two-class classification problem using linear models of the form

$$y(x) = w^T \phi(x) + b$$

where $\phi(x)$ denotes a fixed feature-space transformation.

The training data set comprises N input vectors $x_1, ..., x_N$, with corresponding target values $t_1, ..., t_N$ where $t_n \in \{-1, 1\}$, and new data points x are classified according to the sign of y(x).

We shall assume that the training data set is linearly separable in feature space $\Rightarrow$ there exists at least one choice of the parameters w and b satisfies $y(x_n) > 0$ for points having $t_n = +1$ and $y(x_n) < 0$ for points having $t_n = -1$, so that $t_n y(x_n) > 0$ for all training data points.

Maximum Margin Classifiers (cont.)

The support vector machine approaches this problem through the concept of the margin, which is defined to be the smallest distance between the decision boundary and any of the samples.

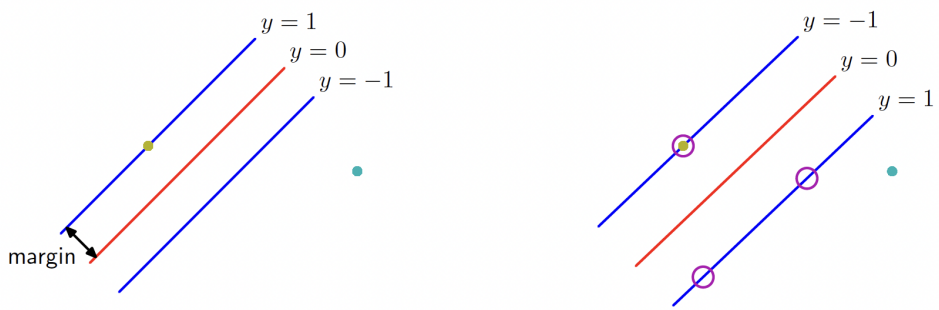

Figure 2: The margin

In support vector machines the decision boundary is chosen to be the one for which the margin is maximized.

Maximum Margin Classifiers (cont.)

The distance of a point x_n to the decision surface defined by y(x) = 0 is given by

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T \phi(x_n) + b}{\|w\|}$$

The margin is given by the perpendicular distance to the closest point x_n from the data set, and we wish to optimize the parameters w and b in order to maximize this distance. Thus the maximum margin solution is found by solving

$$\arg\max_{w,b} \frac{1}{\|w\|} \min_n [t_n(w^T \phi(x_n) + b]$$

Maximum Margin Classifiers (cont.)

Direct solution of this optimization problem would be very complex, and so we shall convert it into an equivalent problem that is much easier to solve.

We note that if we make the rescaling $w \rightarrow \alpha w$ and $b \rightarrow \alpha y$, then the distance from any point x_n to the decision surface, given by $\frac{t_n y(x_n)}{\|w\|}$ is unchanged. We can use this freedom to set:

$$t_n(w^T \phi(x_n) + b) = 1$$

for the point that is closest to the surface. In this case, all data points will satisfy the constraints

$$t_n(w^T \phi(x_n) + b) >= 1, n = 1, ..., N$$

so we have to solve the optimization problem with the above constrains

$$\arg\max_{w,b} \frac{1}{2}\|w\|^2$$

Maximum Margin Classifiers (cont.)

In order to solve this constrained optimization problem, we introduce Lagrange multipliers $a_n \geq 0$, with one multiplier an for each of the constraints, giving the Lagrangian function

$$L(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n(t_n(w^T\phi(x_n) + b) - 1)$$

Setting the derivatives of L(w, b, a) with respect to w and b equal to zero, we obtain the following two conditions

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$0 = \sum_{n=1}^{N} a_n t_n$$

Maximum Margin Classifiers (cont.)

Eliminating w and b from L(w, b, a) using these conditions then gives the dual representation of the maximum margin problem in which we maximize

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

with respect to a subject to the constraints

$$a_n \geq 0$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Here the kernel function is defined by $k(x, x') = \phi(x)^T \phi(x')$

Maximum Margin Classifiers (cont.)

Knowing the a_i we can find the weights w for the maximal margin separating hyperplane.

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b.$$

Kuhn-Tucker theorem: the solution we find here will be the same as the solution to the original problem. Also KKT conditions require that the following three properties hold:

$$a_n \geq 0$$

$$t_n y(x_n) - 1 \geq 0$$

$$a_n(t_n y(x_n) - 1) = 0$$

Maximum Margin Classifiers (cont.)

Thus for every data point, either $a_n = 0$ or $t_n y(x_n) = 1$.

- Any data point for which $a_n = 0$ will not appear in the sum and hence plays no role in making predictions for new data points
- The remaining data points are called support vectors. Because they satisfy $t_n y(x_n) = 1$, they correspond to points that lie on the maximum margin hyperplanes in feature space

We note that $t_n y(x_n) = 1$ then:

$$t_n\Big(\sum_{m\in S} a_m t_m k(x_n, x_m) + b\Big) = 1$$

$$\Rightarrow b = \frac{1}{N_S}\sum_{n\in S}\Big(t_n - \sum_{m\in S} a_m t_m k(x_n, x_m)\Big)$$

Inner Product Kernels

Type of Support Vector Machine	Inner Product Kernel $K(x,x_i)$, I = 1, 2, …, N	Usual inner product
Polynomial learning machine	$(x^T x_i + 1)^p$	Power *p* is specified *a priori* by the user
Radial-basis function (RBF)	$\exp(1/(2\sigma^2)\|\|x-x_i\|\|^2)$	The width σ^2 is specified *a priori*
Two layer neural net	$\tanh(\beta_0 x^T x_i + \beta_1)$	Actually works only for some values of β_0 and β_1

SVM Kernel (cont.)

Figure 3: SVM Kernel

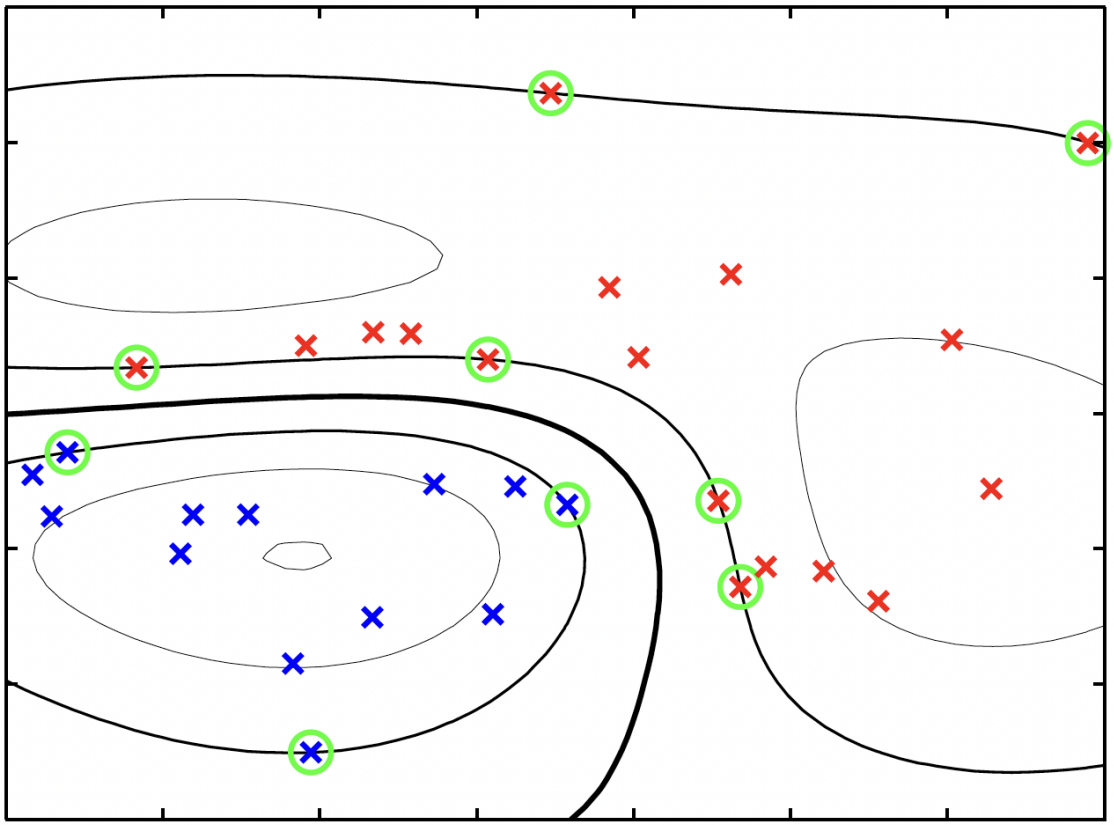

Figure 4: RBF Kernel