SVM (support vector machine classifier)

Resources: Stanford lecture notes by Andrew Ng and an MIT video lecture by Patrick H. Winston

SVM are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The SVM learns a hyper-plane in parameter space which separates the two classes s.t the distance of closest point to the plane (support vectors) is maximal and by this, giving the classifier maximal confidence.

The above is a way to visualize to way SVM works, mathematically, it learns by minimizing hinge-loss function given L_2 regularization in the weight vector.

SVM - Derivation

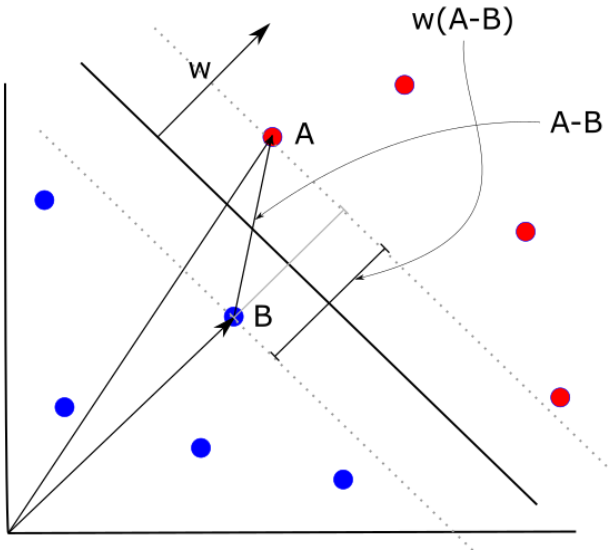
Support vector machines, being a discriminative learning algorithm, learn the posterior probability $p(y|X)$ and use it to infer upon examples, the intuition behind SVM is acquired geometrically by describing the probability function as a separating hyperplane in feature space between the different classes.

Margins are a key level behind the geometric interpretation of SVM, we wish to find a function $p(y|x) = h_\theta(x)$ s.t for each example x the probability of it being in a certain y will be closest to 1, this will tell us that the classifier is a confident one.

Stating the optimization problem

To get in to the geometric intuition, we start with a few definitions:

1. A plane, is defined by a point r_0 and a unit vector perpendicular to it $\vec{w}$ s.t every point $\vec{r}$ on the plane satisfies $w^T(r - r_0) = 0$, if we choose $\vec{b} \equiv w^T r_0$ land $x \equiv r$ we get that every point $\vec{x}$ on the plane satisfies $0 = w^T x + b$.
2. A point $\vec{A}$ (WLOG) above the hyperplane which corresponds to example x_i and label $y_i = 1$, a point $\vec{B}$ below hyperplane with label $y = -1$ both points are on the **on the boundary** of the plane and are among that points of minimal distance from it.



The next step is choosing w, b in such a way that for $y_i = 1$ we get $w^T x_i + b \geq 1$ and for $y_i = -1$ we have $w^T x + b \leq 1$. This can be combined to $y_i(w^T x_i + b) \geq 1$. Noticing that the margins of the separating hyperplane is $\frac{w^T}{||w||}(A - B)$, and having both A and B that satisfy the previous equation exactly s.t $w^T x_i = y_i - b$, we get the margin to be $\frac{2}{||w||}$, by:

$$\text{margin} = \frac{w^T}{||w||}(A - B) = \frac{1}{||w||}(1 - b - (-1 - b)) = \frac{2}{||w||}$$

From this, one can see that the demand for making the margin as wide as possible becomes minimizing w (or w^2) while keeping the classification demand, put simply, the optimization problem becomes:

$$\min_{w,b} \ \frac{1}{2}||w||^2 \ \ \text{s.t} \ \ \forall_i \ \ y_i \left(w^T x_i + b\right) - 1 \geq 0$$

Solving the optimization problem

The form of the above problem as optimization subject to a constraint calls for the use of Lagrange multipliers in addition with solving the dual to the above problem, I will not go into the details here (it is very well written is the above notes) but the takeaway massage from following the derivation is:

The resulting optimal w, b which gives maximal boundaries is received to be:

$$w^T x = \left(\sum_{i \to m} \alpha_i y^i x^i\right)^T x = \sum_{i \to m} \alpha_i y^i \left\langle x^i, x \right\rangle$$

with the α's (a result of the optimization) being **non-zero** only for the x_i points which satisfy $y_i \left(w^T x_i + b\right) - 1 = 0$ which are **the so called support vectors**. Further, one can see that in order to make a prediction on a general x, all that is needed is it's **dot product** with the support vectors, this means that if a complicated (and even infinite) feature space transformation is chosen for the x's, conversion of every example x is not needed, only it's dot product in the given feature space, this is the essence of the **Kernel trick** in which a feature space that offers better separation is chosen and features are mapped to it in a tangible calculation effort.