Separating Hyperplanes

when data can be separated by a linear boundary given by $\{x : \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0\}$

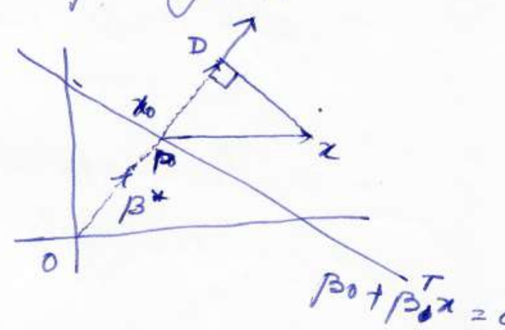
For points that are on one side of the plane

$$\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 > 0$$

and for the other $\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 < 0$

Classifiers that compute the linear combination of input features and return the sign are called <u>Perceptrons</u>

In 3D Geometry, the linear algebra of the hyperplane can be represented as follows:



- Consider a hyperplane or an affine set L given as $f(x) = \beta_0 + \beta^T x = 0$

  For $\mathbb{R}^2$, this is a line

- For any 2 points x_1 & x_2 on the line L

  $\beta^T(x_1 - x_2) = 0$ & hence $\beta^* = \frac{\beta}{\|\beta\|}$ is a unit normal vector.

- For any point x_0 on the surface of L $\beta_0 + \beta^T x_0 = 0$

  or $\beta^T x_0 = -\beta_0$.

- The signed distance of any arbitrary point x with coord(x) to L is given by length of PD in the figure, as.

  $$\vec{PX} = (x - x_0)$$

  Distance PD is the projection length of PX on the vector $\vec{OP}$ given as

  $$\beta^{*T}(x - x_0) = \frac{\beta^T(x - x_0)}{\|\beta\|}$$

  $$= \frac{\beta_0 + \beta^T x}{\|\beta\|} = \frac{f(x)}{\|\beta\|} = \frac{f(x)}{\|f'(x)\|}$$

- Thus $f(x)$ is proportional to the signed distance from x to the hyperplane $f(x) = 0$

(2)

In perceptron Learning algorithm, the method tries to find a separating hyperplane by minimizing the distance of the misclassified points to the decision boundary.

So it tries to minimize $D(\beta, \beta_0) = -\sum_{i \in M} y_i(x_i^T\beta + \beta_0)$

where M is the indexes of the misclassified point

(Details in ESL-2 Page 131)

However there are number of issues with this algorithm

- when the data is separable, there are many solutions and which is found depends on the starting value
- The finite number of steps in the gradient descent may be large. Smaller the learning rate η, longer is the time
- when data is not separable, algorithm will not converge and cycles would be developed.

<u>Optimal Separating Hyperplane.</u>

Consider the optimization problem.

$$\max_{\beta_0, \beta} M$$

$$\text{st.} \quad \|\beta\| = 1$$

$$y_i(x_i^T\beta + \beta_0) \geq M, \quad i = 1, 2 .. N.$$

- This ensures that all the points are at least a signed distance M from the decision boundary defined by β & β_0.
- We seek to find the largest such M associated with these parameters.

We get rid of $\|\beta\| = 1$ by ~~inputing this~~ embedding this ~~constr~~ condition in the 2nd constraint

so $\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$ (This redefine β_0 also)

or $y_i(x_i^T\beta + \beta_0) \geq M\|\beta\|$

For any β, β_0 satisfying these inequalities any positively scaled multiple of β, β_0 would also satisfy

Because $k y_i(x^T\beta + \beta_0) \geq M\|k\beta\|$

So we arbitrarily set $\|\beta\| = \frac{1}{M}$

Thus $\max M$ ~~on~~ would ~~mean~~ be equivalent to

$$\underset{\beta, \beta_0}{\text{minimizing}} \ \frac{1}{2}\|\beta\|^2 \ \text{~~for } \beta_0, \beta~~$$

$$\text{s.t. } y_i(x_i^T\beta + \beta_0) \geq 1 \quad i = 1, 2, \ldots N$$

The constraint now define an empty slab around margin of thickness M i.e $\frac{1}{\|\beta\|}$. We choose β, β_0 to maximize its thickness.

This is a convex optimization problem, since $\frac{1}{2}\|\beta\|^2$ is convex

Thus the optimization function and constraints are given as

$$\min_{\beta, \beta_0} \frac{1}{2}\|\beta\|^2$$

$$\text{s.t. } y_i(x_i^T\beta + \beta_0) \geq 1 \quad i = 1, 2 \ldots N.$$

Solve using Lagrange's multiplier (Primal)

$$L_P = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i\left[y_i(x_i^T\beta + \beta_0) - 1\right] \quad (1)$$

To minimize $\frac{\partial L_P}{\partial \beta} = 0$ $\frac{\partial L_P}{\partial \beta_0} = 0$

$$\frac{\partial L_P}{\partial \beta} = 0 \Rightarrow \beta - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \text{ or } \beta = \sum_{i=1}^{N} \alpha_i y_i x_i \quad (2)$$

$$\frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow 0 - \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ or } \sum_{i=1}^{N} \alpha_i y_i = 0 \quad (3)$$

Substituting (2) & (3) in (1), we get the Wolfe-Dual

$$L_D = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i y_i x_i^T\beta + \beta_0\sum_{i=1}^{N} \alpha_i y_i \beta_0 + \sum_{i=1}^{N} \alpha_i$$

(4)

$$\text{or } L_D = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i y_i x_i^T \beta + 0 + \sum_{i=1}^{N} \alpha_i$$

$$= \frac{1}{2}\left(\beta^T\beta\right) - \sum_{i=1}^{N} \alpha_i y_i x_i^T \beta + \sum_{i=1}^{N} \alpha_i$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i x_i^T \beta\right) - \sum_{i=1}^{N} \alpha_i y_i x_i^T \beta + \sum_{i=1}^{N} \alpha_i$$

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i y_i \alpha_k y_k x_i^T x_k$$

$$\text{subject to } \alpha_i \geq 0 \ \& \ \sum_{i=1}^{N} \alpha_i y_i = 0 \quad - (4)$$

This can be solved by a software. Additionally Karush-Kuhn-Tucker condition must be satisfied as that includes

condition (1), (2), (3), (4) and

$$\alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - 1 \right] = 0 \ \forall i \quad - (5)$$

Thus from (5) we find

- If $\alpha_i > 0$ then $y_i (x_i^T \beta + \beta_0) - 1 = 0$ or x_i lies on the boundary of the slab.
- If $y_i (x_i^T \beta + \beta_0) - 1 > 0$, then $\alpha_i = 0$, i.e ~~for~~ points outside the slab, don't effect the outcome.

Thus points x_i on the boundary are called support points.

- This is unlike LDA & Logistic Regression, where all points affect the outcome, even points ~~eve~~ which are far from the decision boundary.

Handling class overlaps

Considering that the classes overlap in the feature space optimization function would be

$$\max_{\beta_0, \beta} M$$
$$\text{s.t.} \quad \|\beta\|^2 = 1$$
$$\& \quad y_i(x_i^T \beta + \beta_0) \geq M(1-\epsilon_i) \qquad \epsilon_i \text{ is a slack variable.}$$
$$\text{s.t.} \quad \sum \epsilon_i \leq C$$

Missclassifications occur when $\epsilon_i > 1$, so bounding $\sum \epsilon_i$, we are bounding the total number of misclassifications.

Modifying the optimization function

As in previous case we can drop the norm constraint on β define $M = \frac{1}{\|\beta\|}$ and write the equivalent form

$$\min_{\beta, \beta_0} \|\beta\|$$
$$\text{s.t.} \quad y_i(x_i^T \beta + \beta_0) \geq 1 - \epsilon_i \quad \forall i = 1, 2 .. N.$$
$$\epsilon_i \geq 0$$
$$\sum \epsilon_i \leq C.$$

We can combine the ~~cost function~~ ~~constant~~ budget constraint in the objective function as

$$\min_{\beta, \beta_0} \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \epsilon_i$$
$$\text{s.t.} \; \epsilon_i \geq 0, \; y_i(x_i^T \beta + \beta_0) \geq 1 - \epsilon_i \;\; \forall i \qquad - (6)$$

If C is higher then $\sum \epsilon_i$ has to be smaller, i.e less room for errors.

The primal function then becomes

$$L_p = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\left[y_i(x_i^T\beta + \beta_0) - (1-\epsilon_i)\right] - \sum_{i=1}^{N}\mu_i\epsilon_i \qquad - (7)$$

(6) We need to minimize wrt β, β_0 & ϵ_i. Deriving the derivatives $\frac{\partial L_P}{\partial \beta}, \frac{\partial L_P}{\partial \beta_0}$ & $\frac{\partial L_P}{\partial \epsilon_i}$ and setting to 0, we get

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i \quad - (8)$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i \quad - (9)$$

$$\alpha_i = C - \mu_i \;\forall i \quad - (10)$$

where $\alpha_i, \mu_i, \epsilon_i \geq 0 \;\forall i$

Substituting (8), (9) & (10) into (7) we get.

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

This gives a lower bound of the solution of the primal problem for ~~and hence need~~ any feasible point and hence needs to be maximized.

We maximize L_D subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

Further the KKT condition include the following constraints

$$\alpha_i [y_i(x_i^T \beta + \beta_0) - (1 - \epsilon_i)] = 0 \quad - (11)$$

$$\mu_i \epsilon_i = 0 \quad - (12)$$

$$\& \quad y_i(x_i^T \beta + \beta_0) - (1 - \epsilon_i) \geq 0 \quad - (13)$$

The dual maximization is a simpler convex quadratic programming problem than the primal and can be solved using standard techniques.

Given the solutions $\hat{\beta}_0$ & $\hat{\beta}$ the decision function can be written as $\hat{G}(x) = \text{sign}[\hat{f}(x)] = \text{sign}[x^T \hat{\beta} + \hat{\beta}_0]$

Support Vector machines and kernels

Make the procedure more flexible by enlarging the feature space using basis expansions.

For example for each point we can select m basis functions $h_1(x), h_2(x) \ldots h_m(x)$ & produce the function

$$\hat{f}(x) = [h_i(x)]^T \hat{\beta} + \hat{\beta}_0$$

For decision $\hat{G}(x) = \text{sign}[\hat{f}(x)]$

For SVM classifier, ~~this~~ the dimension of the enlarged feature space is allowed to get very large, possibly infinite.

The Lagrange's dual function ~~a~~ has the form

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \, h(x_i)^T h(x_k)$$

$$\text{Thus } f(x) = [h(x)]^T \beta + \beta_0$$

$$= \sum_{i=1}^{N} [h(x)]^T \alpha_i y_i h(x_i) + \beta_0$$

$$= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$$

Thus we require to find the inner-product.

~~However~~ We need not specify the transformation at all, but require only the knowledge of a kernel function

$$K(x, x_i) = \langle h(x), h(x_i) \rangle$$