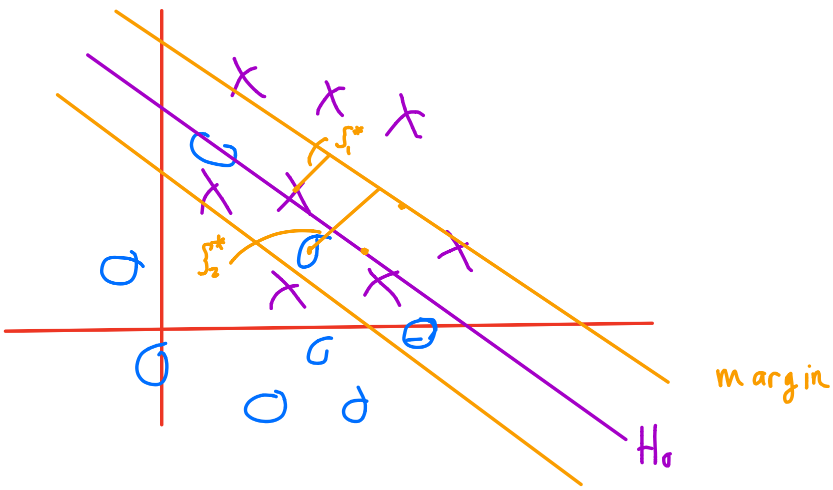
Non-Separable Data SVM



Rmk: We penalize even if H_0 classifies correctly.

That is we penalize all data with the margin of the classifier.

Goal: Given width of margin C we wish to minimize $\sum_{i \in \text{margin}} \xi_i^*$

with respect to the function (classifier) $f(x) = x^T\beta$

Soft Margin Relaxation:

$$\underset{(\beta_0,\beta)}{\operatorname{argmin}} \|\beta\| \quad \text{s.t.} \quad y_i(x_i^T\beta+\beta_0) \geq 1-\xi_i \; ; \; \xi_i \geq 0 \; ; \; \sum_{i=1}^{N} \xi_i \leq c$$

Rmk: $\xi_i = \|\beta\| \xi_i^*$

To solve this problem, we formulate the Lagrangian

$$L_P = \frac{1}{2}\|\beta\|_2^2 + \gamma \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{n} \alpha_i \left[y_i(x_i^T\beta - \beta_0) - (1-\xi_i) \right] - \sum_{i=1}^{n} \mu_i \xi_i$$

Rmk: γ encodes the "c" constraint.

$\mu_i, \alpha_i, \xi_i \geq 0$

Solving...

$$0 = \frac{\partial L_P}{\partial \beta} \Rightarrow \beta = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$0 = \frac{\partial L_P}{\partial \beta_0} \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i$$

$$0 = \frac{\partial L_P}{\partial \xi_i} \Rightarrow \alpha_i = \gamma - \mu_i \quad ; \; i = 1, \ldots, n$$

Setting up the dual Lagrangian

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

$$0 \leq \alpha_i \leq \gamma \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

This gives the KKT conditions

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

$$\mu_i \xi_i = 0$$

$$y_i (x_i^T \beta + \beta_0) - (1 - \xi_i) \geq 0$$

<u>So</u>: $\vec{\beta} = \sum_{i=1}^{n} \alpha_i y_i x_i$

<u>Rmk</u>: When $y_i [(x_i^T \beta + \beta) - (1 - \xi_i)] = 0$ then $\hat{\alpha}_i > 0$

Subconditions:

(A) $\xi_i = 0 \implies x_i$ on margin

Feature Maps

Replace data with $\mathcal{J} = \{h(x_i), y_i\}_{i=1}^{N}$

We'd like to consider the kernel method

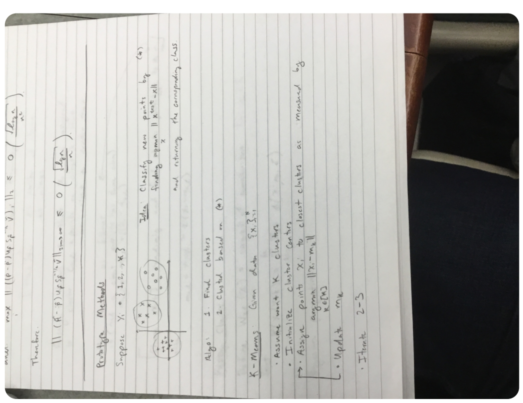
$K(x,y) = h(x)h(y)^T$ and the modified dual

$$\sum \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i' \alpha_k' y_i y_k \, h(x_i) h(x_k)^T, \quad 0 \le \alpha_i' \le \gamma'$$

$$\sum \alpha_i' y_i = 0$$

Ex: $K(x,y) = (1 + x^T y)^2 = h(x)^T h(y)$

$$h(x) = (1 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 \quad x_1^2 \quad x_2^2 \quad \sqrt{2}x_1x_2)^T$$



K-means was done greedily, ignoring the other classes.

__Learning Vector Quantization__ (LVQ)

- Assume we have centers indexed temporally

$$\{ m_{k\ell} \}_{\ell=1}^{R_\ell} \qquad \ell = \text{number of clusters}$$
$$k = \text{class.}$$

- Reassign clusters so that x_i are assigned to cluster $C_{k\ell}$ = points in class k that are closest to center $m_{k\ell}$.

- Move means as follows:
  if ($m_{k\ell}$ is closest prototype has same class)
  $$m_{k\ell} = m_{k\ell} + \varepsilon(x_i - m_{k\ell})$$

[Choose $x_i \in J$ randomly and find closests prototype to x_i]

else
$$m_{k\ell} = m_{k\ell} - \varepsilon(x_i - m_{k\ell})$$

__Fuzzy Cluster__

Define $W_{kr}(x_i)$ = weight of membership for $x_i \in R$

__Start__

$$P(X \mid Y=k) = \sum_{k=1}^{R_k} \pi_{kr} \, \phi(x_i; \mu_{kr}, \Sigma)$$

Bayesian perspective