Support Vector Machines

B. Ravindran

Problem Setting

$$X^{(1)} = \langle 0.15, 0.25 \rangle, Y^{(1)} = -1$$
$$X^{(2)} = \langle 0.4, 0.45 \rangle, Y^{(2)} = +1$$
$$\vdots$$

The input can be thought of as $X = (X_1, X_2, \cdots, X_p)$

The X_i are the features that describe the input.

The output is either a categorical variable, typically denoted by G or a continuous variable denoted by Y

The i-th instance of the input is denoted as x_i and output is denoted as y_i

We loosely state the learning problem as given a value of input X make a good prediction $\hat{Y}$ of Y or $\hat{G}$ of G

Problem Setting

$\chi \subseteq \Re^p$ is the input space

$X = \left(X_1, X_2, \cdots X_p \right)$ is a random variable describing the input

$\Upsilon \subseteq \Re$ or Γ is the output space

Y is a random variable describing the output

$p(X,Y)$ is the data distribution

$p(X,Y) = p(Y|X)p(X)$

$p(Y|x)$ is the predicted output probabilities given an input x

Problem Setting

Assume that $Y = f(X) + \varepsilon$,
where $E(\varepsilon) = 0$, and independent of X.

Note that $f(x) = E\left(Y \middle| X = x\right)$

Goal is to find a f that minimizes expected prediction error (EPE).

For squared error loss, $\text{EPE}\left(f\right) = E\left(Y - f(X)\right)^2$

For classification - assume a $K \times K$ loss matrix, $\mathbf{L}$.

The $\mathbf{L}_{ij}$ entry is the loss suffered by classifying class i as class j.

Typically use a $0-1$ loss function, where the off-diagonal entries of $\mathbf{L}$ are 1.

Problem Setting

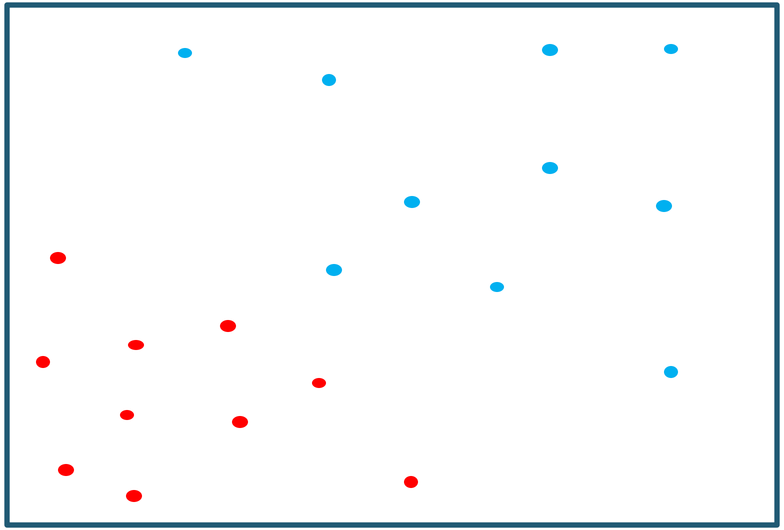
- Different assumptions on f lead to different classifiers/predictors
  - Assuming f is a linear function of the input leads to linear regression
  - Assuming f is constant over small regions
    - kNN assumes the regions are identified by neighbours
    - Decision Trees assume that the regions are given by axis parallel hyperrectangles
- Directly model the posterior probabilities
  - Naïve Bayes, Logistic Regression, LDA, etc.

Support Vector Machines

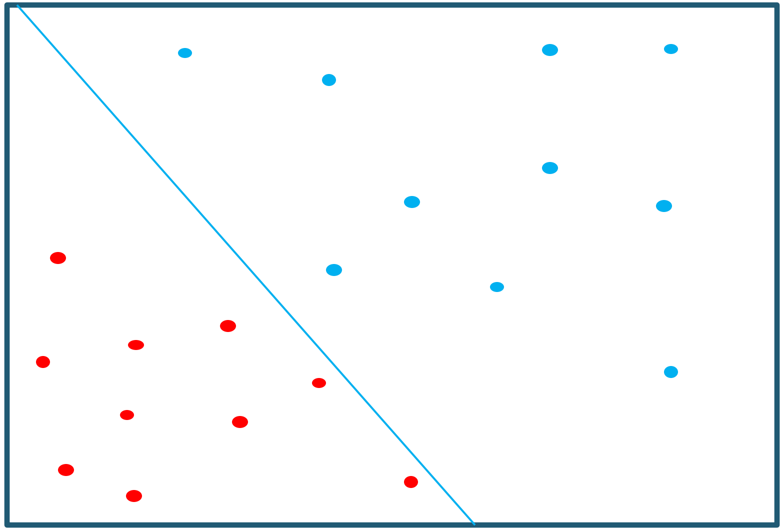
Decision Boundaries

- Typically boundaries depend on the assumptions that we make
- Do we have a notion of *optimal* decision boundary?
  - One that minimizes error on training data
  - What if there are several minimizers?

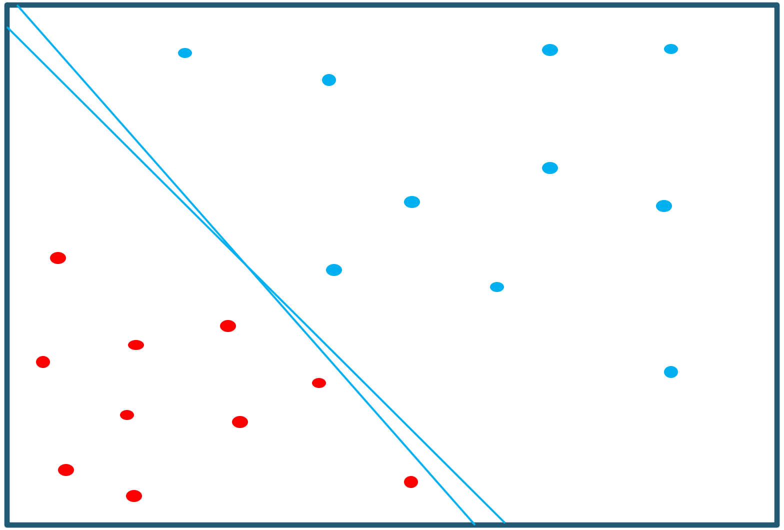
Optimal Separating Hyperplane



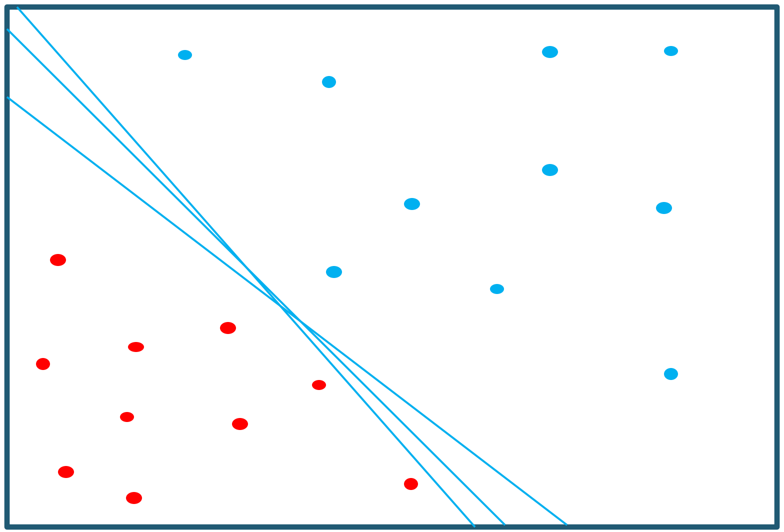
Optimal Separating Hyperplane



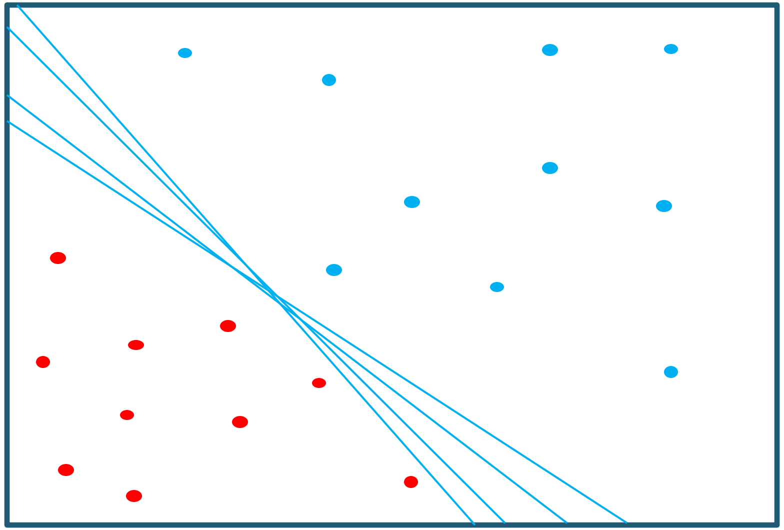
Optimal Separating Hyperplane



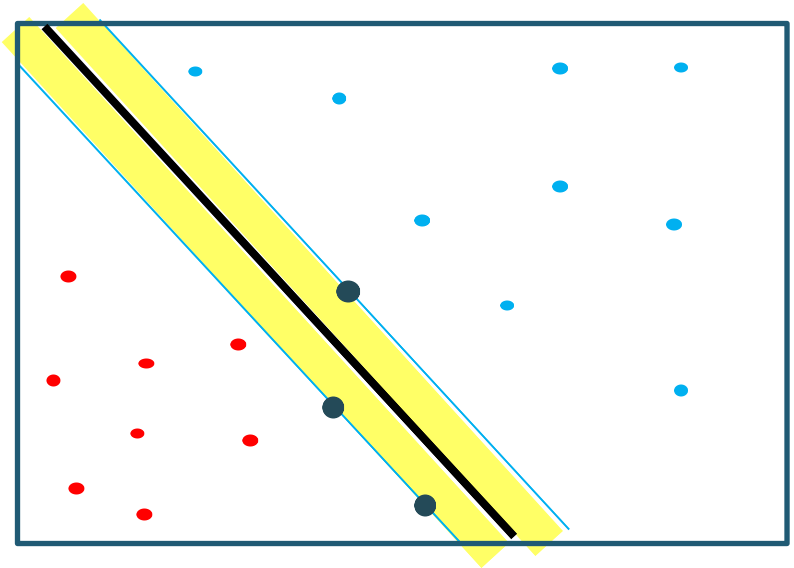
Optimal Separating Hyperplane



Optimal Separating Hyperplane



Optimal Separating Hyperplane



Mathematics of Hyperplanes

L represented by $f(x) = \beta_0 + \beta^T x = 0$

is a hyperplane (line in $\mathbb{R}^2$)

1) For any two points x_1, x_2 lying in L,

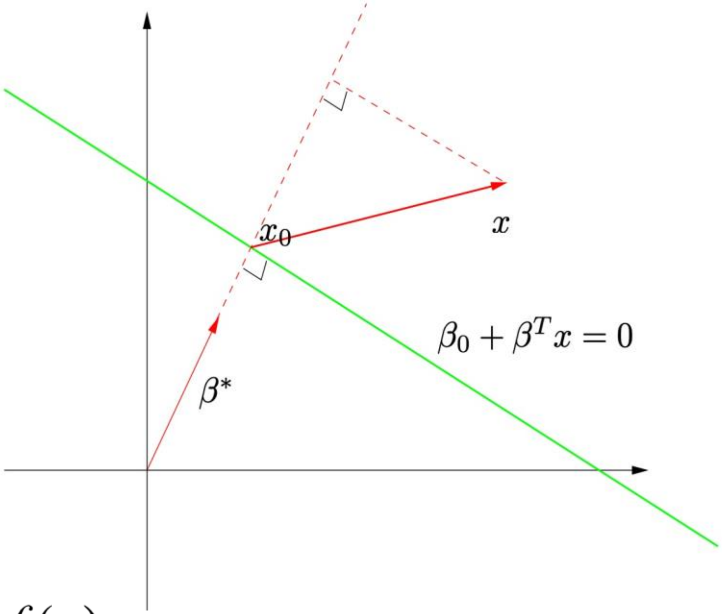
   $\beta^T (x_1 - x_2) = 0$ & hence

   $\beta^* = \frac{\beta}{\|\beta\|}$ is normal to L

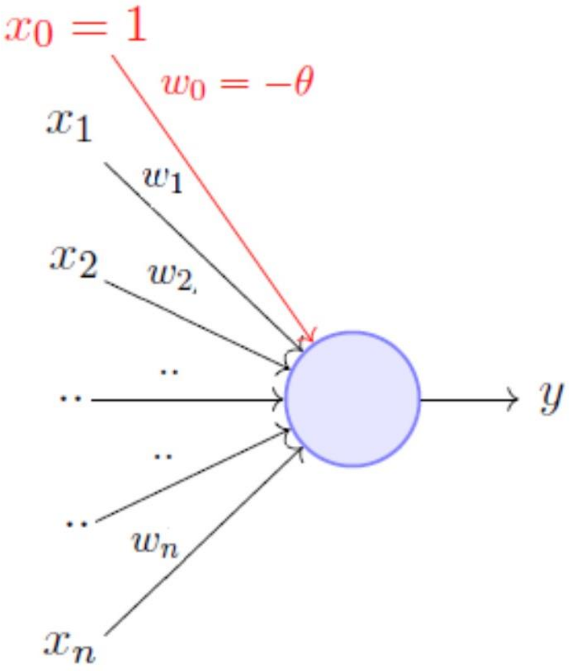
2) For any point x_0 in L, $\beta^T x_0 = -\beta_0$

3) The signed distance of any point x to L

   is given by $\beta^{*T}(x - x_0) = \frac{1}{\|\beta\|}(\beta^T x + \beta_0) = \frac{f(x)}{\|f'(x)\|}$ where $x_0 \in L$



Perceptron



$$y = 1 \quad if \sum_{i=0}^{n} w_i * x_i \geq 0$$

$$= -1 \quad if \sum_{i=0}^{n} w_i * x_i < 0$$

$$where, \quad x_0 = 1 \quad and \quad w_0 = -\theta$$

From TDS

- Simplest form of neural networks
- w_1 , w_2 … w_n are the parameters

Perceptron Learning Algorithm

- Perceptron tries to learn by minimizing the distance of misclassified points to the decision boundary
- If $y_i = 1$ is misclassified, then $x_i^T \beta + \beta_0 < 0$ and vice versa
- $\min D(\beta, \beta_0) = - \sum_{i \in \nu} y_i (x_i^T \beta + \beta_0)$ where ν is the set of misclassified points
- Non negative and proportional to distance of points to decision boundary

$$\frac{\partial D(\beta, \beta_0)}{\partial \beta} = - \sum_{i \in \nu} y_i x_i \, , \quad \frac{\partial D(\beta, \beta_0)}{\partial \beta_0} = - \sum_{i \in \nu} y_i \qquad \nu : fixed$$

- Use stochastic gradient descent,

update after each point [Add the misclassified vectors to the solution!]

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i \, x_i \\ y_i \end{pmatrix} ; \qquad \rho = 1 \text{ usually}$$

Perceptron Learning Algorithm

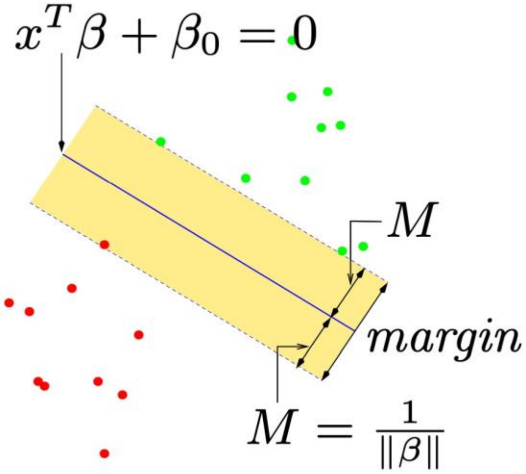
- If the data is linearly separable, converges to <u>some</u> separating hyperplane
  - Solution depends on starting values of weights
  - Number of steps can be very large which is addressed using basis expansions
- Non separable data leads to cycles.
  - Hard to detect since very long

Optimal Hyperplane

- Maximize distance of the closest point to the hyperplane $\max_{\beta, \beta_0, \|\beta\|=1} M$

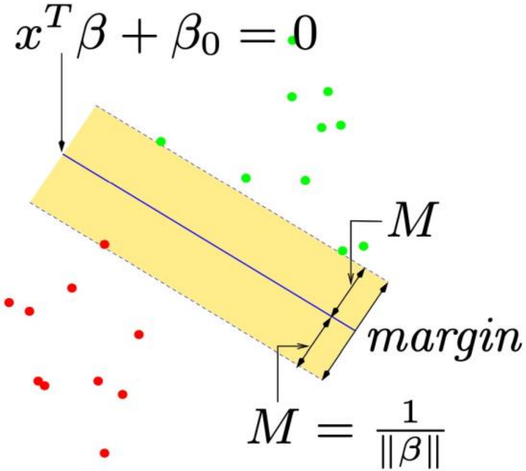
$$\text{subjet to } y_i(x_i^T \beta + \beta_0) \geq M, \; i = 1, \ldots, N,$$

- All points are at least a signed distance M from decision boundary



Optimal Hyperplane

- Maximize distance of the closest point to the hyperplane
$$\max_{\beta,\beta_0,\|\beta\|=1} M$$
$$\text{subjet to } y_i(x_i^T\beta+\beta_0) \geq M,\ i=1,\ldots,N,$$



- All points are at least a signed distance M from decision boundary
- Get rid of $\|\beta\|=1$ by $\frac{1}{\|\beta\|}y_i(x_i^T\beta+\beta_0)\geq M$

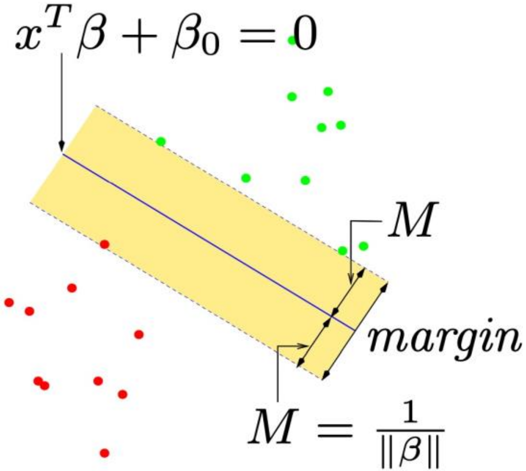
$$\text{i.e. } y_i(x_i^T\beta+\beta_0)\geq M\|\beta\|$$

Optimal Hyperplane

- Maximize distance of the closest point to the hyperplane

$$\max_{\beta, \beta_0, \|\beta\|=1} \quad M$$

$$\text{subjet to } y_i(x_i^T \beta + \beta_0) \geq M, \; i = 1, \dots, N,$$



- All points are at least a signed distance M from decision boundary
- Get rid of $\|\beta\| = 1$ by $\frac{1}{\|\beta\|} y_i(x_i^T \beta + \beta_0) \geq M$

  i.e. $y_i(x_i^T \beta + \beta_0) \geq M \|\beta\|$

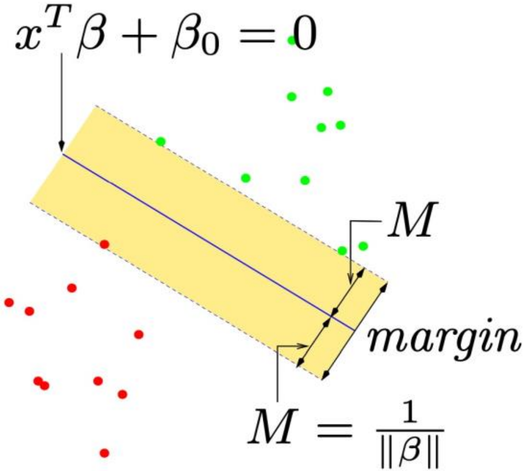
- Any positively scaled β, β_0 satisfies this. Hence, set $\|\beta\| = \frac{1}{M}$

Optimal Hyperplane

- Maximize distance of the closest point to the hyperplane

$$\max_{\beta,\beta_0,\|\beta\|=1} \quad M$$

$$\text{subjet to } y_i(x_i^T\beta+\beta_0) \ \geq \ M, \ i=1,\ldots,N,$$



- All points are at least a signed distance M from decision boundary
- Get rid of $\|\beta\|=1$ by $\frac{1}{\|\beta\|} y_i(x_i^T\beta+\beta_0) \geq M$

i.e. $y_i(x_i^T\beta+\beta_0) \geq M\,\|\beta\|$

- Any positively scaled $\beta,\ \beta_0$ satisfies this. Hence, set $\|\beta\| = \frac{1}{M}$

$$\min_{\beta,\beta_0} \ \|\beta\|$$

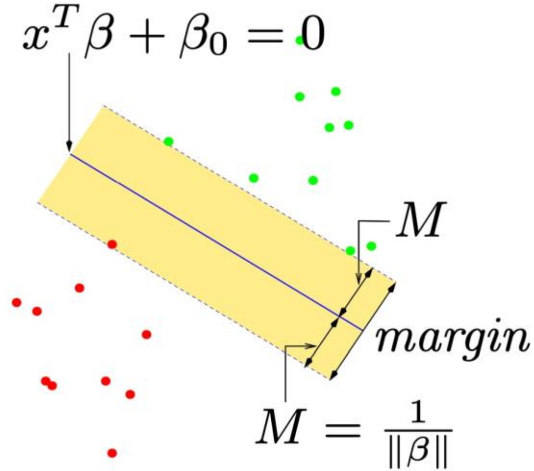
$$\text{subject to } y_i(x_i^T\beta+\beta_0) \ \geq \ 1\,; \ i=1,\ldots\ldots,N$$

Optimal Hyperplane

- Maximize distance of the closest point to the hyperplane

$$\max_{\beta, \beta_0, \|\beta\|=1} M$$

$$\text{subjet to } y_i(x_i^T \beta + \beta_0) \geq M, \; i = 1, \ldots, N,$$



- All points are at least a signed distance M from decision boundary
- Get rid of $\|\beta\| = 1$ by $\frac{1}{\|\beta\|} y_i(x_i^T \beta + \beta_0) \geq M$

  i.e. $y_i(x_i^T \beta + \beta_0) \geq M \|\beta\|$

- Any positively scaled β, β_0 satisfies this.

  Hence, set $\|\beta\| = \frac{1}{M}$

$$\boxed{\begin{aligned} &\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ &\text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1; \; i = 1, \ldots\ldots, N \end{aligned}}$$

- Constraints define a slab around the boundary of thickness $1/\|\beta\|$. This is a convex optimization problem (quadratic objective with linear constraints)
- Lagrangian (primal) $L_P = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i [y_i (x_i^T \beta + \beta_0) - 1]$

setting derivative equal to zero

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

$$0 = \sum_{i=1}^{N} \alpha_i y_i \qquad (2)$$

substituting to get Wolfe Dual

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

subject to $\alpha_i \geq 0$

(Maximize L_D in the positive orthant)

Stationary
Primal Dual feasible
Complementary slackness

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

Karush-Kuhn-Tucker (KKT) Conditions

Stationary

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

$$0 = \sum_{i=1}^{N} \alpha_i y_i \qquad (2)$$

Any solution must satisfy these conditions

Primal

Dual feasibility

$$y_i(x_i^T \beta + \beta_0) \ \geq \ 1 \ ; \ \ i = 1, \ldots\ldots, N \qquad (3)$$

$$\alpha_i \geq 0 \quad \forall i \qquad (4)$$

Complementary slackness

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

Solution

$$\hat{f}_k(x) = \hat{\beta}_0 + x^T \hat{\beta}$$
$$\hat{G}(x) = \text{sign}\, \hat{f}_k(x)$$

Note from KKT conditions we can see:

1. If $\alpha_i > 0$ then $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) = 1$, i.e., x_i is on the boundary of the slab
2. If $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) > 1$, x_i is not on the boundary, and $\alpha_i = 0$

Only when $\alpha_i > 0$ does x_i contribute to the solution, since $\hat{\beta}$ depends on α_i.
Hence such points are known as support points or vectors.

Solution

$$\hat{f}_k(x) = \hat{\beta}_0 + x^T \hat{\beta}$$
$$\hat{G}(x) = \text{sign}\, \hat{f}_k(x)$$

Note from KKT conditions we can see:

1. If $\alpha_i > 0$ then $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) = 1$, i.e., x_i is on the boundary of the slab

2. If $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) > 1$, x_i is not on the boundary, and $\hat{\alpha}_i = 0$

Only when $\hat{\alpha}_i > 0$ does x_i contribute to the solution, since $\hat{\beta}$ depends on $\hat{\alpha}_i$. $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$

Hence such points are known as support points or vectors.

LDA or LR vs SVM?

- LDA/LR pay attention to all data. SVM only to boundary data. Hence, more robust
- If class conditioned densities are truly Gaussian then LDA optimal
- If posteriors are truly logistic then LR optimal
- SVM is distracted by boundary points in such cases

Non-separable case?

$\xi = (\xi_1, \ldots\ldots\ldots, \xi_N)$ Slack Variables

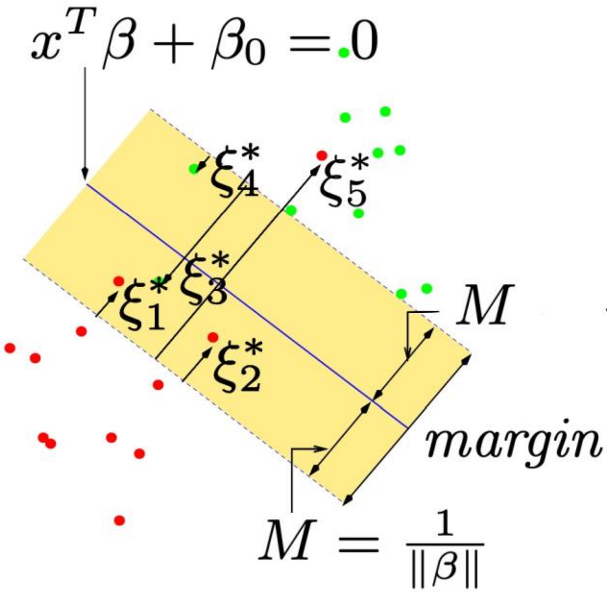
$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i,$$

$$\forall i \xi_i \geq 0; \sum_{i=1}^{N} \xi_i \leq \text{constant}$$

- Simplifying, as earlier

$$\min_{\beta, \beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \xi_i$$

subject to $\xi_i \geq 0$, $y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \ \forall i$,

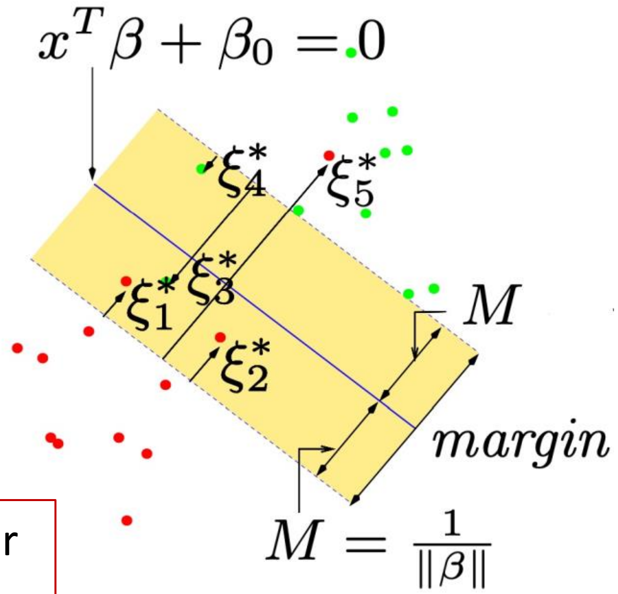


Non-separable case?

$$\xi = (\xi_1, \ldots\ldots\ldots, \xi_N)$$ Slack Variables

$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i,$$

$$\forall i \, \xi_i \geq 0; \sum_{i=1}^{N} \xi_i \leq \text{constant}$$



- Simplifying, as earlier

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i$$

C controls the penalty for the misclassification

subject to $\xi_i \geq 0, \; y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \; \forall i,$

Non-separable Case

- Primal

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

Differentiate w.r.t $\beta, \beta_0 \text{ and } \xi_i$

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i \qquad \alpha_i, \mu_{\mathrm{i}}, \xi_{\mathrm{i}} \geq 0$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i$$

$$\alpha_i = C - \mu_i, \ \forall i$$

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j \ ; \quad 0 \leq \alpha_i \leq C \ \text{ and} \sum_{i=1}^{N}\alpha_i \ y_i = 0$$

Karush-Kuhn-Tucker (KKT) Conditions

Stationary

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

$$0 = \sum_{i=1}^{N} \alpha_i y_i \qquad (2)$$

Any solution must satisfy these conditions

Primal
Dual feasibility

$$y_i(x_i^T \beta + \beta_0) - (1 - \xi_i) \geq 0 \; ; \; i = 1, \ldots\ldots, N \qquad (3)$$

$$\alpha_i = C - \mu_i \quad \forall i \qquad (4)$$

Complementary slackness

$$\alpha_i [y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] = 0 \;\; \forall \; i \qquad (5)$$

$$\mu_i \xi_i = 0 \;\; \forall i \qquad (6)$$

Solution

$$\hat{f}_k(x) = \hat{\beta}_0 + x^T \hat{\beta}$$
$$\hat{G}(x) = \text{sign}\, \hat{f}_k(x)$$

Note from KKT conditions we can see:

1. If $\alpha_i > 0$ then $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) - (1 - \hat{\xi}_i) = 0$

Among these, if $\hat{\xi}_i = 0$, then $0 < \hat{\alpha}_i < C$. If $\hat{\xi}_i > 0$, then $\hat{\alpha}_i = C$

2. If $y_i(\hat{\beta}_0 + x_i^T \hat{\beta}) - (1 - \hat{\xi}_i) > 0$ x_i then $\hat{\alpha}_i = 0$

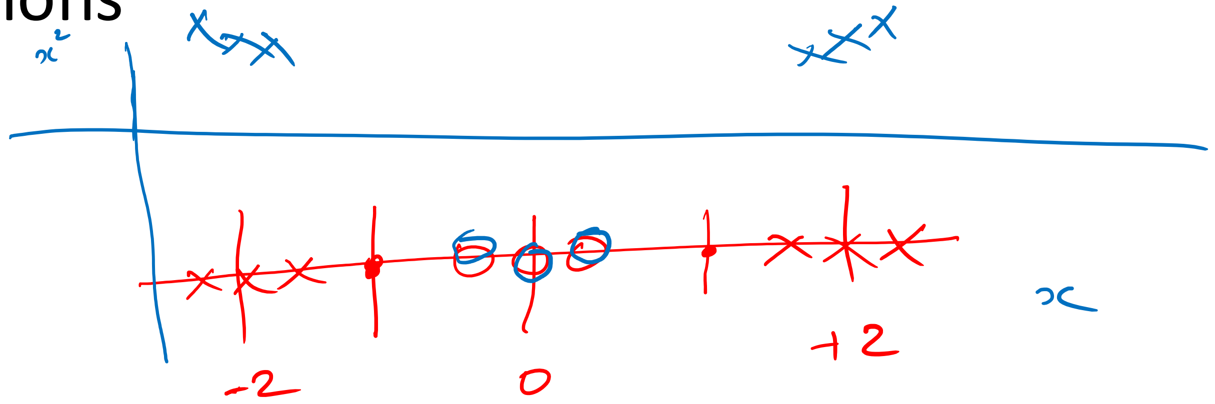
Only when $\hat{\alpha}_i > 0$ does x_i contribute to the solution, since $\hat{\beta}$ depends on $\hat{\alpha}_i$. $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$

Hence such points are known as support points or vectors.

What about non-linear separation?

$\langle (x, x^2)\ y \rangle$ $\langle x, y \rangle$

- Basis Expansion! ✓
- Cover's theorem
  - Data more likely to be linearly separable in higher dimensions



Basis Expansion

- Recall: $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \,; \quad 0 \le \alpha_i \le C$

$$f(x) = x^T \beta + \beta_0$$
$$= \sum_{i=1}^{N} \alpha_i \, y_i \, x^T \, x_i + \beta_0$$

- Transformed data $x \to h(x)$

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle$$

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

Basis Expansion

- Recall: $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j\,; \quad 0 \le \alpha_i \le C$

$$f(x) = x^T \beta + \beta_0$$
$$= \sum_{i=1}^{N} \alpha_i\, y_i\, x^T\, x_i + \beta_0$$

- Transformed data $x \to h(x)$

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j)\rangle$$

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

x always appears as an inner product

Kernel Functions

- A similarity measure define through inner products $K(x,x') = \langle h(x), h(x') \rangle$ [Mercer Kernel]

Choices of K: dth- Degree polynomial: $K(x,x') = (1+\langle x,x' \rangle)^d$

Gaussian: $K(x,x') = \exp(-\gamma \|x-x'\|^2)$

Neural network: $K(x,x') = \tanh(\kappa_1 \langle x,x' \rangle + \kappa_2)$

For a 2-dimensional polynomial in second order

$$
\begin{aligned}
K(X,X') &= (1+\langle X,X' \rangle)^2 \\
&= (1+X_1X_1'+X_2X_2')^2 \\
&= 1+2X_1X_1'+2X_2X_2'+(X_1X_1')^2+(X_2X_2')^2+2X_1X_1'X_2X_2'
\end{aligned}
$$

Polynomial Kernel

$$h_0(x) = 1,\ h_1(x) = \sqrt{2}x_1,\ h_2(x) = \sqrt{2}x_2,\ h_3(x) = x_1^2,$$

$$h_4(x) = x_2^2,\ h_5(x) = \sqrt{2}x_1x_2$$

$$f\hat{(x)} = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0$$

$$\boxed{\begin{aligned} K(X,X') &= (1+\langle X, X'\rangle)^2 \\ &= (1+X_1X_1'+X_2X_2')^2 \\ &= 1+2X_1X_1'+2X_2X_2'+(X_1X_1')^2+(X_2X_2')^2+2X_1X_1'X_2X_2' \end{aligned}}$$

$$\begin{aligned} h(x) = h(x_1, x_2) &= \Big(h_0(x)\ h_1(x)\ h_2(x)\ h_3(x)\ h_4(x)\ h_5(x)\Big) \\ &= \Big(1\ \sqrt{2}x_1\ \sqrt{2}x_2\ x_1^2\ x_2^2\ \sqrt{2}x_1x_2\Big) \\ h(x') &= \Big(1\ \sqrt{2}x_1'\ \sqrt{2}x_2'\ x_1'^2\ x_2'^2\ \sqrt{2}x_1'x_2'\Big) \\ h(x') \cdot h(x) &= \Big(1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_2x_1'x_2'\Big) \end{aligned}$$

Polynomial Kernel

$$h_0(x) = 1,\ h_1(x) = \sqrt{2}x_1,\ h_2(x) = \sqrt{2}x_2,\ h_3(x) = x_1^2,$$
$$h_4(x) = x_2^2,\ h_5(x) = \sqrt{2}x_1x_2$$
$$f\hat{(x)} = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0$$

$$\boxed{\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1X_1' + X_2X_2')^2 \\ &= 1 + 2X_1X_1' + 2X_2X_2' + (X_1X_1')^2 + (X_2X_2')^2 + 2X_1X_1'X_2X_2' \end{aligned}}$$

- Gaussian Kernels project to an infinite dimensional space!

Summary

- Very powerful classifiers
  - Optimal separating hyperplanes, but with a quadratic objective function
  - Solve more efficiently in the *dual* space
- *Structural Risk* minimization
- Kernels allow us to operate in high dimensional space with lesser computation
- Tune C using cross validation
  - Other kernel parameters as well
- Radial Basis or Gaussian kernels are the most popular
  - Different data types have different kernels!