SVM: Optimal Hyperplane

- Use Kuhn-Tucker theorem to convert our problem to:

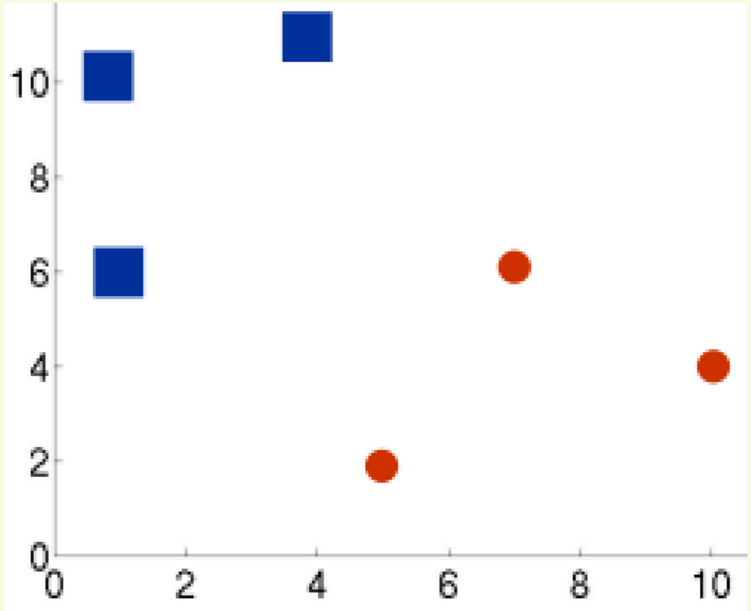
$$\boxed{\begin{array}{ll} \text{maximize} & L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j x_i^t x_j \\ \text{constrained to} & \alpha_i \geq 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i z_i = 0 \end{array}}$$

- $\alpha = \{\alpha_1, \ldots, \alpha_n\}$ are new variables, one for each sample
- $L_D(\alpha)$ can be optimized by quadratic programming
- $L_D(\alpha)$ formulated in terms of α
  - it depends on w and w_0 indirectly

SVM: Example using Matlab

- Class 1: [1,6], [1,10], [4,11]
- Class 2: [5,2], [7,6], [10,4]
- Let's pile all data into array $\boldsymbol{X}$

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 10 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 10 & 4 \end{bmatrix}$$



- Pile $\boldsymbol{z}_i$'s into vector $z = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$

- Matrix $\boldsymbol{H}$ with $H_{ij} = z_i z_j x_i^t x_j$, in matlab use $H = (x * x').*(z * z')$

$$H = \begin{bmatrix} 37 & 61 & 70 & -17 & -43 & -34 \\ 61 & 101 & 114 & -25 & -67 & -50 \\ 70 & 114 & 137 & -42 & -94 & -84 \\ -17 & -25 & -42 & 29 & 47 & 58 \\ -43 & -67 & -94 & 47 & 85 & 94 \\ -34 & -50 & -84 & 58 & 94 & 116 \end{bmatrix}$$

SVM: Example using Matlab

- Matlab expects quadratic programming to be stated in the *canonical* (standard) form which is

$$\boxed{\begin{array}{rl} \text{minimize} & L_D(\alpha) = 0.5\alpha^t H\alpha + f^t\alpha \\ \text{constrained to} & A\alpha \le a \text{ and } B\alpha = b \end{array}}$$

  - where A, B, H are n by n matrices and f, a, b are vectors

- Need to convert our optimization problem to canonical form

$$\text{maximize} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}^t H \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

$$\text{constrained to} \quad \alpha_i \ge 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i z_i = 0$$

SVM: Example using Matlab

- Multiply by –1 to convert to minimization:

$$\text{minimize} \quad L_D(\alpha) = -\sum_{i=1}^{n} \alpha_i + \frac{1}{2}\alpha^t H \alpha$$

- Let $f = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix} = -\boldsymbol{ones}(6,1)$, then can write

$$\text{minimize} \quad L_D(\alpha) = f^t \alpha + \frac{1}{2}\alpha^t H \alpha$$

- First constraint is $\alpha_i \geq 0 \quad \forall i$
- Let $A = \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix} = \text{-}\boldsymbol{eye}(6)$, $a = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \boldsymbol{zeros}(6,1)$
- Rewrite the first constraint in canonical form:

$$A\alpha \leq a$$

SVM: Example using Matlab

- Our second constraint is $\sum_{i=1}^{n} \alpha_i z_i = 0$
- Let $B = \begin{bmatrix} z_1 & \cdots & \cdots & z_6 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} = [[z]; [zeros(5,6)]]$ → `[z.';zeros(5,6)]`

  and $b = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = zeros(6,1)$

- Second constraint in canonical form is:

$$B\alpha = b$$

- Thus our problem is in canonical form and can be solved by matlab:

$$\boxed{\begin{array}{rl} \text{minimize} & L_D(\alpha) = 0.5\alpha^t H\alpha + f^t\alpha \\ \text{constrained to} & A\alpha \le a \text{ and } B\alpha = b \end{array}}$$

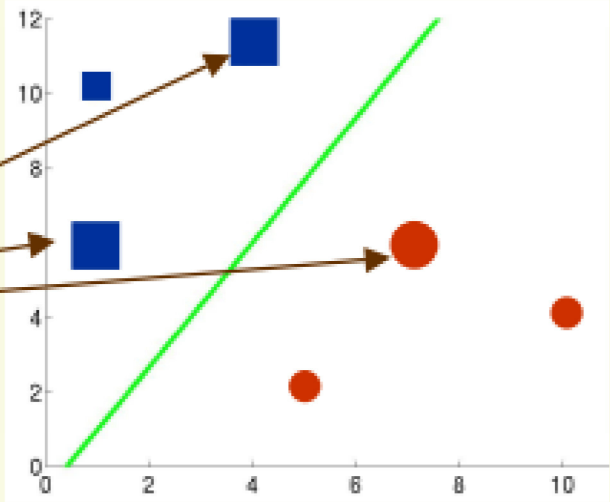
SVM: Example using Matlab

- α = quadprog($\boldsymbol{H}$+eye(6)*0.001, $\boldsymbol{f}$, $\boldsymbol{A}$, $\boldsymbol{a}$, $\boldsymbol{B}$, $\boldsymbol{b}$)

  for stability

- Solution $\alpha = \begin{bmatrix} 0.036 \\ 0 \\ 0.039 \\ 0 \\ 0.076 \\ 0 \end{bmatrix}$

  support vectors



- find $\boldsymbol{w}$ using $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i z_i \boldsymbol{x}_i = (\alpha .* z)^t \boldsymbol{x} = \begin{bmatrix} -0.33 \\ 0.20 \end{bmatrix}$

- since $\alpha_1 > 0$, can find $\boldsymbol{w_0}$ using

$$w_0 = \frac{1}{z_1} - w^t x_1 = 0.13$$

can solve for $\boldsymbol{w_0}$ using any $\alpha_i > 0$ and $\alpha_i \left[z_i \left(w^t x_i + w_0 \right) - 1 \right] = 0$

$$w_0 = \frac{1}{z_i} - w^t x_i$$