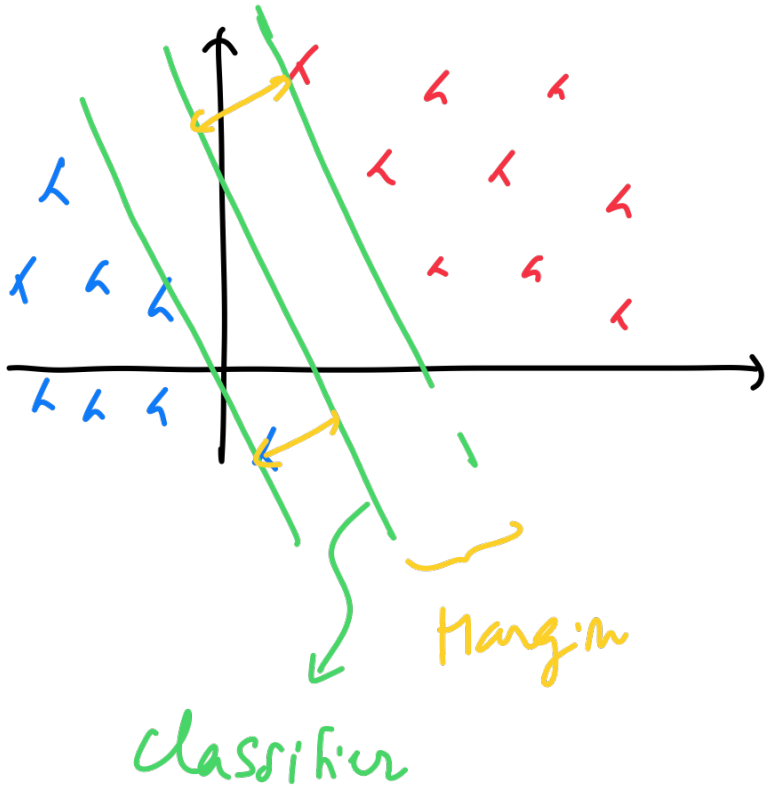
last Friday

Support vector Machines (Max Margin Classifiers)



→ First formulation

$$\beta^*, \beta_0^* = \underset{\beta, \beta_0}{\operatorname{argmax}} \min_i \frac{y(x^{(i)})\, t^{(i)}}{\|\beta\|}$$

$$= \underset{\beta, \beta_0}{\operatorname{argmax}} \min_i \frac{(\beta_0 + \beta^T x^{(i)})\, t^{(i)}}{\|\beta\|}$$

→ using the invariance of the objective w.r.t any scaling of β, β_0, we derived a more convenient constrained formulation

$$\min_{\beta, \beta_0} \|\beta\|^2 \quad \text{s.t} \quad y(x^{(i)}) t^{(i)} \geq 1 \quad \text{for every training pair } (x^{(i)}, t^{(i)}) \in T \qquad (*)$$

→ using KKT theorem and the Lagrangian, we were able to derive 2 sets of conditions on β, β_0

$$\beta = \sum_{i=1}^{N} \lambda_i t^{(i)} \underline{x}^{(i)} \qquad \text{(zero gradient of lagrangian w.r.t } \beta\text{)}$$

$$\sum_{i=1}^{N} \lambda_i t^{(i)} = 0 \qquad \text{(zero gradient of lagrangian w.r.t } \beta_0\text{)} \qquad (*)$$

$$\lambda_i \cdot (1 - y(x^{(i)}) t^{(i)}) = 0 \quad (*)$$ for every pair $(x^{(i)}, t^{(i)}) \in T$

Substituting (*) into the lagrangian,

$$\mathcal{L}(x, \lambda, t) = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^{N} \lambda_i \left(y(x^{(i)}) t^{(i)} - 1 \right)$$

and use the equivalence between (*) and the lagrangian formulation

$$\min_{\beta, \beta_0} \max_{\lambda} \mathcal{L}(x, \lambda, t)$$

we can get rid of β, β_0 and end up with a maximization problem on λ (quadratic optimization problem) known as the dual formulation

$$L(x, t, \lambda) = \frac{1}{2}\beta^T\beta - \sum_{i=1}^{N} \lambda_i \left((\beta_0 + \beta^T x^{(i)}) t^{(i)} - 1 \right)$$

$$= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j t^{(i)} t^{(j)} (x^{(i)T} x^{(j)})$$

$$- \sum_{i=1}^{N} \lambda_i t^{(i)} \beta_0 - \sum_{i=1}^{N} t^{(i)} \lambda_i \sum_{j=1}^{N} \lambda_j t^{(j)} (x^{(j)T} x^{(i)})$$

using $\beta = \sum_{i=1}^{N} \lambda_i t^{(i)} x^{(i)}$

$\sum_{i=1}^{N} t^{(i)} \lambda_i = 0$

$$0 \qquad + \sum_{i=1}^{N} \lambda_i$$

the dual formulation thus reduces to the maximization problem

$$\max_{\lambda} \; -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \lambda_i \lambda_j (x^{(i)T} x^{(j)}) + \sum \lambda_i$$

if we replace the original vectors $x^{(i)}$, $x^{(j)}$ with more general vectors $\phi(x^{(i)})$ $\phi(x^{(j)})$ we get

$$\max_{\lambda} \; -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \lambda_i \lambda_j \; \phi(x^{(i)})^T \phi(x^{(j)}) + \sum_{i=1}^{N} \lambda_i$$

on which we can apply the kernel trick, substituting $k(i,j)$ for $\phi(x^{(i)})^T \phi(x^{(j)})$. We then get the kernel based formulation

$$\max_{\lambda} \; -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t^{(i)} t^{(j)} \lambda_i \lambda_j \, k(i,j) + \sum_{i=1}^{N} \lambda_i$$

$$\lambda_i \geq 0 \quad \forall i$$

the final classifier $y(x) = \beta^T \phi(x) + \beta_0$

using (*) $$y(x) = \sum_{i=1}^{N} \lambda_i t^{(i)} \phi(x^{(i)})^T \phi(x) + \beta_0$$

$$= \sum_{i=1}^{N} \lambda_i t^{(i)} k(x^{(i)}, x) + \beta_0 \quad (*)$$

→ the Maximum Margin classifier can be learned with a kernel (instead of feature vectors) and has a final expression which consists of a superposition of the kernel function on each of the training example.

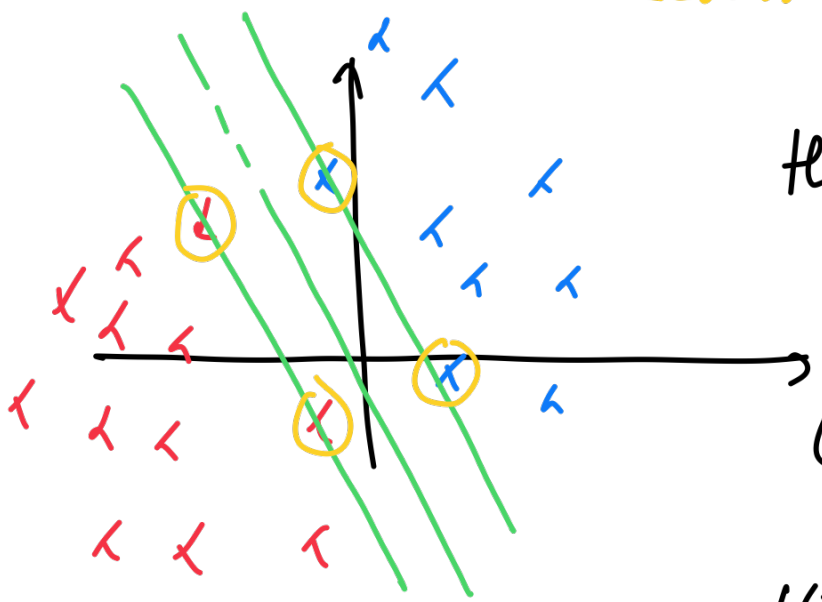
Moreover, if we use (*) we see that for every example from $\{x^{(i)}, t^{(i)}\}_{i=1}^{N}$ either $\lambda_i = 0$ or the constraint is tight: i.e

$$(t^{(i)} y(x^{(i)}) - 1) = 0$$

But a tight constraint also means that the $x^{(i)}$ is minimizing its distance to the plane (or exactly on the margins)

→ As a result, the only training pairs that will show up in the final expression of our classifier are those that are lying on the margins (also known as the support vectors). Using the complementary slackness condition (*) we can simplify the final expression of our classifier (*) as

$$y(x) = \sum_{i \in \text{support vectors}} t^{(i)} \lambda_i K(x^{(i)}, x) + \beta_0$$



to get an expression for β_0 that only depends on the support vectors

we go back to the constraints (which are tight at the support vectors)

$$t^{(i)}(\beta_0 + \beta^T \phi(x^{(i)})) = 1 \quad \text{for every } i \in \text{support vectors}$$

multiplying by $t^{(i)}$ we get (for each support vector)

$$(t^{(i)})^2 (\beta_0 + \sum_{j \in \text{support vector}} t^{(j)} \lambda^{(j)} \phi(x^{(j)})^T \phi(x^{(i)})) = t^{(i)}$$

$(t^{(i)})^2 = 1$ as $t^{(i)} \in \{\pm 1\}$

To get a more robust estimate of β_0, we can use sum the constraints over the support vectors

$$N\beta_0 + \sum_{i \in \text{support vectors}} \left(\sum_{j \in \text{support vectors}} t^{(j)} \lambda^{(j)} K(j, i) \right) = \sum_{i \in \text{support vector}} t^{(i)}$$

$$\beta_0 = \frac{1}{N} \left(\sum_i t^{(i)} - \sum_i \sum_j t^{(j)} \lambda^{(j)} K(x^{(j)}, x^{(i)}) \right) \quad (**)$$

using (*) and (**) we get a final expression for our max margin classifier only depends on the support vectors

For this reason, the max margin classifier is also known as

Sparse vector Machine

Sparse kernel Machine

Support vector Machine

Max Margin classifier