Support Vector Machines

Intelligent Systems and Control

2019

Sepehr Maleki

University of Lincoln
School of Engineering

Lagrange Multipliers

Objective: Minimise/Maximise a multivariable function $f(x, y, ...)$ subject to $g(x, y, ...) = c$.

I. Introduce a new variable α (Lagrange multiplier) and define the Lagrangian function $\mathcal{L}$ as follows:

$$\mathcal{L}(x, y, ..., \alpha) = f(x, y, ...) - \alpha\Big(g(x, y, ...) - c\Big)$$

II. Set the gradient of $\mathcal{L}$ equal to the zero vector to find its "critical points":

$$\nabla\mathcal{L}(x, y, ..., \alpha) = \mathbf{0}$$

III. Plug each solution (x_0, y_0, α_0) into f and whichever one gives the greatest (or smallest) value is the maximum (or minimum).

Lagrange Multipliers – Example

Maximise $f(x, y) = x^2 y$ subject to $x^2 + y^2 = 1$.

- $\nabla g(x, y) = \nabla(x^2 + y^2) = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$.

- $\nabla f(x, y) = \nabla(x^2 y) = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix}$.

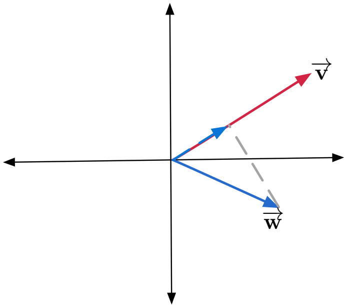
- $\begin{bmatrix} 2xy \\ x^2 \end{bmatrix} = \alpha \begin{bmatrix} 2x \\ 2y \end{bmatrix} \Longrightarrow \begin{cases} 2xy = \alpha 2x \longrightarrow y = \alpha \\ x^2 = \alpha 2y \longrightarrow x^2 = 2y^2 \\ \color{red}{x^2 + y^2 = 1} \longrightarrow 3y^2 = 1 \end{cases}$.

from which we can find four solutions:

$$(\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}}), (-\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}}), (\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}}), (-\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}}).$$

Dot Product

Dot product is an algebraic operation that takes two equal-length vectors and returns a single number.



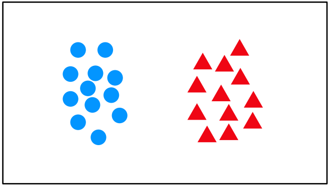
$$\underbrace{\begin{bmatrix} 4 \\ 1 \end{bmatrix}}_{\vec{\mathbf{v}}} \cdot \underbrace{\begin{bmatrix} 2 \\ -1 \end{bmatrix}}_{\vec{\mathbf{w}}} = (\text{Length of projected } \vec{\mathbf{w}})\,(\text{ Length of } \vec{\mathbf{v}})$$

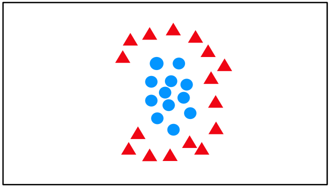
Binary Classification

Given training data $(\mathbf{x}_i, y_i)$, with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

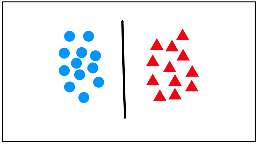
i.e., $y_i f(\mathbf{x}_i) > 0$ for a correct classification.

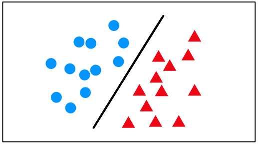




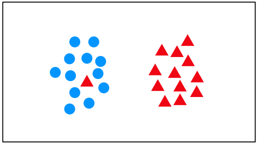
Linear Separability

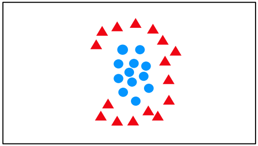
Linearly Separable



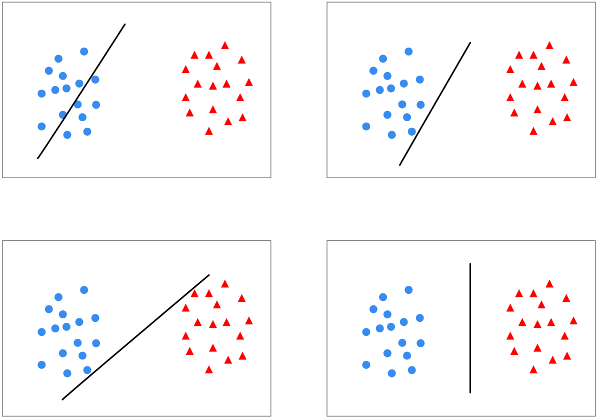


Not Linearly Separable





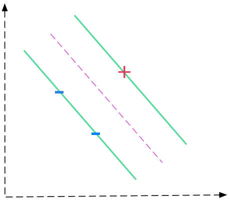
What is The Best Separation?

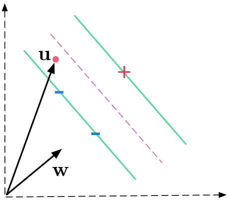


Maximum margin solution: Most stable under perturbations of inputs

Maximum Margin Separation

Let $\mathbf{w}$ be a vector (of any length, for now) perpendicular to the margins. Moreover, let $\mathbf{u}$ be a vector that points to an unknown. We want to find the correct class (i.e., $+$ or $-$) for the unknown point.



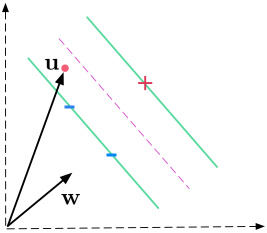


If the projection of $\mathbf{u}$ onto $\mathbf{w}$ (dot product) is greater than some constant C, we can decide that the unknown is $+$.

We can formulate this as the following decision rule:

$$\mathbf{w}^{\top}.\,\mathbf{u} + b \geq 0 \Longrightarrow {\color{red}+}$$

Decision Rule



$$\star \quad \boxed{\begin{array}{l} \mathbf{w}^\top . \mathbf{x}_+ + b \geq 1 \\ \mathbf{w}^\top . \mathbf{x}_- + b \leq -1 \end{array}}$$

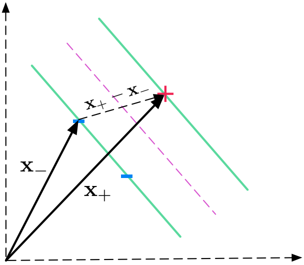
Let $y_i = \begin{cases} +1 & \text{for + samples} \\ -1 & \text{for - samples} \end{cases}$. Then multiplying both sides of $\star$ by y_i gives:

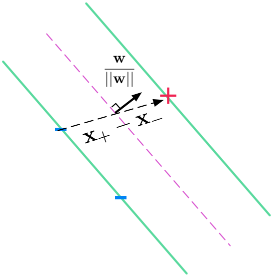
$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \ .$$

Moreover, where $\mathbf{x}_i$ is exactly on the margins, we require:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 = 0 \ .$$

Width of The Margin





$$\text{Width} = \frac{\mathbf{w}^\top}{||\mathbf{w}||} \cdot (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2}{||\mathbf{w}||} \; .$$

Our objective is o have the maximum possible width:

$$\max \frac{2}{||\mathbf{w}||} \longrightarrow \max \frac{1}{||\mathbf{w}||} \longrightarrow \min ||\mathbf{w}|| \longrightarrow \min \frac{1}{2}||\mathbf{w}||^2 \; .$$

Width of The Margin

Our objective is:

$$\min \frac{1}{2}||\mathbf{w}||^2 \;\text{ such that }\; y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1 \ .$$

We use the Lagrangian optimisation:

$$\mathcal{L} = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i \Big[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \Big]$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \ .$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \Longrightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \ .$$

Width of The Margin

We start with:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \ , \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \ ,$$

$$\mathcal{L} = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right] \ .$$

Then the Lagrangian function can be written:

$$\mathcal{L} = \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i\right)\left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j\right) - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j\right)$$

$$- \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \ \mathbf{x}_i^\top \mathbf{x}_j \ .$$

New Decision Rule

Recall the decision rule:

$$\mathbf{w}^\top . \, \mathbf{u} + b \geq 0 \Longrightarrow + \, .$$

Now that we know:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \; ,$$

we can re-write the decision rule for the unknown vector $\mathbf{u}$:

$$\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \mathbf{u} + b \geq 0 \Longrightarrow + \, .$$

So Far ...

N is number of training points, and d is dimension of feature vector $\mathbf{x}$.

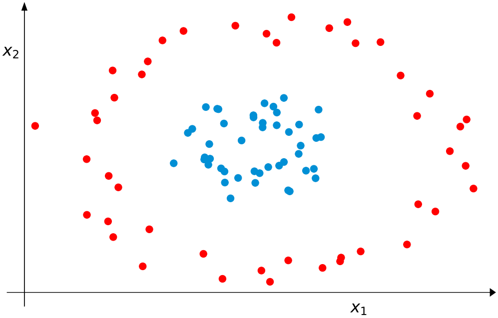
The SVM classifier is given by:

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i (\mathbf{x}_i^\top \mathbf{x}) + b \ ,$$

which is learned by the following optimisation problem over α_i:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \ \mathbf{x}_i^\top \mathbf{x}_j \ .$$

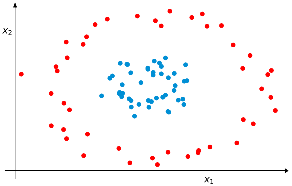
Non-Linear Decision Boundary

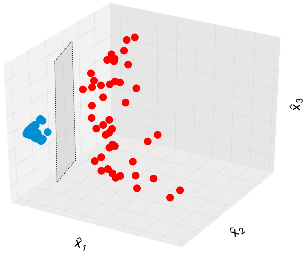


Linear classifier?

Non-Linear Decision Boundary

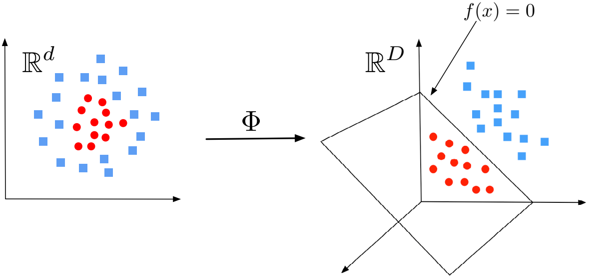
$$\Phi : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \to \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{bmatrix} \qquad \mathbb{R}^2 \to \mathbb{R}^3$$





Data is linearly separable when mapped to higher dimensions.

SVM Classifiers in a Transformed Feature Space



$$\Phi : \mathbf{x} \rightarrow \Phi(\mathbf{x}) \qquad \mathbb{R}^d \rightarrow \mathbb{R}^D$$

Learn a classifier:

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i \, \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}) + b \, ,$$

where $\Phi(\mathbf{x})$ is a feature map.

Kernel Functions

Define $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$ as a *kernel*, then:

Classifier:

$$f(\mathbf{x}_j) = \sum_i^N \alpha_i y_i \, k(\mathbf{x}_i, \mathbf{x}_j) + b$$

Learning:

$$\max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \, k(\mathbf{x}_i^\top \mathbf{x}_j) \; .$$

The Kernel Trick

- Classifier can be learnt and applied without explicitly computing $\Phi(\mathbf{x})$.
- All that is required is the kernel $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^2$
- Complexity of learning depends on N not on D.

Example Kernels:

- Linear: $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$.
- Polynomial: $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^d \;\; \forall d > 0$
- Gaussian:

$$k(\mathbf{x}, \mathbf{x}') = e^{\left(\dfrac{-||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)} \;\; \forall \sigma > 0$$