Ciência de Dados Quântica 2021/22

Kernel Based Methods and Feature Maps

LUÍS PAULO SANTOS

Material de Consulta

- [Schuld2021] – Secs. 2.5.4, 3.6.1; Chap. 6
- "Support Vector Machines: All you need to know!"
  https://youtu.be/ny1iZ5A8ilA
- "The Kernel Trick in Support Vector Machine (SVM)"
  https://youtu.be/Q7vT0--5VII
- "Supervised learning with quantum enhanced feature spaces"
  Vojtech Havlicek et al.
  https://arxiv.org/pdf/1804.11326.pdf
- Qiskit Global Summer School 2021: Lectures 6.1, 6.2
  https://www.youtube.com/watch?v=xgA4Dx_7q34&list=PLOFEBzvs-VvqJwybFxkTiDzhf5E11p8BI

Kernel Methods: concept

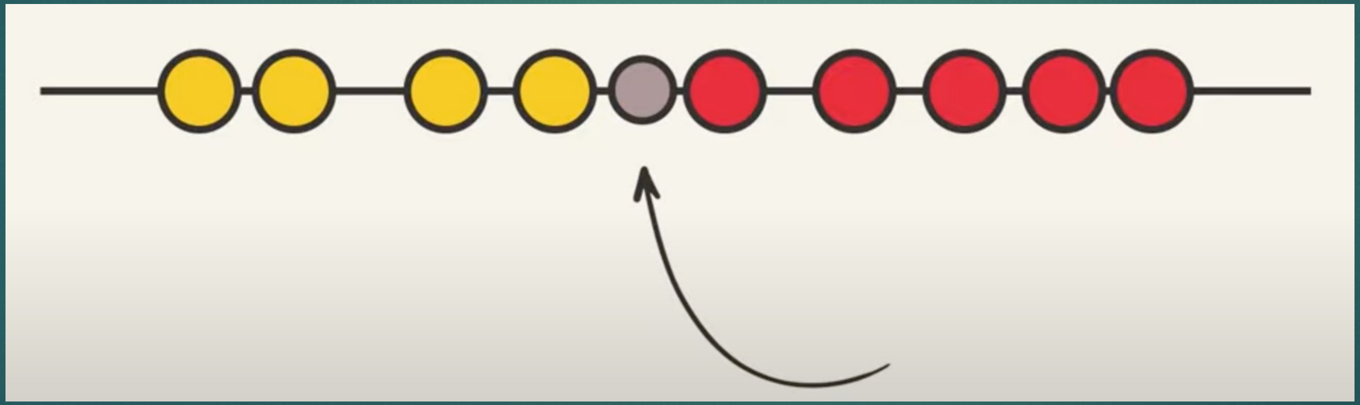
- Kernel methods are based on a **similarity measure** between data points
- **Definition:** for a data domain χ a kernel is a positive semi-definite bivariate function $\varkappa : \chi \times \chi \longrightarrow \mathbb{R}$
  - positive semi-definite means:
    - $\varkappa(x, x') \geq 0$
    - $\varkappa(x, x') = \varkappa(x', x)^*$

Support Vector Machines (SVM)

- Family of kernel based methods for classification
  - we focus on binary classification, but multi-class (SVC) and regression(SVR) also possible
- Supervised method which finds an hyperplane separating the classes
  - training input (M labelled data points): $\mathcal{D} = \{(\boldsymbol{x}^1, y^1), \cdots, (\boldsymbol{x}^M, y^M)\}$
  - each $\boldsymbol{x}^i \in \mathbb{R}^N$, $\boldsymbol{x}^i = \left(x_1^i, \cdots, x_N^i\right)^T$ for N features
  - each $y^1 \in \{-1,1\}$ for binary classification
- The training stage finds the class separating hyperplane
- The classification stage classifies a previously unseen unlabelled data point $\boldsymbol{u} \in \mathbb{R}^N$ as $y^u \in \{-1,1\}$

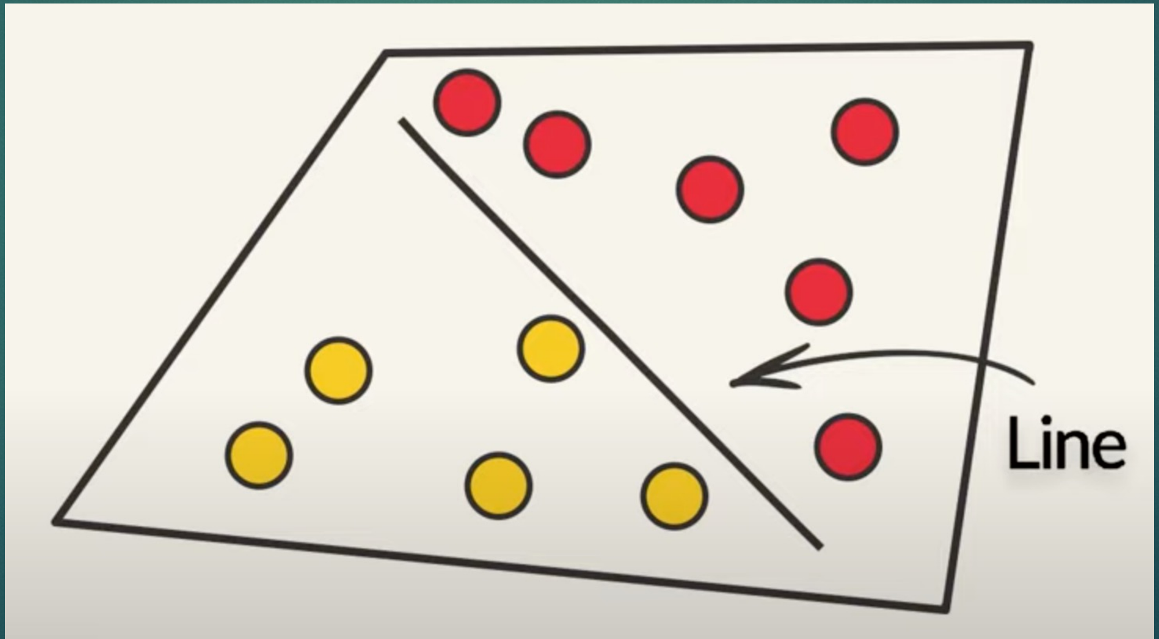
Support Vector Machines (SVM)

- The hyperplane is defined by a vector $\boldsymbol{w} \in \mathbb{R}^N$ with the same dimensionality N as the data points
- 1D – The hyperplane is a point



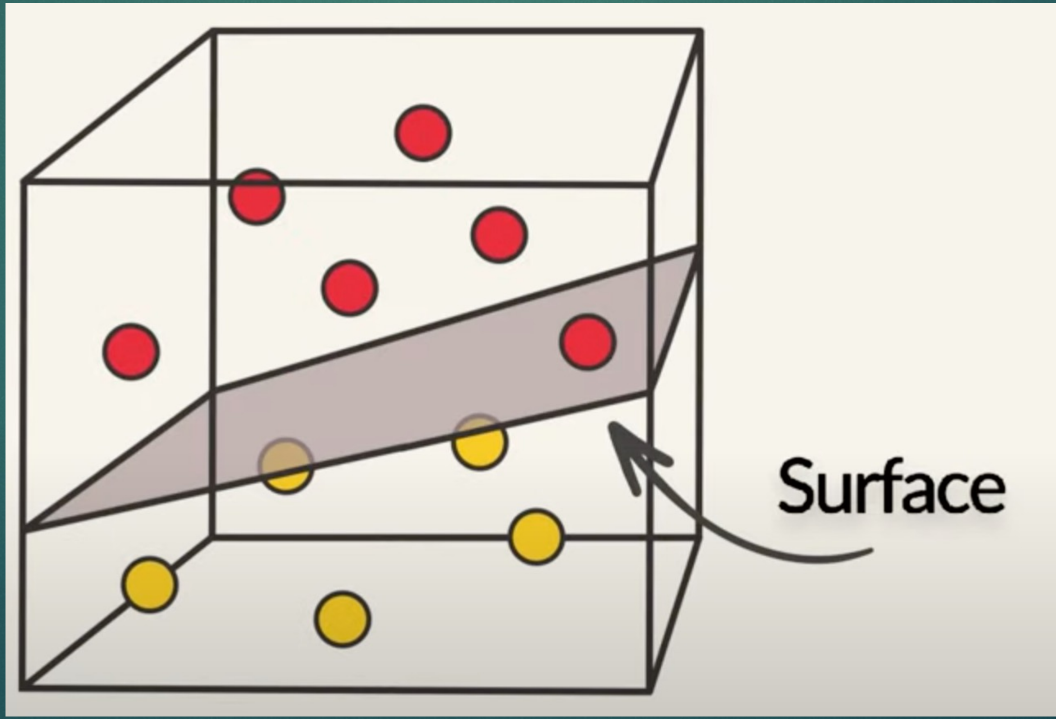
Support Vector Machines (SVM)

- The hyperplane is defined by a vector $\boldsymbol{w} \in \mathbb{R}^N$ with the same dimensionality N as the data points
- 2D – The hyperplane is a line



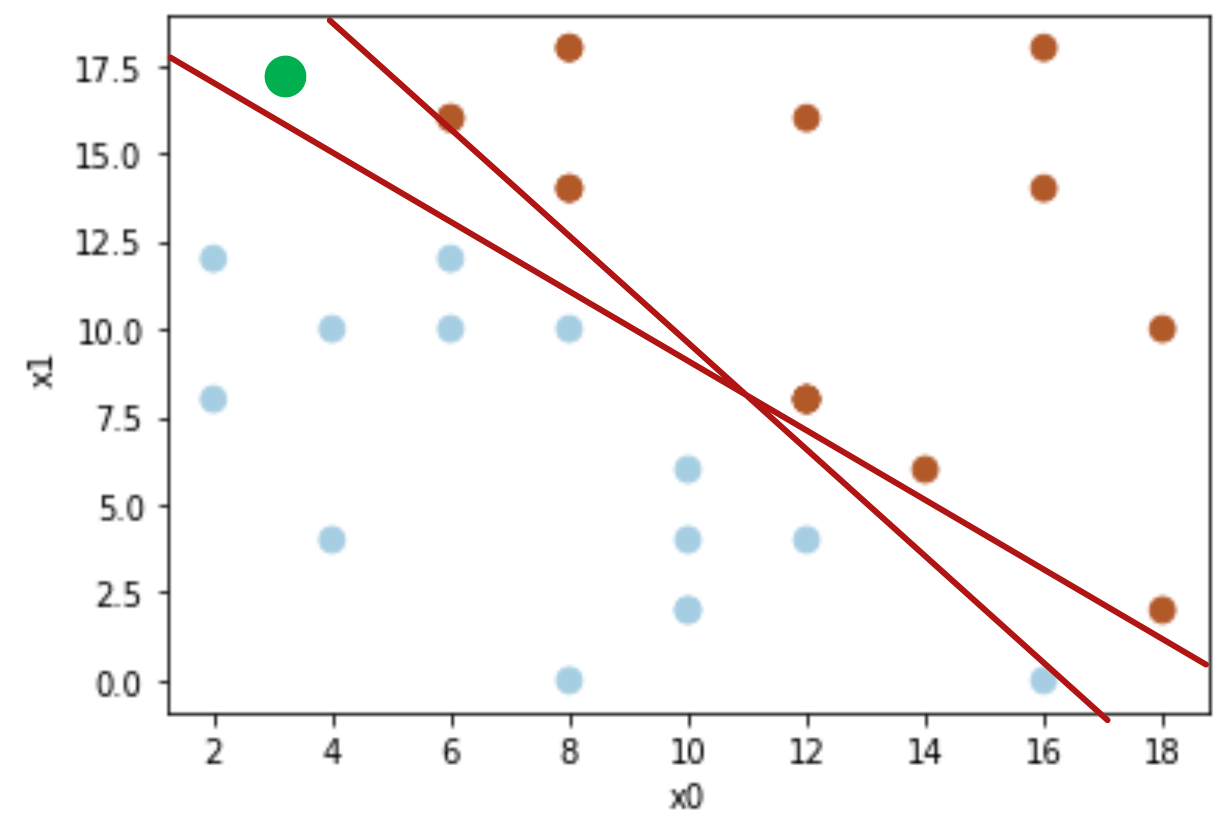
Support Vector Machines (SVM)

- The hyperplane is defined by a vector $\boldsymbol{w} \in \mathbb{R}^N$ with the same dimensionality N as the data points
- 3D – The hyperplane is a plane



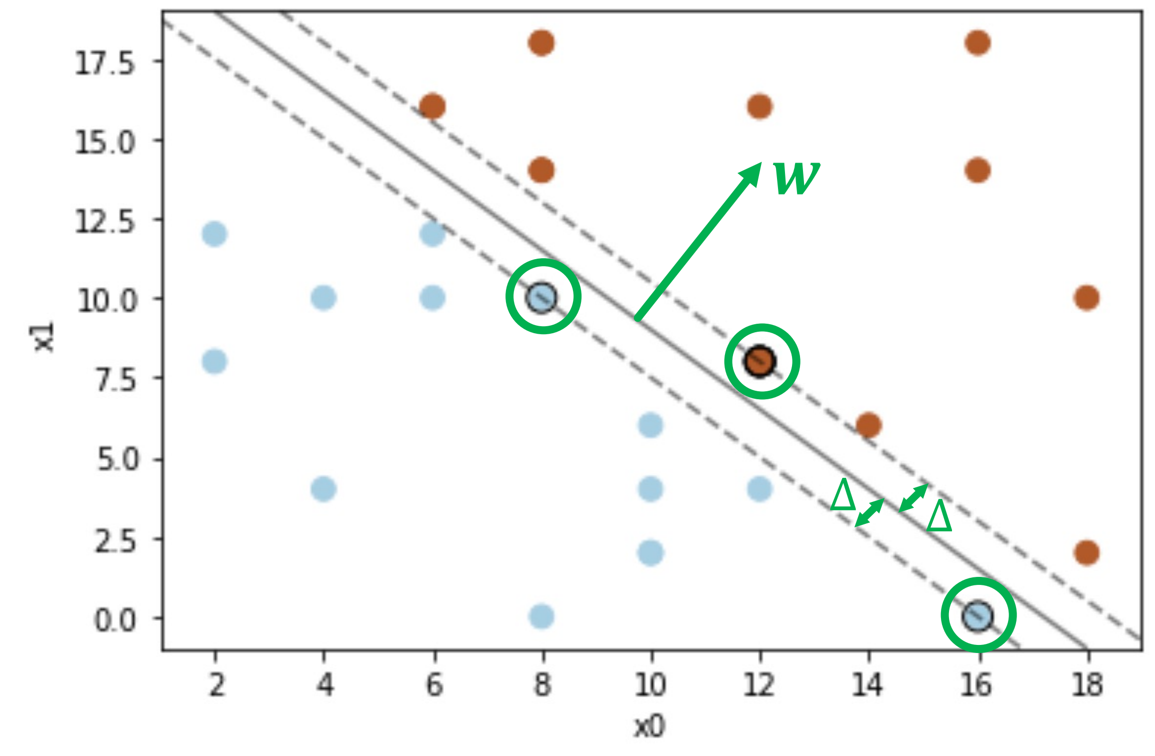
Support Vector Machines (SVM)

- There are infinite hyperplanes separating the two classes. Which hyperplane to select?



Support Vector Machines (SVM)

- Select the hyperplane which maximizes the "margin", i.e, the distance to the nearest points on each class.
- These are the support vectors

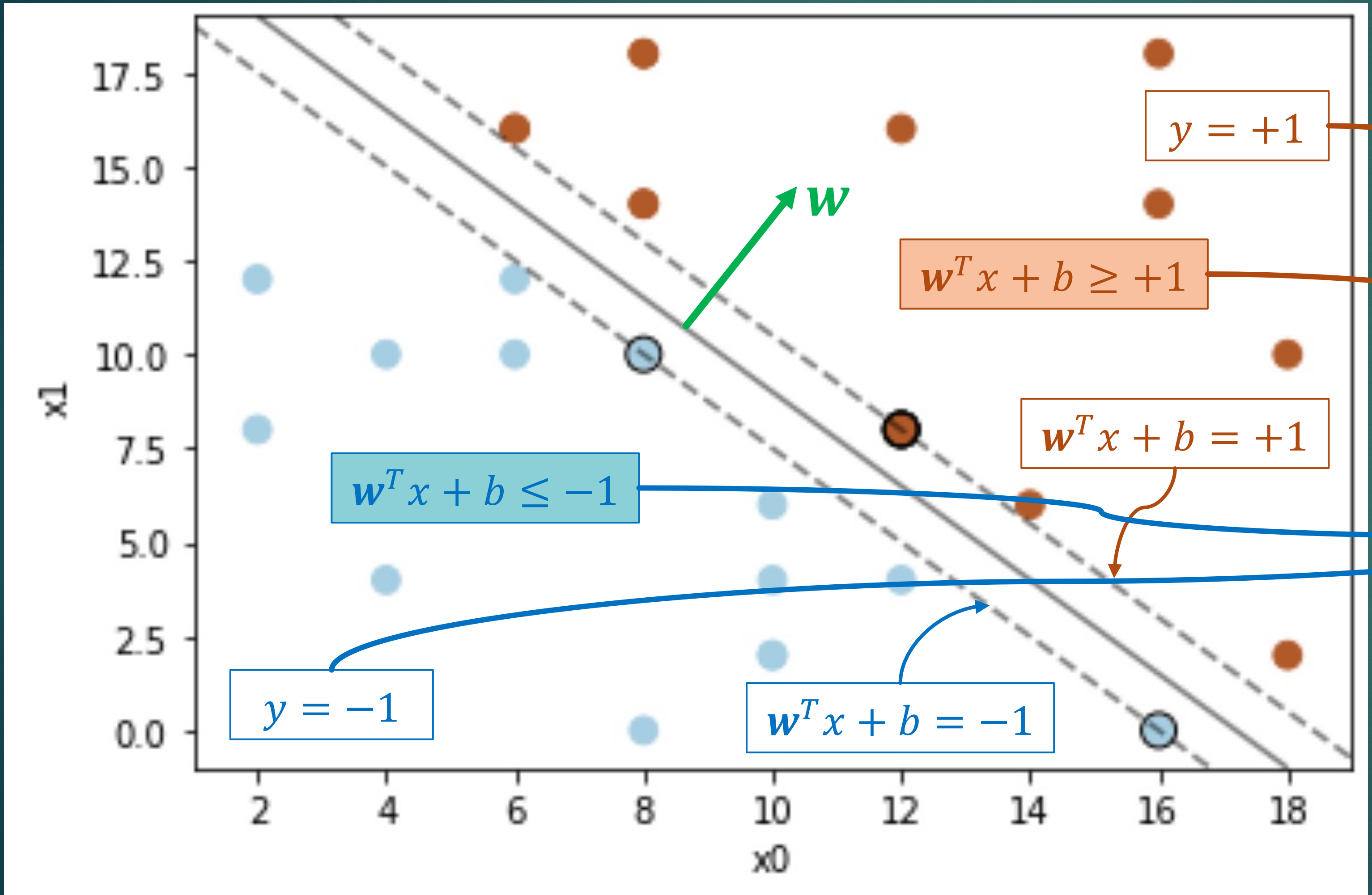


- The vector $\boldsymbol{w} \in \mathbb{R}^N$ is perpendicular to the hyperplane
- The points in the hyperplane obey

$$f_w(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = 0$$

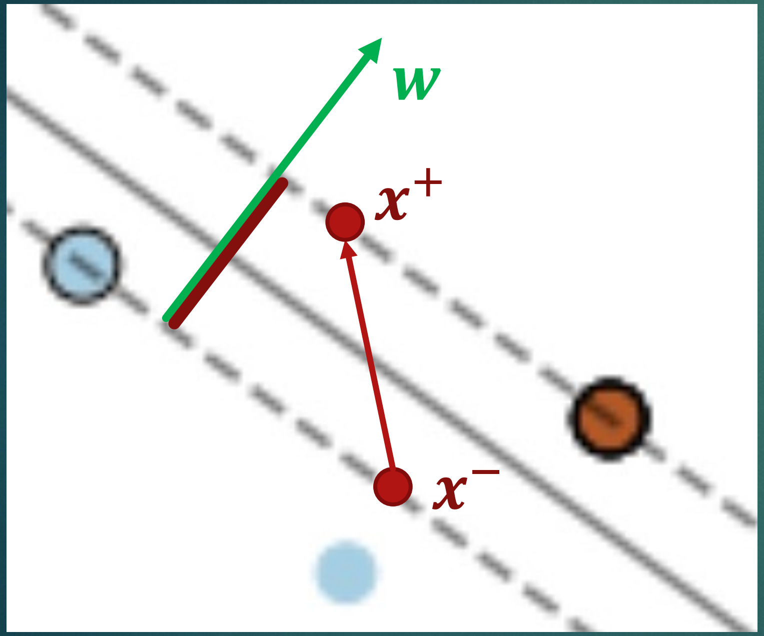
- The training stage optimizes over $\boldsymbol{w}$ and b, maximizing the margin Δ

Support Vector Machines (SVM)



$$y(\boldsymbol{w}^T x + b) \geq 1$$

Support Vector Machines (SVM)



- $\max_{\boldsymbol{w},b} \left((\boldsymbol{x}^+ - \boldsymbol{x}^-) \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \right) \; s.t. \; y^m(\boldsymbol{w}^T \boldsymbol{x}^m + b) \geq 1$
- $\max_{\boldsymbol{w},b} \left(\frac{\boldsymbol{x}^+ \boldsymbol{w} - \boldsymbol{x}^- \boldsymbol{w}}{\|\boldsymbol{w}\|} \right) \; s.t. \; y^m(\boldsymbol{w}^T \boldsymbol{x}^m + b) \geq 1$
- $\max_{\boldsymbol{w},b} \left(\frac{(1-b)-(-1-b)}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|} \right) \; s.t. \; y^m(\boldsymbol{w}^T \boldsymbol{x}^m + b) \geq 1$
- $\min_{\boldsymbol{w},b} \left(\frac{1}{2} \|\boldsymbol{w}\|^2 \right) \; s.t. \; y^m(\boldsymbol{w}^T \boldsymbol{x}^m + b) \geq 1$

Support Vector Machines (SVM)

- $\min_{\boldsymbol{w},b}\left(\frac{1}{2}\|\boldsymbol{w}\|^2\right)\ s.t.\ y^m(\boldsymbol{w}^T\boldsymbol{x}^m+b)\geq 1$
- $\min_{\boldsymbol{w},b,\boldsymbol{\alpha}}\left[\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})=\frac{1}{2}\boldsymbol{w}^2-\sum_{\boldsymbol{m}=\boldsymbol{1}}^{\boldsymbol{M}}(\alpha^m y^m(\boldsymbol{w}^T\boldsymbol{x}^m+b)-1)\right]$
- $\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}=\boldsymbol{w}-\sum_{\boldsymbol{m}=\boldsymbol{1}}^{\boldsymbol{M}}(\alpha^m y^m\boldsymbol{x}^m)=\boldsymbol{0}\Leftrightarrow\boldsymbol{w}=\sum_{\boldsymbol{m}=\boldsymbol{1}}^{\boldsymbol{M}}(\alpha^m y^m\boldsymbol{x}^m)$
- $\frac{\partial\mathcal{L}}{\partial b}=\sum_{\boldsymbol{m}=\boldsymbol{1}}^{\boldsymbol{M}}(\alpha^m y^m)=\boldsymbol{0}$
- $\min_{\boldsymbol{\alpha}}\left(\mathcal{L}(\boldsymbol{\alpha})=\sum_{m=1}^{M}\alpha^m-\frac{1}{2}\sum_{\boldsymbol{m},\boldsymbol{m}'=\boldsymbol{1}}^{\boldsymbol{M}}\alpha^m\alpha^{m'}y^m y^{m'}\boldsymbol{x}^{m\boldsymbol{T}}\boldsymbol{x}^{m'}\right)$

$$s.t.\sum_{m=1}^{M}\alpha^m y^m=0$$

$$\underbrace{\boldsymbol{x}^{m\boldsymbol{T}}\boldsymbol{x}^{m'}}_{\langle\boldsymbol{x}^m,\boldsymbol{x}^{m'}\rangle}$$

Depends on $\boldsymbol{w}$, thus $\mathcal{O}(NM)$

Doesn't depend on $\boldsymbol{w}$, thus $\mathcal{O}(M)$

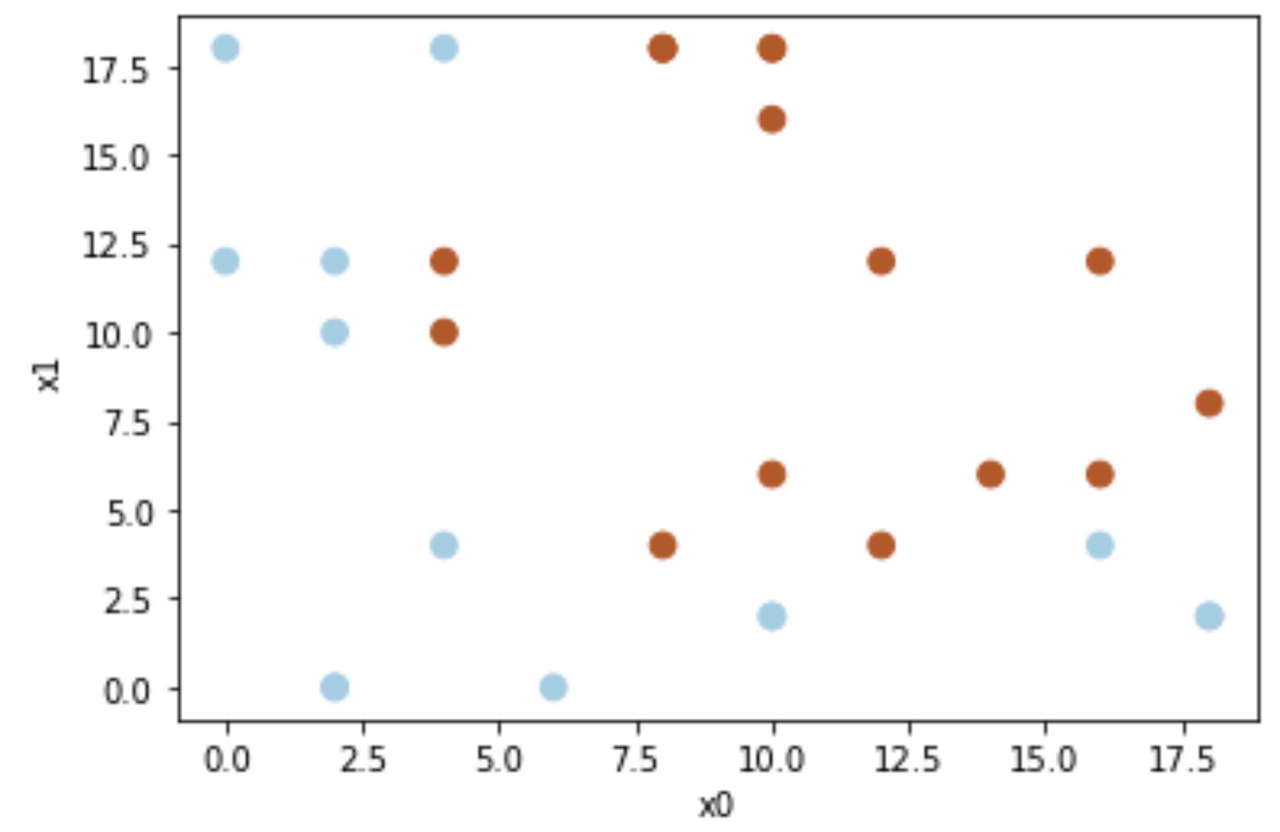
Support Vector Machines (SVM)

- $\min_{\boldsymbol{\alpha}} \left(\mathcal{L}(\boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha^m - \frac{1}{2} \sum_{\boldsymbol{m},\boldsymbol{m}'=\mathbf{1}}^{\boldsymbol{M}} \alpha^m \alpha^{m'} y^m y^{m'} \langle \boldsymbol{x}^m, \boldsymbol{x}^{m'} \rangle \right) \; s.t. \; \sum_{m=1}^{M} \alpha^m y^m = 0$
- Let $\varkappa(\boldsymbol{x}^m, \boldsymbol{x}^{m'}) = \langle \boldsymbol{x}^m, \boldsymbol{x}^{m'} \rangle$ then:
  - $\min_{\boldsymbol{\alpha}} \left(\mathcal{L}(\boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha^m - \frac{1}{2} \sum_{\boldsymbol{m},\boldsymbol{m}'=\mathbf{1}}^{\boldsymbol{M}} \alpha^m \alpha^{m'} y^m y^{m'} \varkappa(\boldsymbol{x}^m, \boldsymbol{x}^{m'}) \right) \; s.t. \; \sum_{m=1}^{M} \alpha^m y^m = 0$
- $\boldsymbol{w} = \sum_{\boldsymbol{m}=\mathbf{1}}^{\boldsymbol{M}} (\alpha^m y^m \boldsymbol{x}^m)$
  - $\alpha^m \neq 0$ only for the support vectors, Let S be such set $S = \{s = \alpha^m \mid \alpha^m \neq 0, m = 1 \cdots M\}$
- Classification of an unseen point $\boldsymbol{u} \in \mathbb{R}^N$:

$$f_{\boldsymbol{w}}(\boldsymbol{u}) = sign\left[\left(\sum_{\boldsymbol{s} \epsilon \boldsymbol{S}} (\alpha^s y^s \boldsymbol{x}^s) \right)^T \boldsymbol{u} + b \right] = sign[\varkappa(\boldsymbol{w}, \boldsymbol{u}) + b]$$

Feature Maps

- What if the data is not linearly separable on its original domain?



Feature Maps

- A feature map $\phi(\boldsymbol{x})\colon \mathbb{R}^N \longrightarrow \mathbb{R}^F$ is a non-linear transformation of the data, which increases its dimensionality ($F > N$)
- The goal is to reach linearly separability in the higher dimensional feature space
- The SVM training equations become:
  - $\min_{\boldsymbol{\alpha}} \left(\mathcal{L}(\boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha^m - \frac{1}{2} \sum_{\boldsymbol{m},\boldsymbol{m}'=\boldsymbol{1}}^{\boldsymbol{M}} \alpha^m \alpha^{m'} y^m y^{m'} \langle \phi(\boldsymbol{x}^m), \phi\left(\boldsymbol{x}^{m'}\right) \rangle \right) \;\; s.t.\ \sum_{m=1}^{M} \alpha^m y^m = 0$
  - $\boldsymbol{w} = \sum_{\boldsymbol{s} \in \boldsymbol{S}} \left(\alpha^s y^s \phi(\boldsymbol{x}^m) \right)$
- Classification of an unseen point $\boldsymbol{u} \in \mathbb{R}^N$:

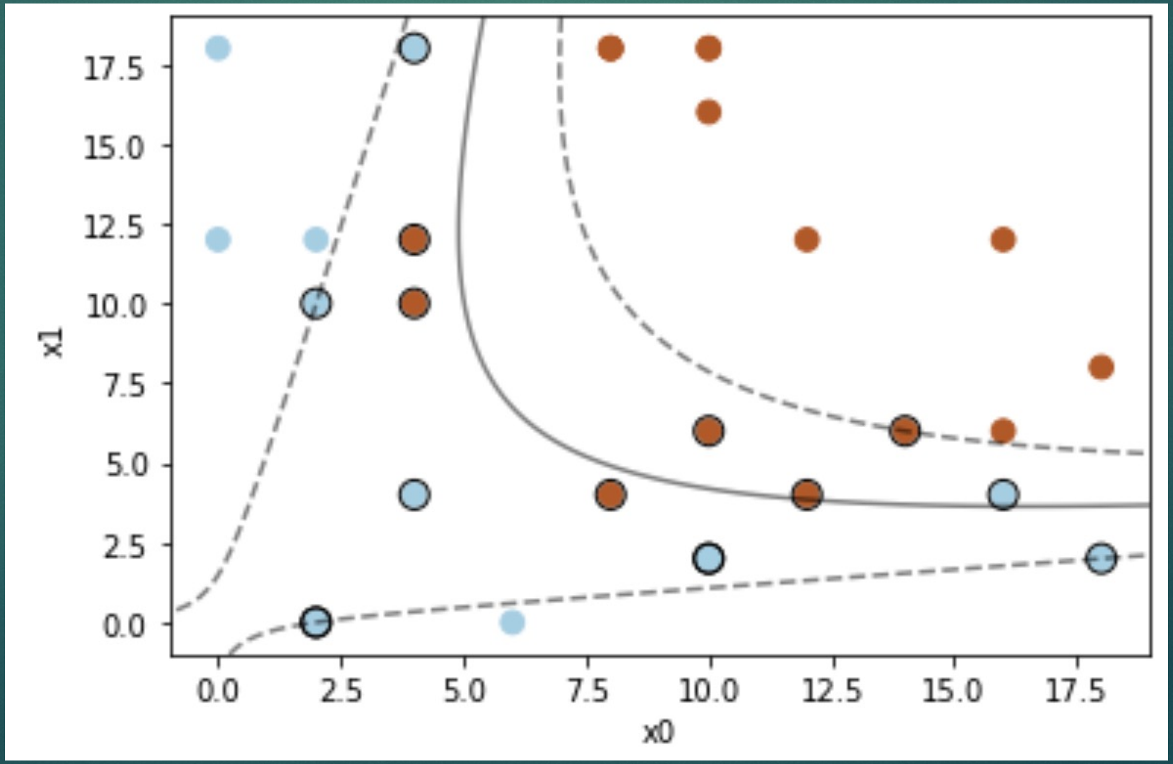
$$f_{\boldsymbol{w}}(\boldsymbol{u}) = sign\left[\left(\sum_{\boldsymbol{s} \epsilon \boldsymbol{S}} \left(\alpha^s y^s \phi(\boldsymbol{x}^s) \right) \right)^T \phi(\boldsymbol{u}) + b \right] = sign[\langle \phi(\boldsymbol{w}), \phi(\boldsymbol{u}) \rangle + b]$$

Feature Maps

- Polynomial feature map (N=2, degree=2, F=6)

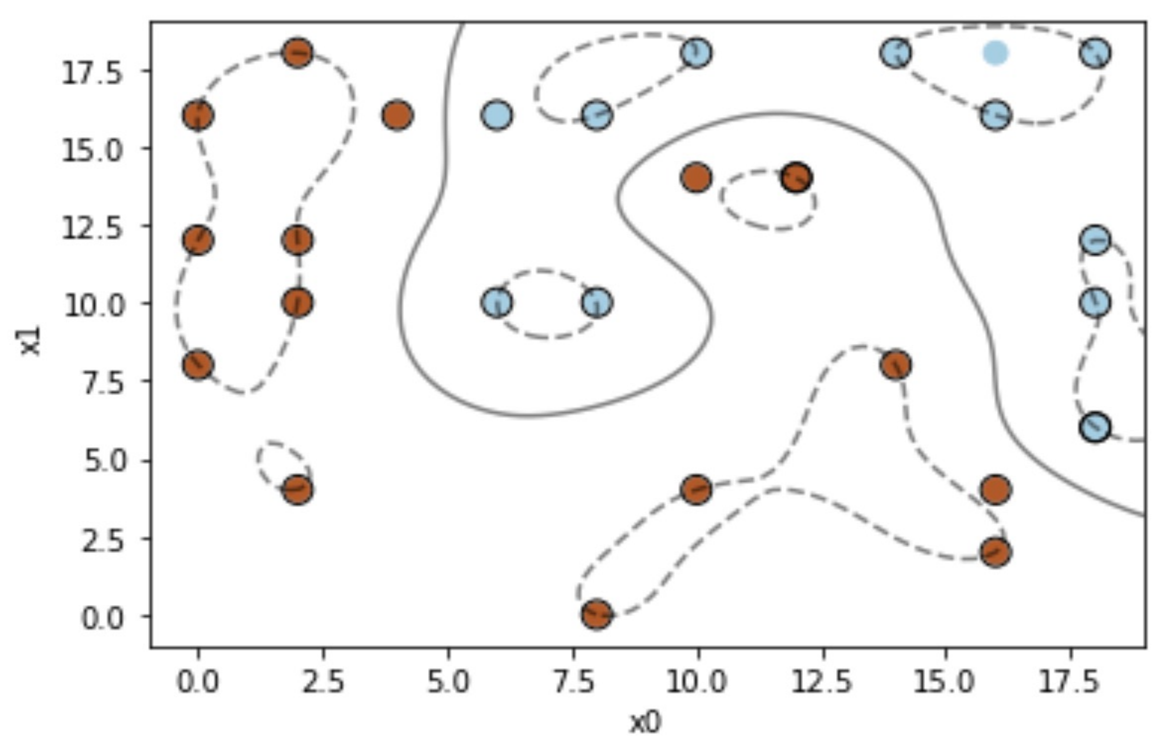
$$\phi(\boldsymbol{x}) : (x_0, x_1) \longrightarrow \left(1, \sqrt{2}x_0, \sqrt{2}x_1, \sqrt{2}x_0x_1, x_0^2, x_1^2\right)$$

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle = 1 + 2\, x_0 z_0 + 2\, x_1 z_1 + 2 x_0 z_0 x_1 z_1 + x_0^2 z_0^2 + x_1^2 z_1^2$$



Feature Maps

- The dimensionality of the feature space can be very large (even infinite) compromising computational efficiency



polynomial feature map
degree=4
degree=7

RBF feature map

Feature Maps and "Kernel Trick"

- Certain feature maps can be captured analytically by a kernel, without ever computing or representing $\phi(\boldsymbol{x})$
- Polynomial kernel of degree d:

$$\langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\rangle = \varkappa(\boldsymbol{x},\boldsymbol{z}) = (1+\langle\boldsymbol{x},\boldsymbol{z}\rangle)^{\boldsymbol{d}}$$

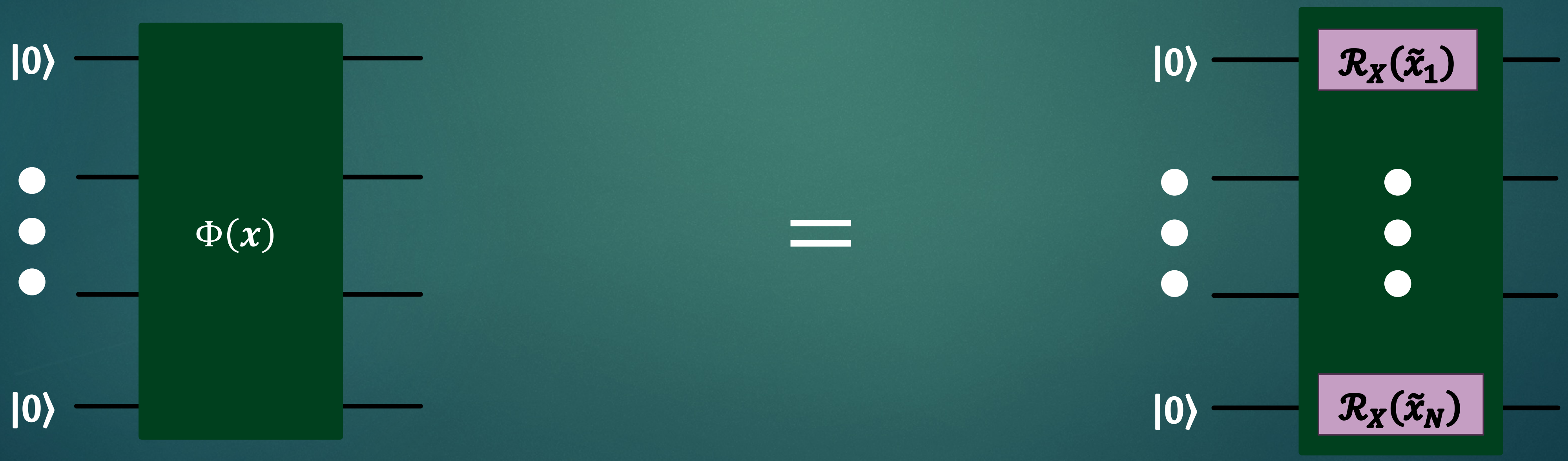
- Radial Basis Function kernel (infinitely dimensional):

$$\langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\rangle = \varkappa(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{e}^{-\gamma\|\boldsymbol{x}-\boldsymbol{z}\|^2}$$

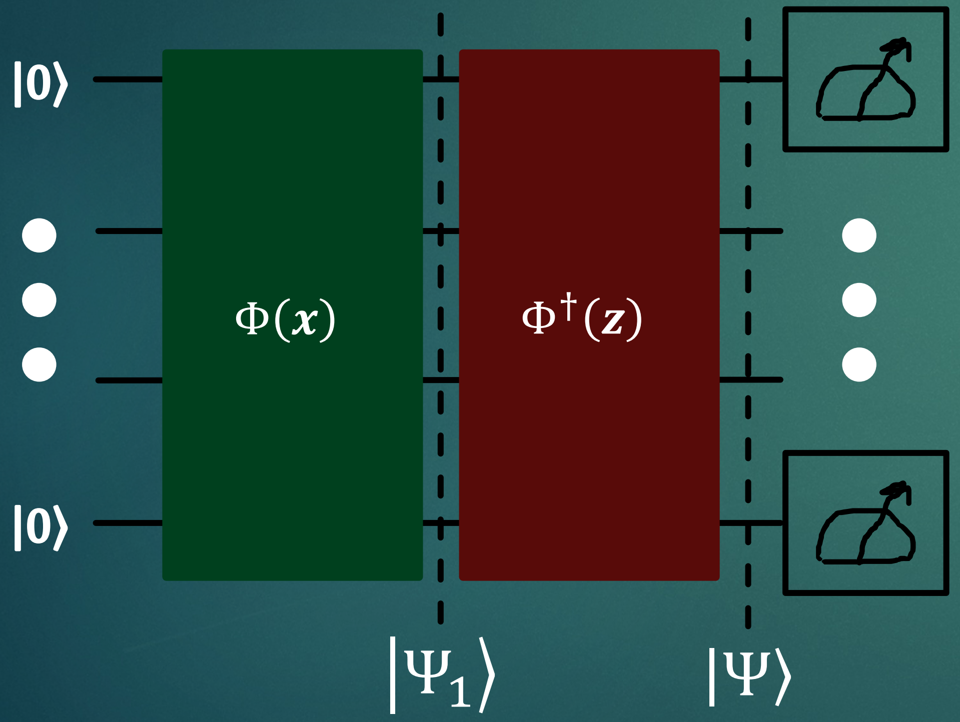
- The "kernel trick" allows for memory and computation efficient implementations of high dimensional feature maps

Quantum Feature Map

- The quantum feature map, operator $\Phi(\boldsymbol{x})$ might be seen as the encoding of data into a quantum state.
- Example: angle (or rotation) encoding $\tilde{x}_i = \frac{x_i}{\max\limits_{x_j}(abs(x))} * \pi$



Quantum $\varkappa(\boldsymbol{x}, \boldsymbol{z})$



- $|\Psi_1\rangle = \Phi(\boldsymbol{x})|0\rangle$
- $|\Psi\rangle = \Phi^\dagger(\boldsymbol{z})\Phi(\boldsymbol{x})|0\rangle$
- $P(|0\rangle) = \langle 0|\Phi^\dagger(\boldsymbol{z})\Phi(\boldsymbol{x})|0\rangle = |\langle\Phi(\boldsymbol{x})|\Phi(\boldsymbol{z})\rangle|^2$
- $|\langle\Phi(\boldsymbol{x})|\Phi(\boldsymbol{z})\rangle|^2 = P(|0\rangle)$

Quantum advantage:

$\Phi(\boldsymbol{x})$ must be computationally hard to evaluate classically

Necessary but not sufficient condition