Kernel methods

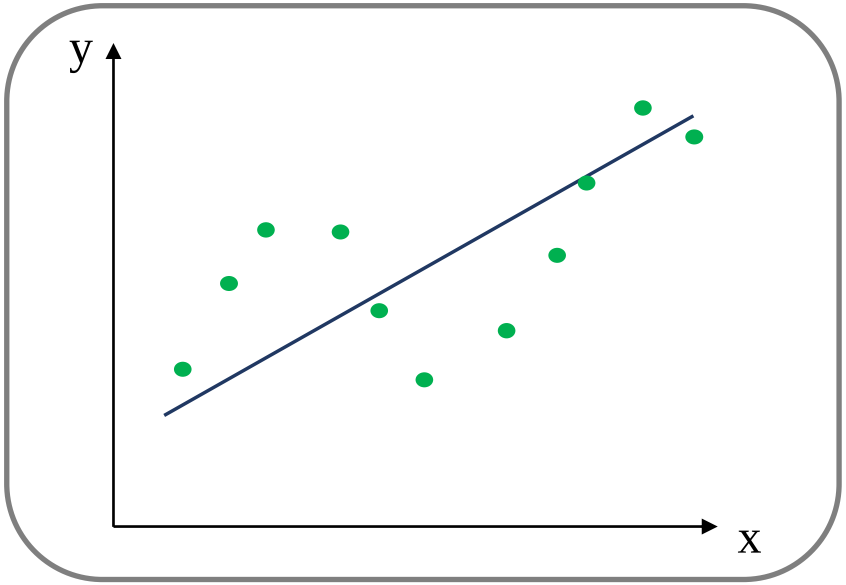
Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India

Machine Learning
DA 220

Sem 2, 2019-20

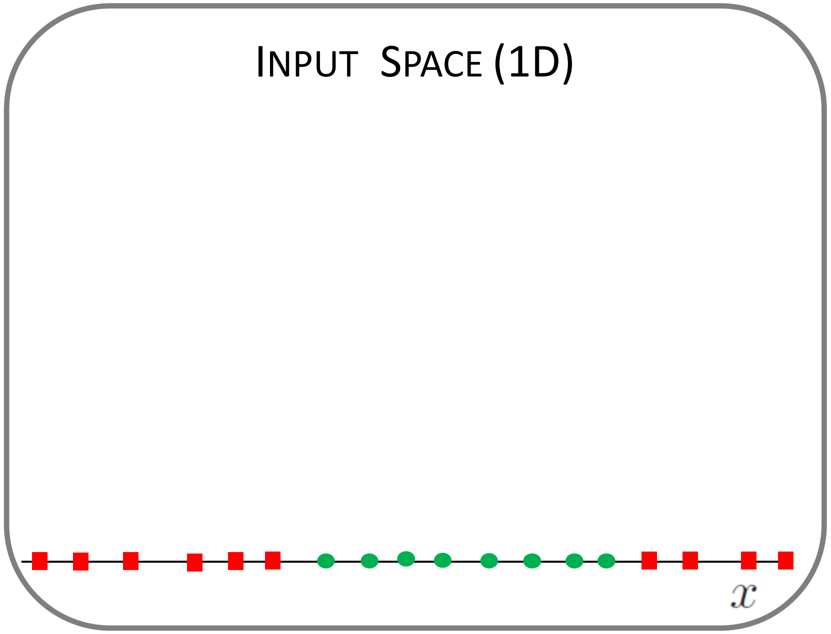
Introduction

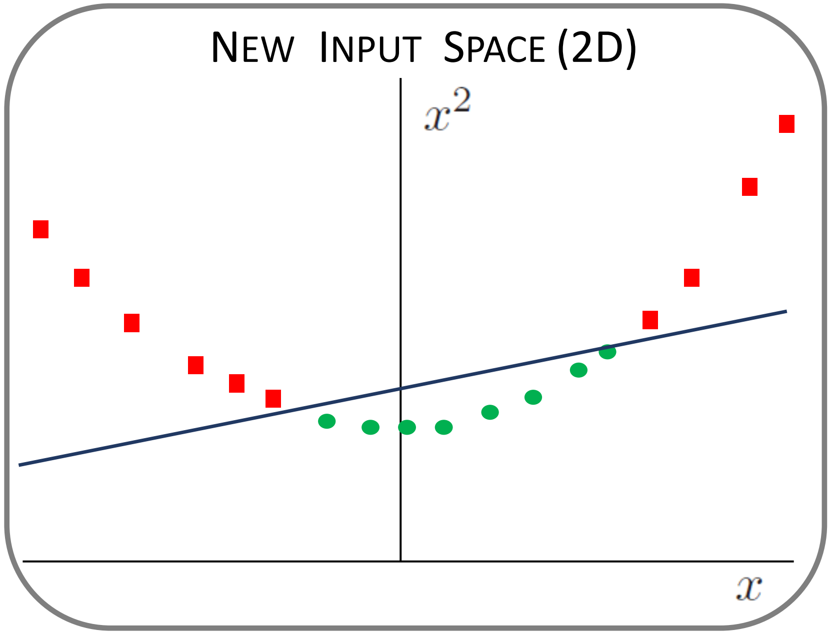
- Structures in real-world data are often non-linear.
  - Linear models are not suitable in such cases.



- Kernels project data to a higher dimensional space where the structures are linear.
  - The transformation facilitates application of linear models in the new space.
- Explicit evaluation of feature mappings can be computationally expensive, but kernel methods overcome the issue....

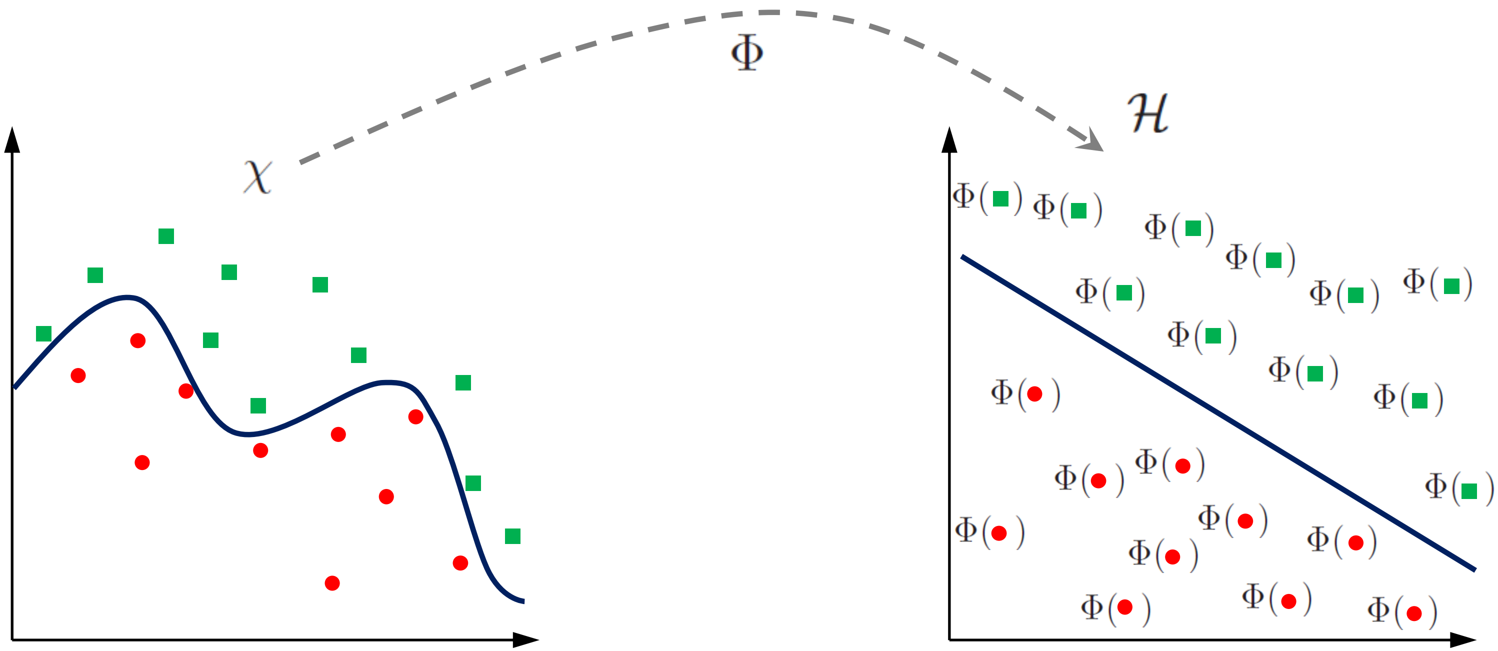
Binary classification problem



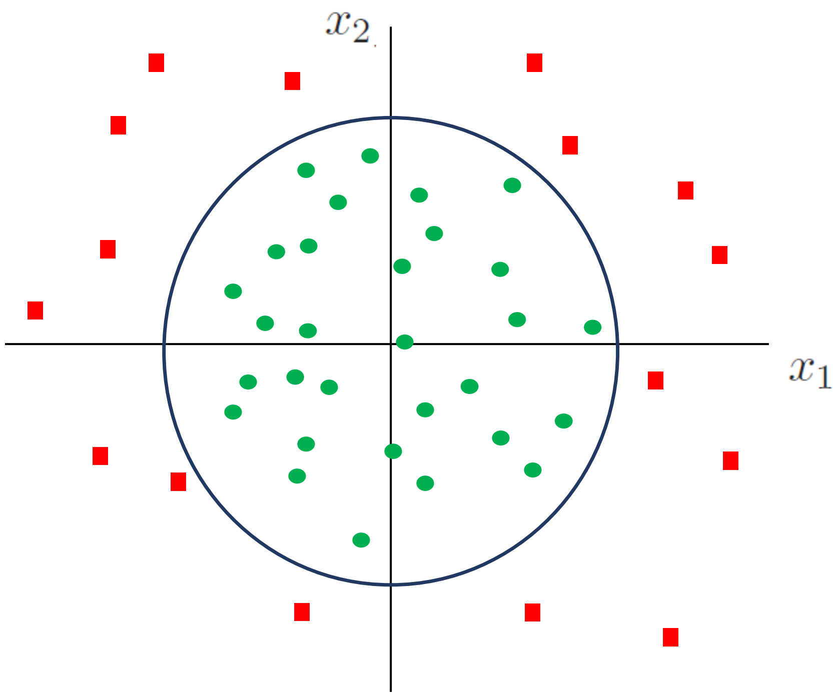


- Linear separation of data is not possible.
- Consider the following mapping: $\Phi(x) : \ x \to [x, \ x^2]$
- The dimension of the new input space is 2 as there are two features.
- Data linearly separable in the new input space.

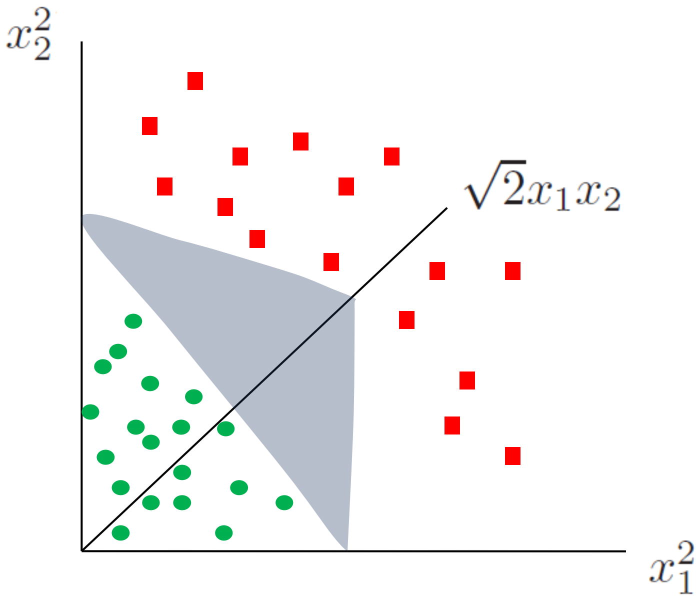
Mapping



Example



- Input space: $\mathbf{x} = [x_1 \;\; x_2]$.
- Data **not** linearly separable in input space.



- Feature space: $\Phi(\mathbf{x}) = [x_1^2 \;\; \sqrt{2}x_1x_2 \;\; x_2^2]$.
- Data linearly separable in feature space.

Figures for illustration only

Kernels

- From the previous example we have

$$\Phi(\mathbf{x}^{(i)}) = \left[(\mathbf{x}_1^{(i)})^2 \quad \sqrt{2}\mathbf{x}_1^{(i)}\mathbf{x}_2^{(i)} \quad (\mathbf{x}_2^{(i)})^2 \right] \quad \text{and} \quad \Phi(\mathbf{x}^{(j)}) = \left[(\mathbf{x}_1^{(j)})^2 \quad \sqrt{2}\mathbf{x}_1^{(j)}\mathbf{x}_2^{(j)} \quad (\mathbf{x}_2^{(j)})^2 \right]$$

- The inner product of $\Phi(\mathbf{x}^{(i)})$ and $\Phi(\mathbf{x}^{(j)})$ yields

$$\begin{aligned}
\left\langle \Phi(\mathbf{x}^{(i)}), \Phi(\mathbf{x}^{(j)}) \right\rangle &= \left\langle [(\mathbf{x}_1^{(i)})^2 \quad \sqrt{2}\mathbf{x}_1^{(i)}\mathbf{x}_2^{(i)} \quad (\mathbf{x}_2^{(i)})^2], [(\mathbf{x}_1^{(j)})^2 \quad \sqrt{2}\mathbf{x}_1^{(j)}\mathbf{x}_2^{(j)} \quad (\mathbf{x}_2^{(j)})^2] \right\rangle \\
&= (\mathbf{x}_1^{(i)})^2(\mathbf{x}_1^{(j)})^2 + 2\mathbf{x}_1^{(i)}\mathbf{x}_2^{(i)}\mathbf{x}_1^{(j)}\mathbf{x}_2^{(j)} + (\mathbf{x}_2^{(i)})^2(\mathbf{x}_2^{(j)})^2 \\
&= \left(\mathbf{x}_1^{(i)}\mathbf{x}_1^{(j)} + \mathbf{x}_2^{(i)}\mathbf{x}_2^{(j)} \right)^2 \\
&= < \mathbf{x}^{(i)}, \mathbf{x}^{(j)} >^2
\end{aligned}$$

- So the kernel function is

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = < \mathbf{x}^{(i)}, \mathbf{x}^{(j)} >^2$$

Kernels

- High dimensional mapping can lead to large number of features.
  - Computing the mapping and using the mapped representation could be inefficient.
- Kernels address these shortcomings.
- Kernels implicitly define a mapping to a high dimensional space

$$K(\mathbf{x}, \mathbf{x}') =< \Phi(\mathbf{x}), \Phi(\mathbf{x}') >.$$

- Kernel $K(\mathbf{x}, \mathbf{x}')$ efficiently computes the inner product $< \Phi(\mathbf{x}), \Phi(\mathbf{x}') >$.
- Explicitly computing $\Phi(\mathbf{x}), \Phi(\mathbf{x}')$ and then doing the inner product is more expensive.

Linear space of functions

- Consider the following space of functions:

$$\mathcal{H} = \left\{ f : \chi \to \mathcal{R} \middle| \exists \mathbf{w} \in \mathcal{R}^D, f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}, \mathbf{x} \in \mathcal{R}^D \right\}$$

- Properties:
  - Linear space
  - One-to-one correspondence between $\mathbf{w}$ and f: $\mathbf{w} \mapsto f$
- Can define the inner product of any two functions f and f' to be

$$< f, f' > = < \mathbf{w}, \mathbf{w}' >$$

- Note $f(\mathbf{x}) = < \mathbf{w}, \mathbf{x} >$
- If the weights are close, then the functions are close at all points.

$$\begin{aligned} \left| f(\mathbf{x}) - f'(\mathbf{x}) \right| &= \left| < \mathbf{w}, \mathbf{x} > - < \mathbf{w}', \mathbf{x} > \right| \\ &= \left| < \mathbf{w} - \mathbf{w}', \mathbf{x} > \right| \\ &\leq \left|\left| < \mathbf{w} - \mathbf{w}' \right|\right| \, \left|\left| \mathbf{x} \right|\right| \end{aligned}$$

Linear space of functions

- Consider a set of linearly independent features (functions):

$$\mathcal{D} = \Big\{ \phi : \chi \to \mathcal{R}, i = 1, 2, ..., P \Big\}, \quad P \in \mathbb{N}$$

where ϕ are called features. $\mathcal{D}$ is known as finite dictionary.

- Can build a space of functions such that

$$\mathcal{H} = \Big\{ f : \chi \to \mathcal{R} \Big| \exists \mathbf{w} \in \mathcal{R}^P, f(\mathbf{x}) = \sum_{i=1}^{P} w_i \phi_i(\mathbf{x}) \Big\}$$

- We have $\mathbf{w} \mapsto f$ and $< f, f' >=< \mathbf{w}, \mathbf{w}' >_{\mathcal{R}^P}$

- Want to work in a space which is infinite-dimensional.

Reproducing Kernel Hilbert space

- RKHS: Hilbert space of functions from set χ to $\mathcal{R}$.
- Key properties:
  - Hilbert space: Inner product space + completeness
  - Evaluation functionals are continuous
    This function maps functions to values. If the functions (e.g. f and f') are close then the values are also close.

$$\text{Eval}_x : \mathcal{H} \to \mathcal{R}$$

$$\text{Eval}_x(f) = f(x) \quad \forall x \in \chi$$

$$||f - f'|| < \delta \qquad \exists \epsilon_\delta \quad \text{such that} \quad |f(x) - f'(x)| < \epsilon_\delta$$

Reproducing kernel

- The function $K : \chi \times \chi \mapsto \mathcal{R}$ is called a reproducing kernel of a Hilbert space $\mathcal{H}$ when the following two conditions are satisfied:
  - $\forall \mathbf{x} \in \chi$

    $$K(\mathbf{x}, \,) = K_{\mathbf{x}} \in \mathcal{H}$$

    where $K_{\mathbf{x}}$ is a function of single variable with $\mathbf{x}$ fixed.

  - $\forall \mathbf{x} \in \chi$ and $\forall f \in \mathcal{H}$

    $$< f, K_{\mathbf{x}} >_{\mathcal{H}} = f(\mathbf{x})$$

    This is called the **reproducing property**: Inner product of the functions f and $K_{\mathbf{x}}$ yields the evaluation of the function at the point $\mathbf{x}$.

- If such a kernel K exists then $\mathcal{H}$ is called a Reproducing Kernel Hilbert space (RKHS).

Reproducing kernel

- Theorem: Function $K : \chi \times \chi \mapsto \mathcal{R}$ is **positive definite** iff it is a **reproducing kernel**.
  - For $\mathbf{x} \in \chi$ and $\mathbf{x}' \in \chi$ we have:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= < K_{\mathbf{x}}, K_{\mathbf{x}'} >_{\mathcal{H}} \\ &= < K_{\mathbf{x}'}, K_{\mathbf{x}} >_{\mathcal{H}} = K(\mathbf{x}', \mathbf{x}) \end{aligned}$$

  - For $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) \in \chi^N$, and $(a_1, a_2, ..., a_N) \in \mathcal{R}^N$ we have

$$\begin{aligned} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j < K_{\mathbf{x}_i}, K_{\mathbf{x}_j} >_{\mathcal{H}} \\ &= || \sum_{i=1}^{N} a_i K_{\mathbf{x}_i} ||_{\mathcal{H}}^2 \\ &\geq 0 \end{aligned}$$

Reproducing kernel

- Theorem (Aronszajn): For a positive definite kernel K on set χ there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\Phi : \chi \mapsto \mathcal{H}$$

such that

$$K(\mathbf{x}, \mathbf{x}') =< \Phi(\mathbf{x}), \Phi(\mathbf{x}') >_{\mathcal{H}}$$

for $\forall \mathbf{x} \in \chi$ and $\mathbf{x}' \in \chi$.

  - Consider the mapping $\Phi : \chi \mapsto \mathcal{H}$ such that $\forall \mathbf{x} \in \chi$: $\Phi(\mathbf{x}) = K_{\mathbf{x}}$.
  - Then the reproducing property yields:

$$\begin{aligned} < \Phi(\mathbf{x}), \Phi(\mathbf{x}') >_{\mathcal{H}} &= < K_{\mathbf{x}}, K_{\mathbf{x}'} >_{\mathcal{H}} \\ &= K(\mathbf{x}, \mathbf{x}') \end{aligned}$$

Kernel combinations

- If K_1 and K_2 are positive definite kernels, then the following combinations are also valid kernels:
  - $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$,
  - $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}')K_2(\mathbf{x}, \mathbf{x}')$,
  - $K(\mathbf{x}, \mathbf{x}') = \beta K_1(\mathbf{x}, \mathbf{x}')$, where $\beta \geq 0$.
- New kernels can be created by using the above rules.

Sum of kernels

$$
\begin{aligned}
k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') &= < \Phi_1(\mathbf{x}), \Phi_1(\mathbf{x}') > + < \Phi_2(\mathbf{x}), \Phi_2(\mathbf{x}') > \\
&= < [\Phi_1(\mathbf{x})\Phi_2(\mathbf{x})], [\Phi_1(\mathbf{x}')\Phi_2(\mathbf{x}')] > \\
&= k_3(\mathbf{x}, \mathbf{x}')
\end{aligned}
$$

- The summation of the two kernels corresponds to the concatenation of their respective feature spaces.

Product of kernels

$$k_1(\mathbf{x},\mathbf{x}')k_2(\mathbf{x},\mathbf{x}') = \sum_{p=1}^{P} \Phi_{1p}(\mathbf{x})\Phi_{1p}(\mathbf{x}') \sum_{m=1}^{M} \Phi_{2m}(\mathbf{x})\Phi_{2m}(\mathbf{x}')$$

$$= \sum_{p=1}^{P}\sum_{m=1}^{M} \Big(\Phi_{1p}(\mathbf{x})\Phi_{2m}(\mathbf{x})\Big)\Big(\Phi_{1p}(\mathbf{x}')\Phi_{2m}(\mathbf{x}')\Big)$$

$$= \sum_{k=1}^{PM} \Big(\Phi_{12k}(\mathbf{x})\Phi_{12k}(\mathbf{x}')\Big)$$

where $\Phi_{12}(\mathbf{x}) = \Phi_1(\mathbf{x})\Phi_2(\mathbf{x})$ is the Cartesian product

$$= < \Phi_{12}(\mathbf{x}), \Phi_{12}(\mathbf{x}') >$$

$$= k_3(\mathbf{x},\mathbf{x}')$$

Gaussian (RBF) kernel

- Consider scalar inputs:

$$
\begin{aligned}
k(x, x') &= \exp\left[-\frac{(x - x')^2}{2l^2}\right] \\
&= \exp\left[-\frac{x^2 + x'^2 - 2xx'}{2l^2}\right] \\
&= \exp\left[-\frac{x^2}{2l^2}\right] \exp\left[-\frac{x'^2}{2l^2}\right] \exp\left[\frac{xx'}{l^2}\right] \\
&= \exp\left[-\frac{x^2}{2l^2}\right] \exp\left[-\frac{x'^2}{2l^2}\right] \sum_{k=0}^{\infty} \frac{x^k x'^k}{k! l^{2k}} \\
&= \sum_{k=0}^{\infty} \left(\frac{x^k}{\sqrt{k!} l^k} \exp\left[-\frac{x^2}{2l^2}\right]\right) \left(\frac{x'^k}{\sqrt{k!} l^k} \exp\left[-\frac{x'^2}{2l^2}\right]\right) \\
&= \phi(x)\phi(x')
\end{aligned}
$$

- So this kernel maps data to an infinite-dimensional feature space.

Kernel trick

- Algorithms that can be expressed in terms of pairwise inner products of inputs can also be applied to the feature space of a kernel by replacing the inner product with a kernel evaluation.

- This is possible because the kernel is an inner product in the feature space.

Representer theorem (simplified)

- Given a set χ, a kernel k, corresponding RKHS $\mathcal{H}$, and a (loss) function $\mathcal{L}(.\,,\,.)$, the solutions of the optimization problem

$$\arg\min_{f \in \mathcal{H}} \sum_{n=1}^{N} \mathcal{L}\left(f\left(\mathbf{x}^{(n)}\right), y^{(n)}\right) + \lambda ||f||_{\mathcal{H}}^{2}$$

admits the following representation

$$f = \sum_{n=1}^{N} \alpha_n k\left(\mathbf{x}^{(n)},\ .\ \right)$$

$$\Rightarrow f(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k\left(\mathbf{x}^{(n)}, \mathbf{x}\right)$$

Although the optimization problem can potentially be in an infinite dimensional space $\mathcal{H}$, the solution lies in the span of N kernels centered at the N data points.

Representer theorem: derivation

- Let $\mathcal{H}_s$ be the linear span of vectors $k(\mathbf{x}^{(n)}, \cdot)$ in $\mathcal{H}$, i.e.

$$\mathcal{H}_s = \left\{ f \in \mathcal{H} : f(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}^{(n)}, \mathbf{x}), (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathcal{R}^N \right\}$$

- Any function $f \in \mathcal{H}$ can be expressed as

$$f = f_s + f_\perp$$

where $f \in \mathcal{H}_s$ and $f_\perp \perp \mathcal{H}_s$ (component perpendicular to the subspace $\mathcal{H}_s$).

- Can write

$$\begin{aligned} ||f||^2_{\mathcal{H}} &= ||f_s||^2_{\mathcal{H}} + ||f_\perp||^2_{\mathcal{H}} \\ &\geq ||f_s||^2_{\mathcal{H}} \end{aligned}$$

  - Therefore $||f||_{\mathcal{H}}$ is minimized when f lies in $\mathcal{H}_s$.

Representer theorem: derivation

- Also from the reproducing property of the kernel we have for each n

$$\begin{aligned} f(\mathbf{x}^{(n)}) &= \langle f, k(\mathbf{x}^{(n)}, \cdot) \rangle_{\mathcal{H}} \\ &= \langle f_s, k(\mathbf{x}^{(n)}, \cdot) \rangle_{\mathcal{H}} + \langle f_\perp, k(\mathbf{x}^{(n)}, \cdot) \rangle_{\mathcal{H}} \\ &= \langle f_s, k(\mathbf{x}^{(n)}, \cdot) \rangle_{\mathcal{H}} \\ &= f_s(\mathbf{x}^{(n)}) \end{aligned}$$

- Therefore

$$\sum_{n=1}^{N} \mathcal{L}\big(f(\mathbf{x}^{(n)}), y^{(n)}\big) = \sum_{n=1}^{N} \mathcal{L}\big(f_s(\mathbf{x}^{(n)}), y^{(n)}\big)$$

  - So the loss function depends only on components of f that lie in $\mathcal{H}_s$.

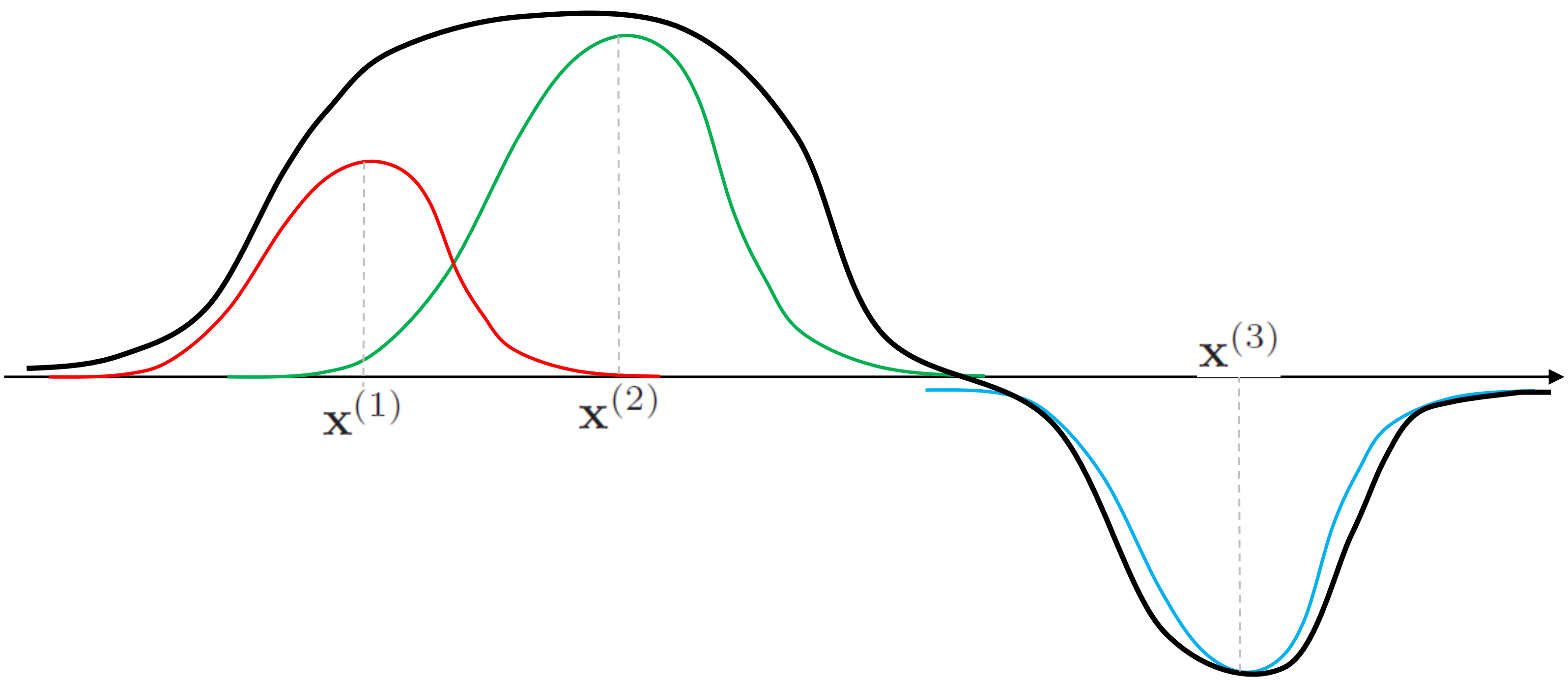
- Since we are minimizing the objective, so taking $||f_\perp||_{\mathcal{H}} = 0$, we can express the solution to be of the form

$$f(\cdot) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}^{(n)}, \cdot)$$

Representer theorem

If we are given three input points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$, then we have

$$f(\mathbf{x}) = \alpha_1 k(\mathbf{x}^{(1)}, \mathbf{x}) + \alpha_2 k(\mathbf{x}^{(2)}, \mathbf{x}) + \alpha_3 k(\mathbf{x}^{(3)}, \mathbf{x})$$



Kernel-SVM

- Recall the SVM objective (dual formulation):

$$\max_{0 \le \boldsymbol{\lambda} \le C} -\frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} \lambda_m \lambda_n y^{(m)} y^{(n)} ((\mathbf{x}^{(m)})^{\mathrm{T}} \mathbf{x}^{(n)}) + \sum_{n=1}^{N} \lambda_n \quad \text{subject to} \quad \sum_{n=1}^{N} \lambda_n y^{(n)} = 0$$

- Let Φ be the feature map corresponding to some kernel k. Then

$$k(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) =< \Phi(\mathbf{x}^{(m)}), \Phi(\mathbf{x}^{(n)}) >$$

- Replacing the inner product $< \mathbf{x}^{(m)}, \mathbf{x}^{(n)} >$ in the objective function by $< \Phi(\mathbf{x}^{(m)}), \Phi(\mathbf{x}^{(n)}) >$ yields

$$\max_{0 \le \boldsymbol{\lambda} \le C} -\frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} \lambda_m \lambda_n y^{(m)} y^{(n)} < \Phi(\mathbf{x}^{(m)}), \Phi(\mathbf{x}^{(n)}) > + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad \sum_{n=1}^{N} \lambda_n y^{(n)} = 0$$

- By using this formulation the SVM learns a linear separator in the feature space $\mathcal{H}$ which corresponds to a non-linear decision boundary in the original input space.

Kernel-SVM

- Solution to $\mathbf{w}$ in original SVM formulation:

$$\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}^{(n)}$$

- Solution to $\mathbf{w}$ in kernel-SVM formulation:

$$\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \Phi\left(\mathbf{x}^{(n)}\right) = \sum_{n=1}^{N} \lambda_n y^{(n)} k\left(\mathbf{x}^{(n)},\ .\ \right)$$

- Test prediction at $\mathbf{x}^*$ in original SVM formulation:

$$y^* = \operatorname{sign}\left(\mathbf{w}^{\mathrm{T}} \mathbf{x}^*\right) = \operatorname{sign}\left(\sum_{n=1}^{N} \lambda_n y^{(n)} \left(\mathbf{x}^{(n)}\right)^{\mathrm{T}} \mathbf{x}^*\right)$$

- Test prediction at $\mathbf{x}^*$ in kernel-SVM formulation:

$$y^* = \operatorname{sign}\left(\mathbf{w}^{\mathrm{T}} \Phi\left(\mathbf{x}^*\right)\right) = \operatorname{sign}\left(\sum_{n=1}^{N} \lambda_n y^{(n)} < \Phi\left(\mathbf{x}^{(n)}\right), \Phi\left(\mathbf{x}^*\right) > \right)$$

$$= \operatorname{sign}\left(\sum_{n=1}^{N} \lambda_n y^{(n)} k\left(\mathbf{x}^{(n)}, \mathbf{x}^*\right)\right)$$

Kernel matrix

- Also known as Gram matrix.
- Formed by applying the kernel function k to all pairs of data points in $\mathbf{X}$.

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & \cdot & \cdot & k(\mathbf{x}^{(1)}, \mathbf{x}^{(N)}) \\ k(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}) & \cdot & \cdot & k(\mathbf{x}^{(2)}, \mathbf{x}^{(N)}) \\ k(\mathbf{x}^{(3)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(3)}, \mathbf{x}^{(2)}) & \cdot & \cdot & k(\mathbf{x}^{(3)}, \mathbf{x}^{(N)}) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ k(\mathbf{x}^{(N)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(N)}, \mathbf{x}^{(2)}) & \cdot & \cdot & k(\mathbf{x}^{(N)}, \mathbf{x}^{(N)}) \end{bmatrix}$$

- Square matrix of size $N \times N$.
- Symmetric.