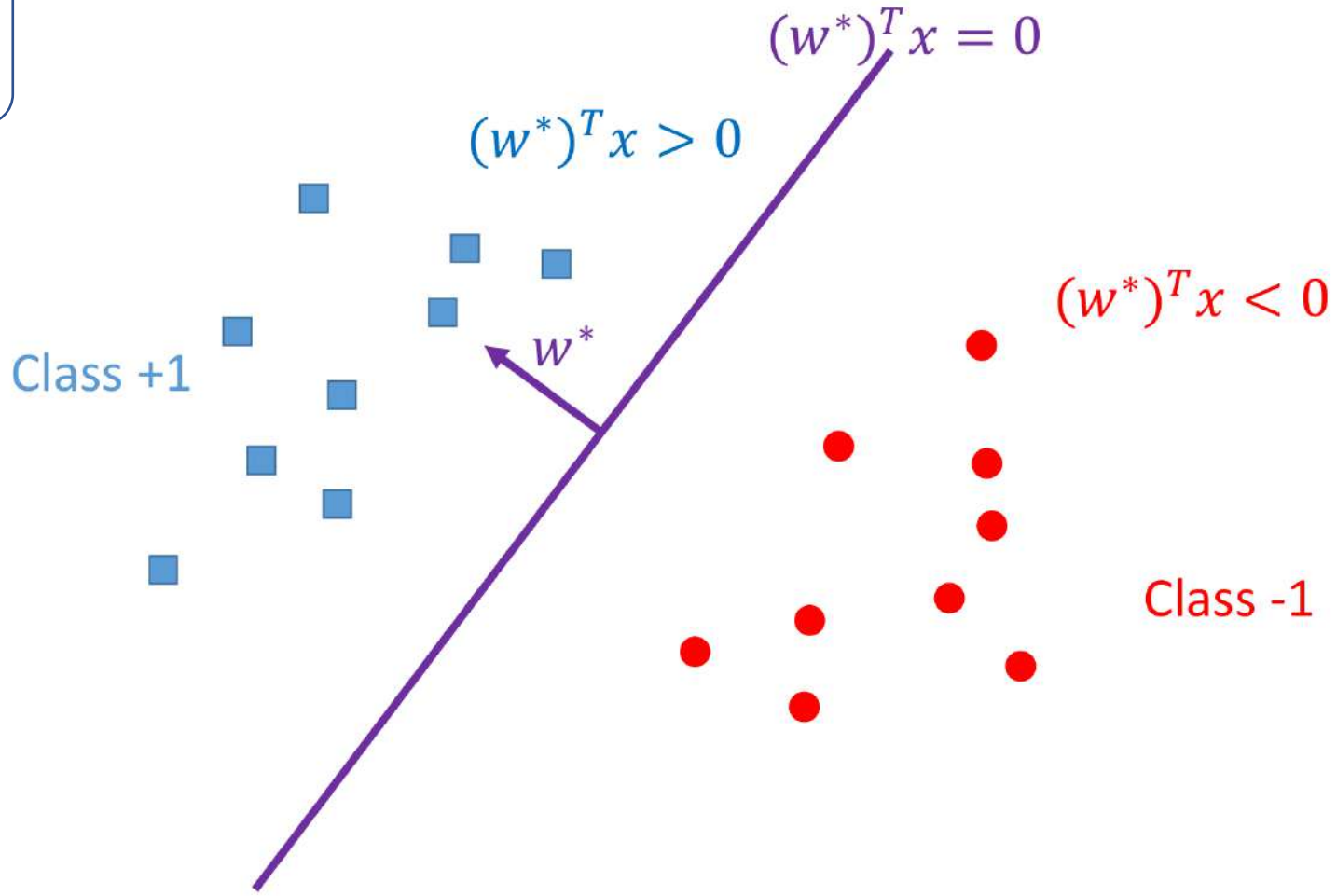
From kernel machines to neural networks

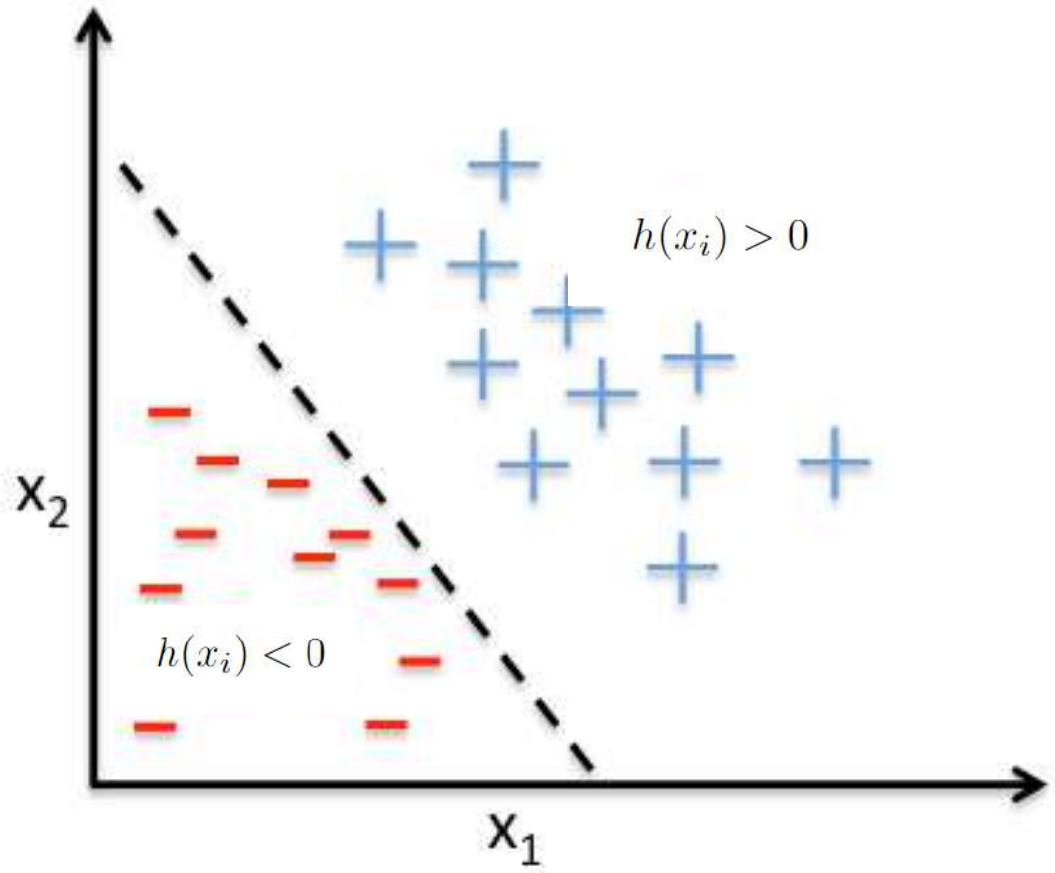
Class1

- Recap of kernel origin: perceptron support vector machines
- Embedding into feature spaces.
- From linear regression to kernels, kernel trick.

Binary Task



Classifying points



$$h(x_i) = sign(w^T x_i + b)$$

Absorbing the bias

b is the bias term (without the bias term, the hyperplane that $\mathbf{w}$ defines would always have to go through the origin). Dealing with b can be a pain, so we 'absorb' it into the feature vector $\mathbf{w}$ by adding one additional *constant* dimension. Under this convention,

$$\mathbf{x}_i \text{ becomes } \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$$

$$\mathbf{w} \text{ becomes } \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$$

We can verify that

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}^\top \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \mathbf{w}^\top \mathbf{x}_i + b$$

Using this, we can simplify the above formulation of $h(\mathbf{x}_i)$ to

$$h(\mathbf{x}_i) = \text{sign}(\mathbf{w}^\top \mathbf{x})$$

Summarizing:

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Hypothesis $f_w(x) = w^T x$
  - $y = +1$ if $w^T x > 0$
  - $y = -1$ if $w^T x < 0$
- Prediction: $y = \text{sign}(f_w(x)) = \text{sign}(w^T x)$

- Goal: minimize classification error

How do we minimize the error?

Perceptron Algorithm

- Assume for simplicity: all x_i has length 1

1. Start with the all-zeroes weight vector $\mathbf{w}_1 = \mathbf{0}$, and initialize t to 1.
2. Given example $\mathbf{x}$, predict positive iff $\mathbf{w}_t \cdot \mathbf{x} > 0$.
3. On a mistake, update as follows:
   - Mistake on positive: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}$.
   - Mistake on negative: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}$.

   $t \leftarrow t + 1$.

Why is this a reasonable strategy?

Intuition: correct the current mistake

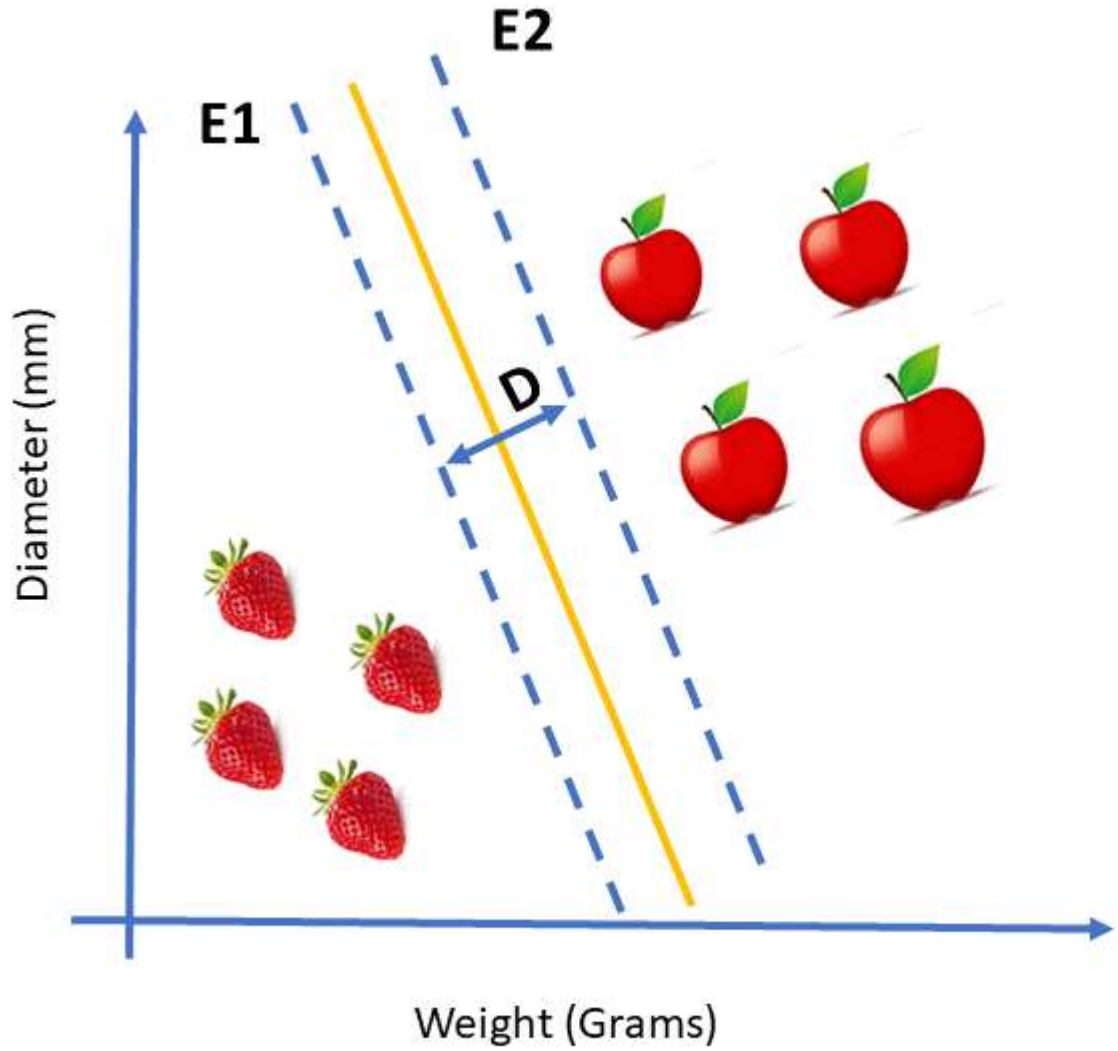
- If mistake on a positive example

$$w_{t+1}^T x = (w_t + x)^T x = w_t^T x + x^T x = w_t^T x + 1$$

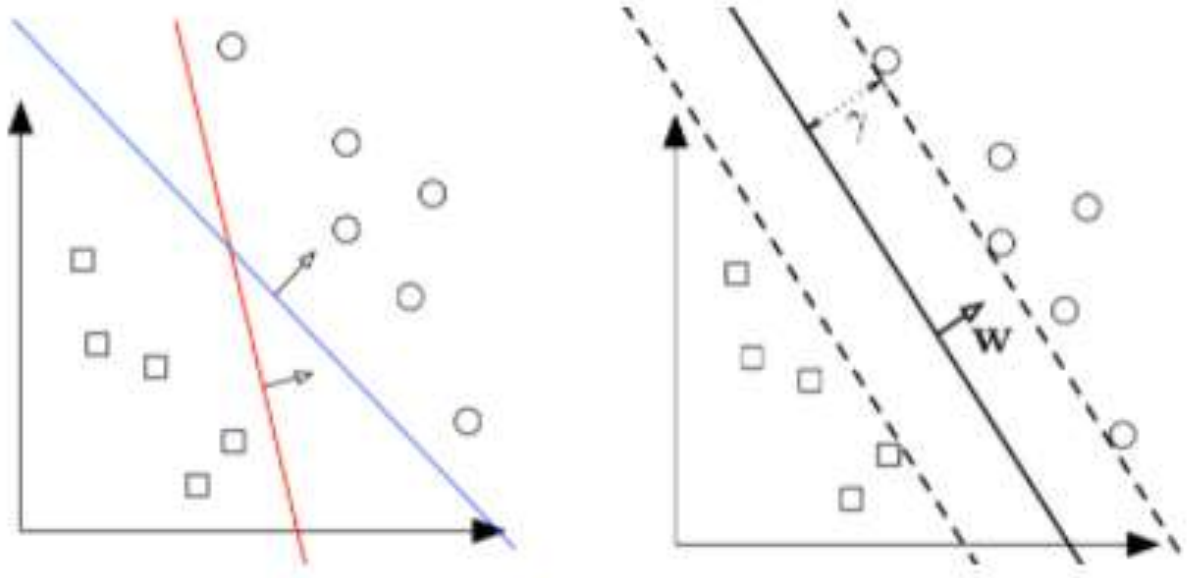
- If mistake on a negative example

$$w_{t+1}^T x = (w_t - x)^T x = w_t^T x - x^T x = w_t^T x - 1$$

Beyond the perceptron: Maximum margin



Maximum margin



What is the best separating hyperplane?
What about robustness?

The one that maximizes the distances to the closest data points from both classes.
This is the hyperplane with maximum margin
Let's define more precisely the concept of margin

What is the distance of a point $\mathbf{x}$ to the hyperplane $\mathcal{H}$?

Consider some point $\mathbf{x}$. Let $\mathbf{d}$ be the vector from $\mathcal{H}$ to $\mathbf{x}$ of minimum length. Let $\mathbf{x}^P$ be the projection of $\mathbf{x}$ onto $\mathcal{H}$. It follows then that:

$\mathbf{x}^P = \mathbf{x} - \mathbf{d}$.

$\mathbf{d}$ is parallel to $\mathbf{w}$, so $\mathbf{d} = \alpha\mathbf{w}$ for some $\alpha \in \mathbb{R}$.

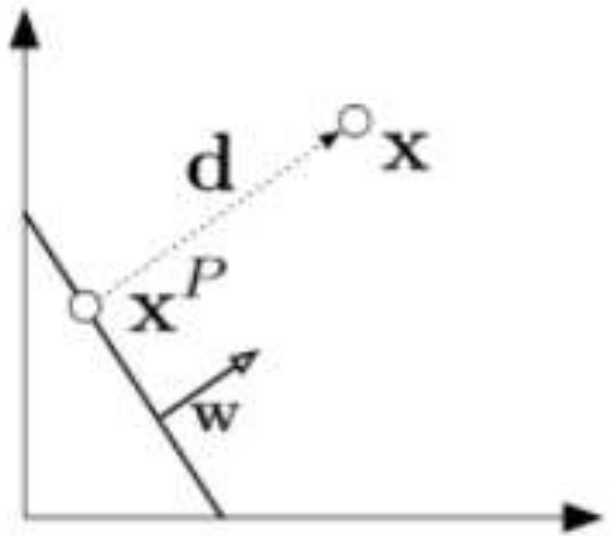
$\mathbf{x}^P \in \mathcal{H}$ which implies $\mathbf{w}^T\mathbf{x}^P + b = 0$

therefore $\mathbf{w}^T\mathbf{x}^P + b = \mathbf{w}^T(\mathbf{x} - \mathbf{d}) + b = \mathbf{w}^T(\mathbf{x} - \alpha\mathbf{w}) + b = 0$

which implies $\alpha = \frac{\mathbf{w}^T\mathbf{x}+b}{\mathbf{w}^T\mathbf{w}}$

The length of $\mathbf{d}$:

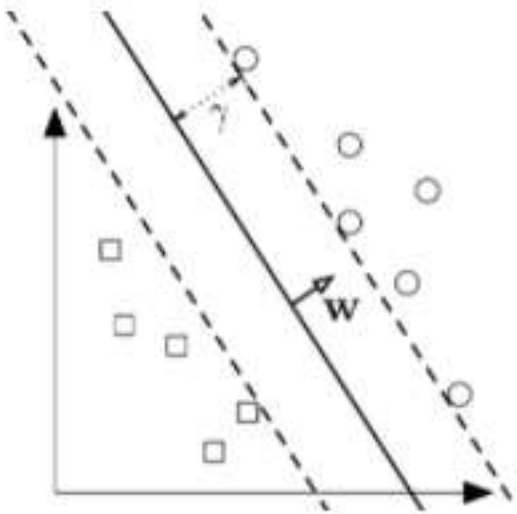
$$\|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^T\mathbf{d}} = \sqrt{\alpha^2\mathbf{w}^T\mathbf{w}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\sqrt{\mathbf{w}^T\mathbf{w}}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}$$



Margin of $\mathcal{H}$ with respect to D: $\gamma(\mathbf{w}, b) = \min_{\mathbf{x}\in D} \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}$

We can formulate our search for the maximum margin separating hyperplane as a constrained optimization problem. The objective is to maximize the margin under the constraints that all data points must lie on the correct side of the hyperplane:

$$\underbrace{\max_{\mathbf{w},b} \gamma(\mathbf{w}, b)}_{\textit{maximize margin}} \text{ such that } \underbrace{\forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0}_{\textit{separating hyperplane}}$$



If we plug in the definition of γ we obtain:

$$\underbrace{\max_{\mathbf{w},b} \underbrace{\frac{1}{\|\mathbf{w}\|_2} \min_{\mathbf{x}_i \in D} \left|\mathbf{w}^T \mathbf{x}_i + b\right|}_{\gamma(\mathbf{w},b)}}_{\textit{maximize margin}} \quad s.t. \quad \underbrace{\forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0}_{\textit{separating hyperplane}}$$

Because the hyperplane is scale invariant, we can fix the scale of $\mathbf{w}, b$ anyway we want. Let's be clever about it, and choose it such that

$$\min_{\mathbf{x} \in D} \left|\mathbf{w}^T \mathbf{x} + b\right| = 1.$$

We can add this re-scaling as an equality constraint. Then our objective becomes:

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2} \cdot 1 = \min_{\mathbf{w},b} \|\mathbf{w}\|_2 = \min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w}$$

(Where we made use of the fact $f(z) = z^2$ is a monotonically increasing function for $z \geq 0$ and $\|\mathbf{w}\| \geq 0$; i.e. the $\mathbf{w}$ that maximizes $\|\mathbf{w}\|_2$ also maximizes $\mathbf{w}^\top \mathbf{w}$.)

The new optimization problem becomes:

$$\begin{aligned} \min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w} \\ \text{s.t.} \quad \begin{aligned} \forall i,\ y_i(\mathbf{w}^T\mathbf{x}_i + b) &\geq 0 \\ \min_i \left|\mathbf{w}^T\mathbf{x}_i + b\right| &= 1 \end{aligned} \end{aligned}$$

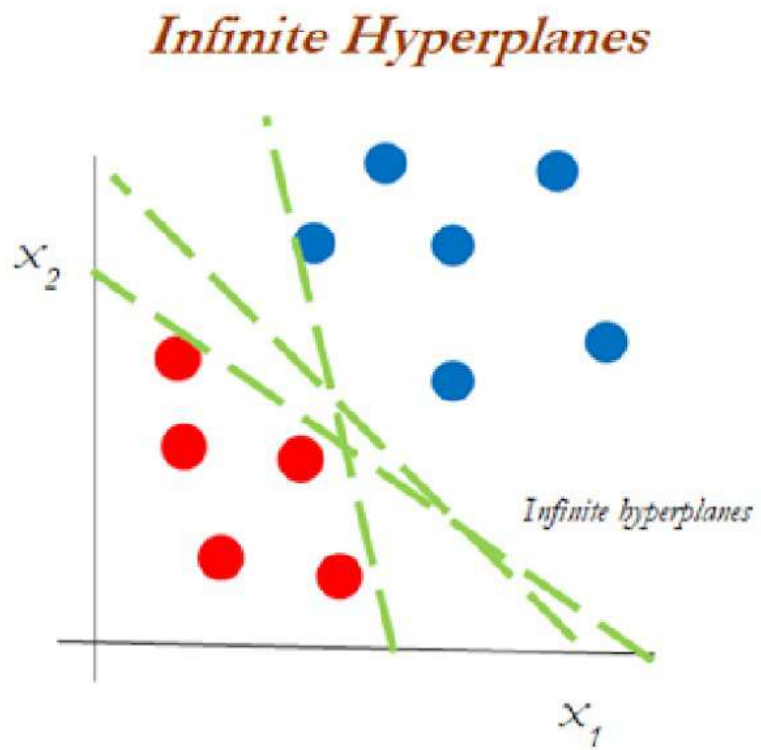
These constraints are still hard to deal with, however luckily we can show that (for the optimal solution) they are equivalent to a much simpler formulation. (Makes sure you know how to prove that the two sets of constraints are equivalent.)

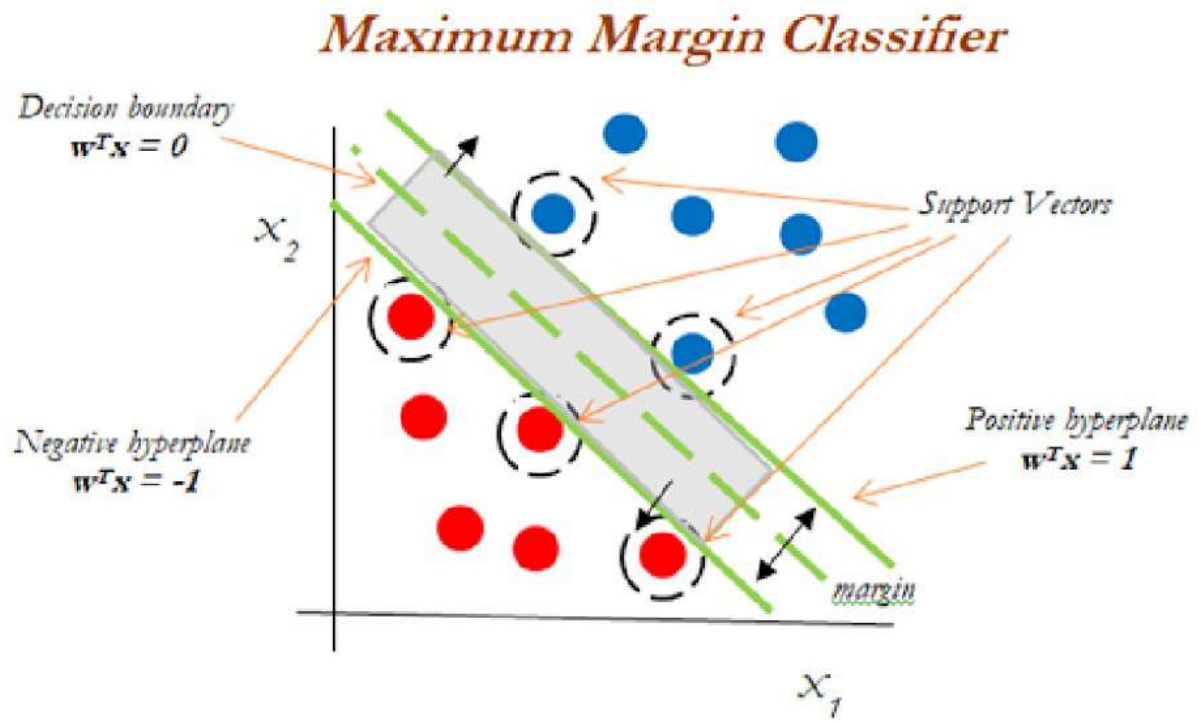
$$\begin{aligned} &\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} \\ &\text{s.t.} \quad \forall i\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \end{aligned}$$

Support vectors

For the optimal $\mathbf{w}, b$ pair, some training points will have tight constraints, i.e.

$$y_i(\mathbf{w}^T\mathbf{x_i} + b) = 1.$$





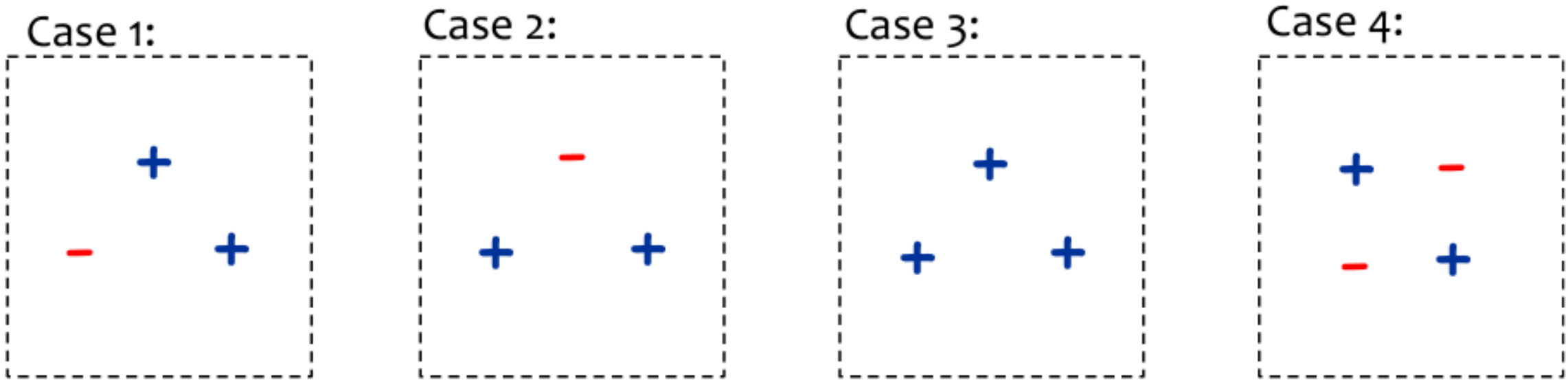
Beyond maximal margin classifiers

Maximal margin classifier limitations.

Soft margin classifier, **slack parameters**: learning, role of the parameter C

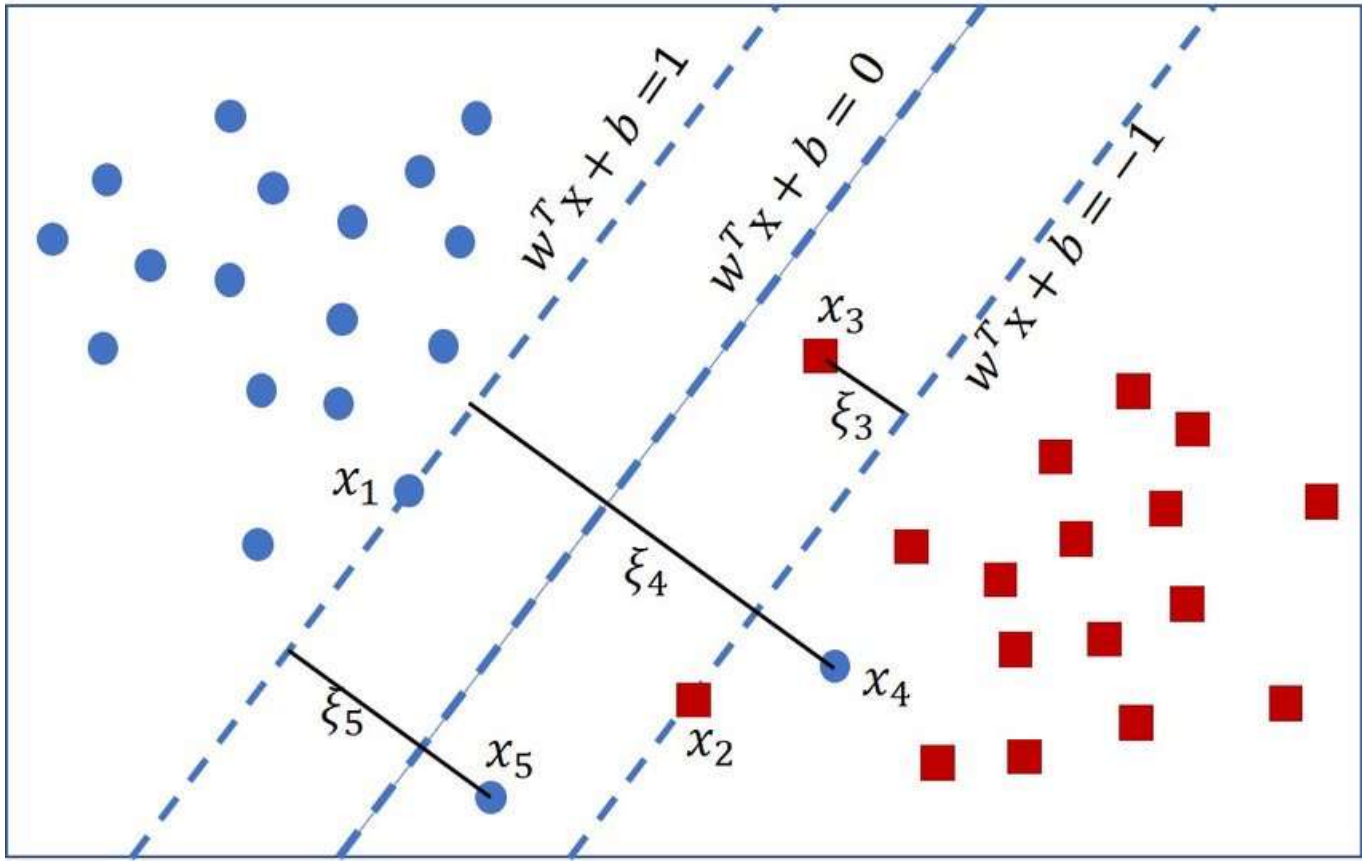
Dual problem and optimization: origin of kernels

Linear separability

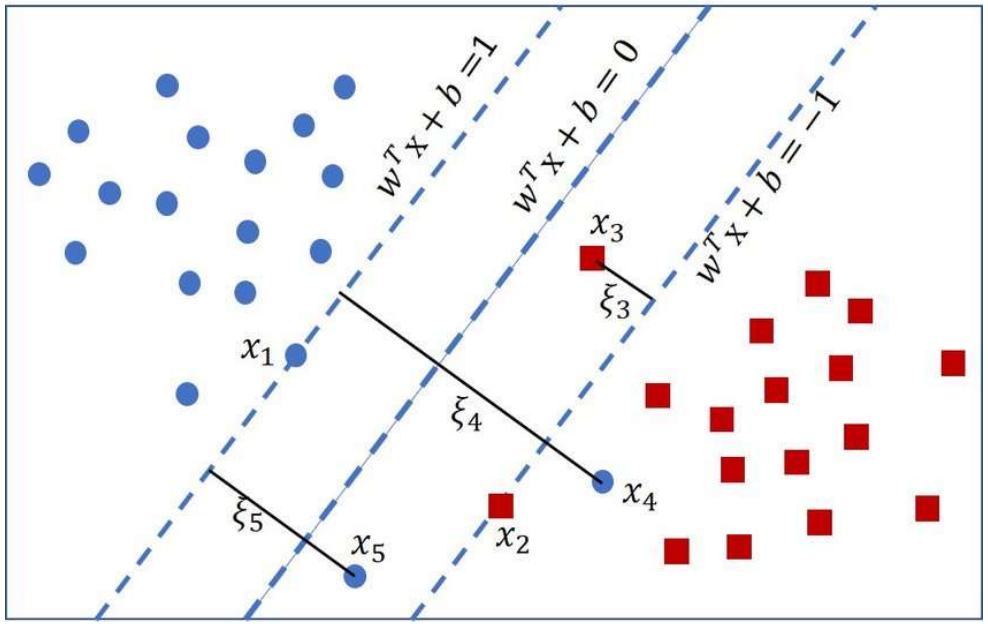


Can you separate + and - with a plane in all cases?

Which is the problem here?



Maximal margin classification and limitations



If the data is low dimensional it is often the case that there is no separating hyperplane between the two classes. In this case, there is no solution to the optimization problems stated above. We can fix this by allowing the constraints to be violated ever so slight with the introduction of slack variables:

$$\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \xi_i$$
$$s.t.\ \forall i\ y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$$
$$\forall i\ \xi_i \geq 0$$

The slack variable ξ_i allows the input $\mathbf{x_i}$ to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack". If C is very large, the SVM becomes very strict and tries to get all points to be on the right side of the hyperplane. If C is very small, the SVM becomes very loose and may "sacrifice" some points to obtain a simpler (i.e. lower $\|\mathbf{w}\|_2^2$) solution.

New optimization problem

Let us consider the value of ξ_i for the case of $C \neq 0$. Because the objective will always try to minimize ξ_i as much as possible, the equation must hold as an *equality* and we have:

$$\xi_i = \begin{cases} 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) & \text{if } y_i(\mathbf{w}^T\mathbf{x}_i + b) < 1 \\ 0 & \text{if } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \end{cases}$$

This is equivalent to the following closed form:

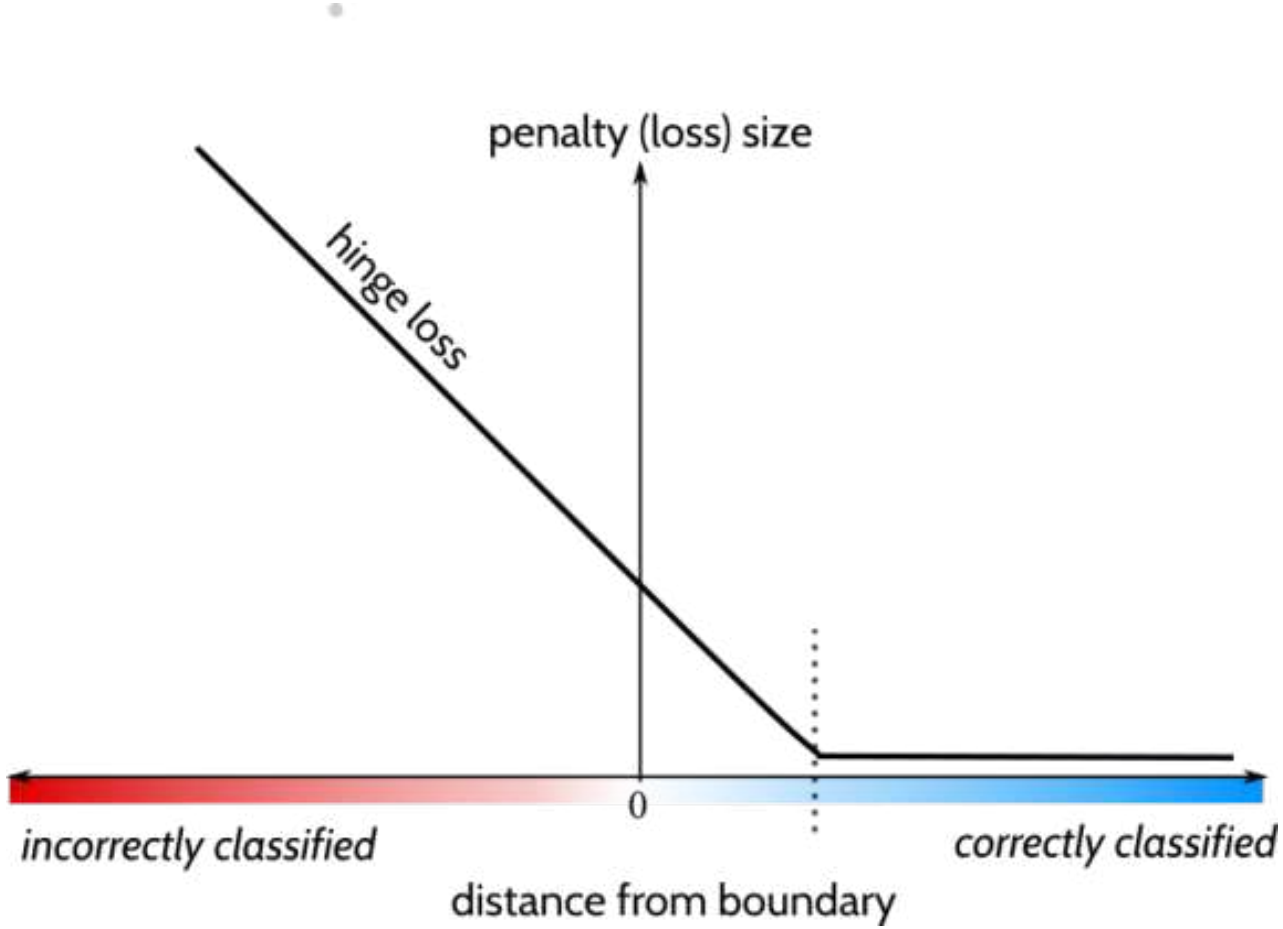
$$\xi_i = \max(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b), 0).$$

If we plug this closed form into the objective of our SVM optimization problem, we obtain the following *unconstrained* version as loss function and regularizer:

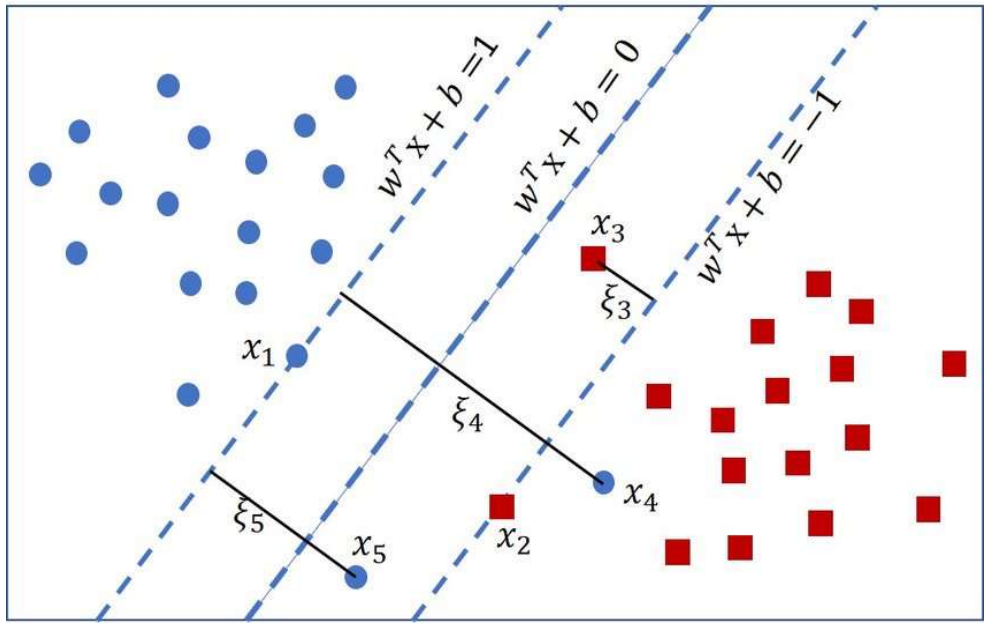
$$\min_{\mathbf{w},b} \underbrace{\mathbf{w}^T\mathbf{w}}_{l_2-regularizer} + C \sum_{i=1}^{n} \underbrace{\max\left[1 - y_i(\mathbf{w}^T\mathbf{x} + b), 0\right]}_{hinge-loss}$$

This formulation allows us to optimize the SVM paramters $(\mathbf{w}, b)$ just like logistic regression (e.g. through gradient descent). The only difference is that we have the **hinge-loss** instead of the **logistic loss**.

Hinge loss


penalty (loss) size
hinge loss
0
incorrectly classified
correctly classified
distance from boundary

Solving for the new optimization problem

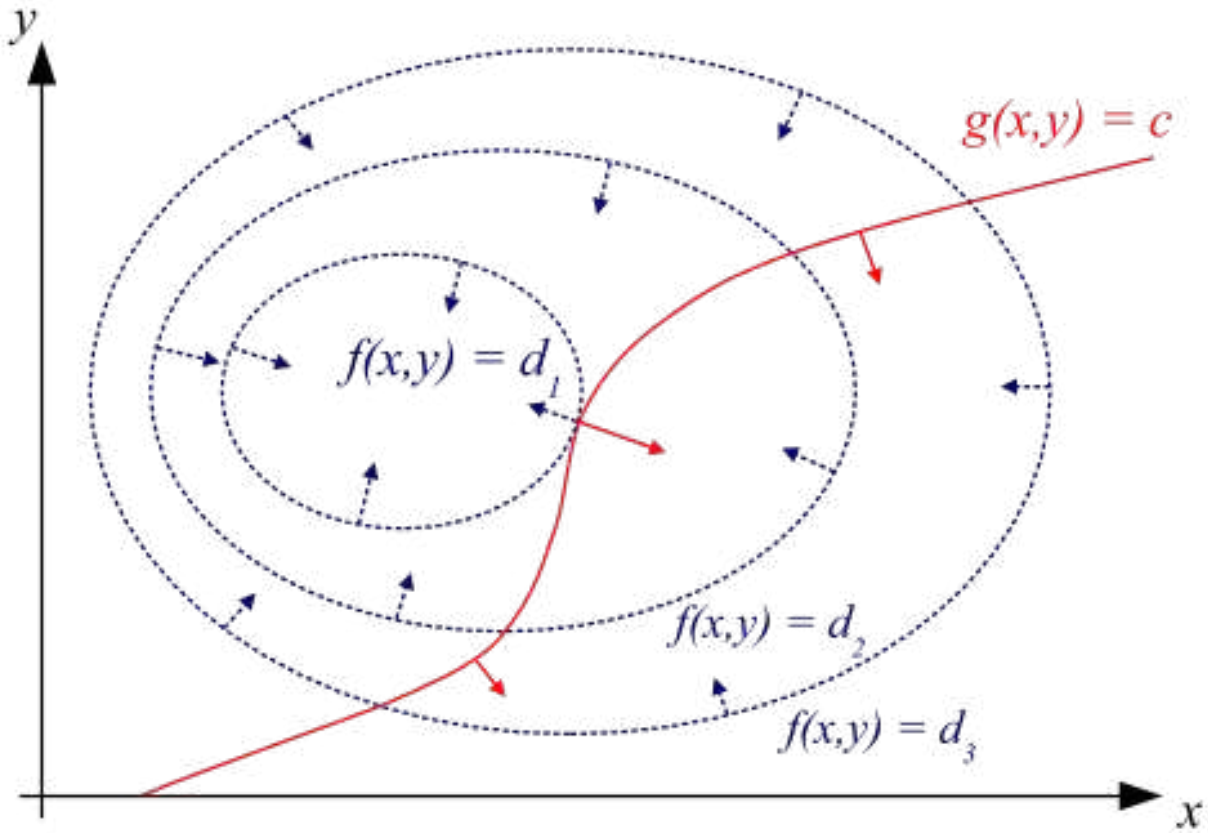


slack penalty

$$\min_{\substack{w \in \mathbb{R}^d \\ \xi \geq 0}} \quad \frac{1}{2} w^\top w + C \sum_{i=1}^{n} \xi_i$$

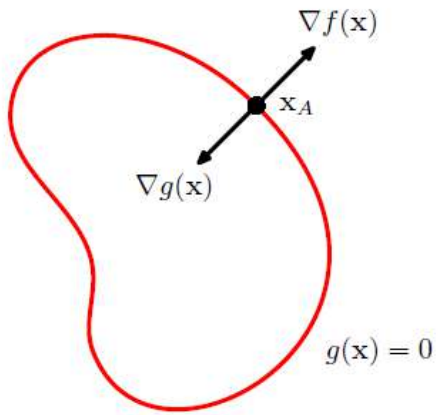
$$\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i \qquad \forall i \in \{1, \ldots, n\}$$

Lagrangian multipliers



$$\max \ f(x,y), \quad s.t. \quad g(x,y) = 0$$

Lagrange Multipliers



Consider the problem:

$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) = 0$$

- Points on $g(\boldsymbol{x}) = 0$ must have $\nabla g(\boldsymbol{x})$ normal to surface
- A stationary point must have no change in f in the direction of the constraint surface, so $\nabla f(\boldsymbol{x})$ must also be normal to the surface.
  - So there must be some $\lambda \neq 0$ such that $\nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$
- Define Lagrangian:

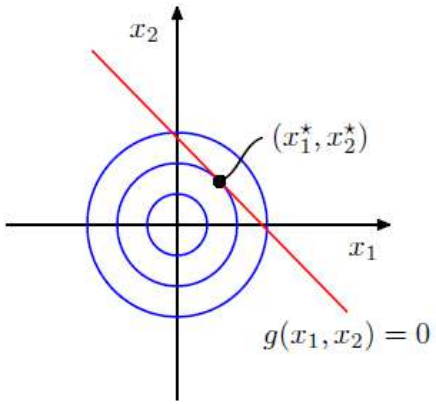
$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

- Stationary points of $L(\boldsymbol{x}, \lambda)$ have $\nabla_x L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$ and $\nabla_\lambda L(\boldsymbol{x}, \lambda) = g(\boldsymbol{x}) = 0$

This is because on the curves g is constant

This is because they are the same vector up to a multiplicative constant

Lagrange Multipliers Example



- Consider the problem

$$\max_{\boldsymbol{x}} f(x_1, x_2) = 1 - x_1^2 - x_2^2$$
$$s.t. \quad g(x_1, x_2) = x_1 + x_2 - 1 = 0$$

- Lagrangian:

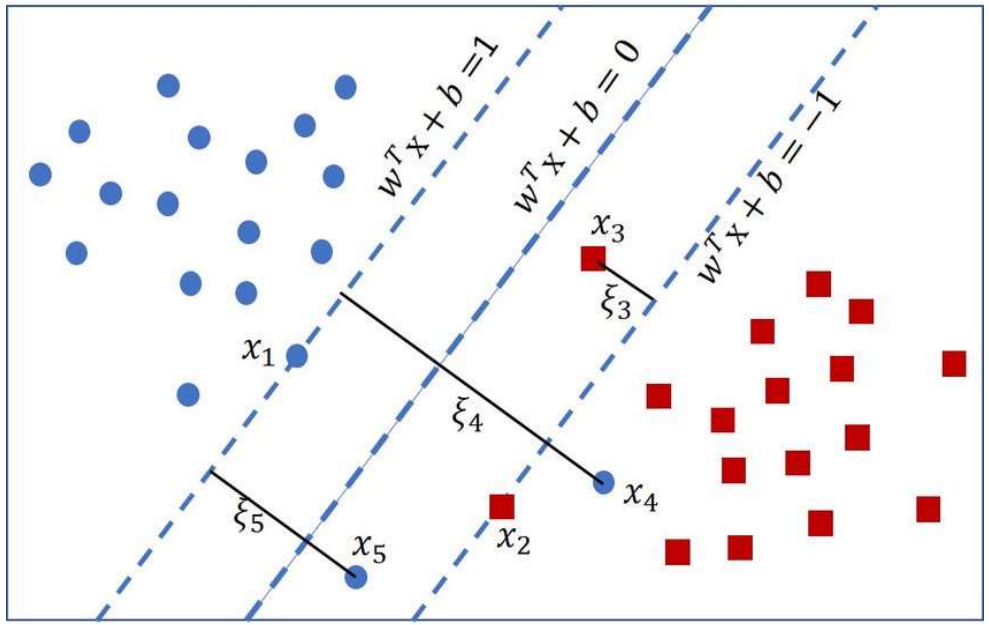
$$L(\boldsymbol{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

- Stationary points require:

$$\begin{aligned} \partial L/\partial x_1 &= -2x_1 + \lambda = 0 \\ \partial L/\partial x_2 &= -2x_2 + \lambda = 0 \\ \partial L/\partial \lambda &= x_1 + x_2 - 1 = 0 \end{aligned}$$

- So stationary point is $(x_1^*, x_2^*) = (\frac{1}{2}, \frac{1}{2})$, $\lambda = 1$

Solving for the new optimization problem



slack penalty

$$\min_{\substack{w \in \mathbb{R}^d \\ \xi \geq 0}} \quad \frac{1}{2} w^\top w + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i \qquad \forall i \in \{1, \ldots, n\}$$

The Lagrangian: $L(w, b, \xi, \alpha, \lambda)$

$$= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\left(w^\top x_i + b\right) - \xi_i\right) + \sum_{i=1}^{n}\lambda_i(-\xi_i)$$

with dual variable constraints

$$\alpha_i \geq 0, \qquad \lambda_i \geq 0.$$

Minimize wrt the primal variables w, b, and ξ.

Derivative wrt w:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \qquad w = \sum_{i=1}^{n} \alpha_i y_i x_i.$$

Derivative wrt b:

$$\frac{\partial L}{\partial b} = \sum_{i} y_i \alpha_i = 0.$$

Derivative wrt ξ_i:

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \lambda_i = 0 \qquad \alpha_i = C - \lambda_i.$$

Noting that $\lambda_i \geq 0$,

$$\alpha_i \leq C.$$

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^\top w + C \sum_i \xi_i - \sum_i \alpha_i (y_i (w^\top x_i + b) - 1 + \xi_i) - \sum_i \beta_i \xi_i$$

Gradients

$$\nabla_w L = w - \sum_i \alpha_i y_i x_i = 0$$

$$\nabla_b L = -\sum_i \alpha_i y_i = 0$$

$$\nabla_\xi L = C - \alpha - \beta = 0$$

Consequences

$$w = \sum_i \alpha_i y_i x_i$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha = C - \beta$$

$$\beta = C - \alpha$$

w is a linear combination of the training points!

Rewriting the Lagrangian…

$$
\begin{aligned}
L(\alpha,\lambda) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\left(w^\top x_i + b\right) - \xi_i\right) \\
&\quad + \sum_{i=1}^{n}\lambda_i(-\xi_i) \\
&= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^\top x_j + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^\top x_j \\
&\quad - b\underbrace{\sum_{i=1}^{m}\alpha_i y_i}_{0} + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\underbrace{(C-\alpha_i)}_{\lambda_i}\xi_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^\top x_j.
\end{aligned}
$$

$$L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^\top x_j,$$

subject to the constraints

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

subject to the constraints

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

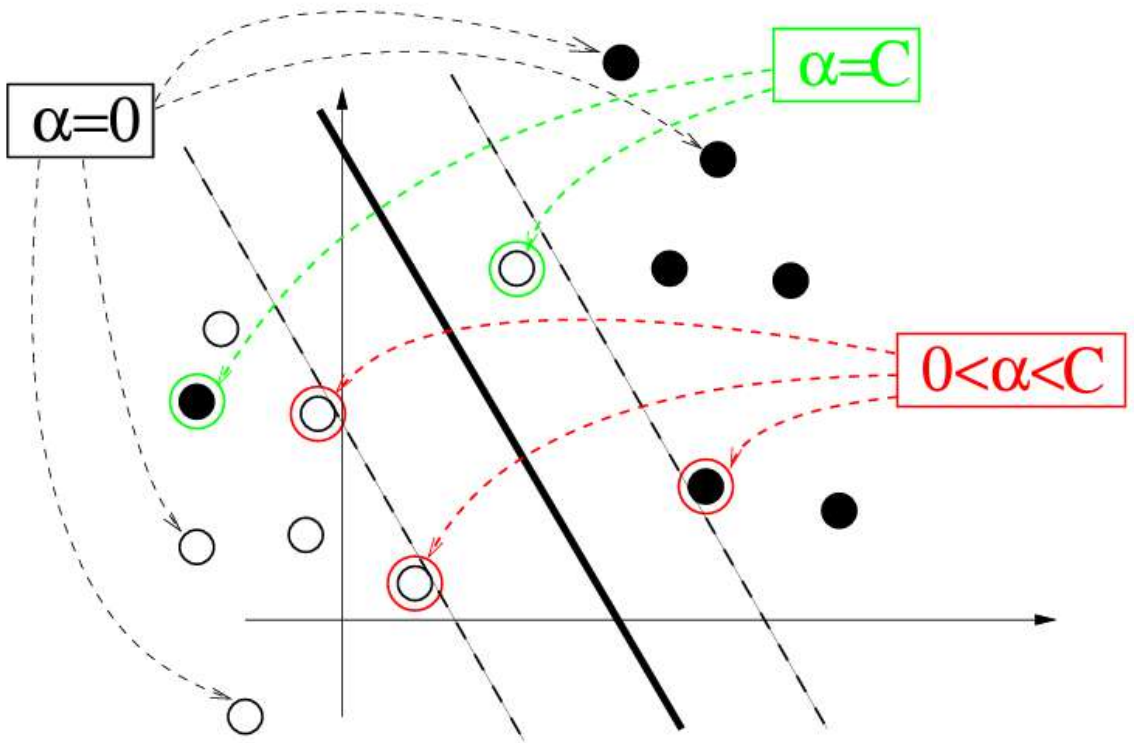
Support vectors

- The solution w can be written as a linear combination of the data
- Only those data for which $0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} y_i \alpha_i = 0$ are counted for the solution

Support vectors

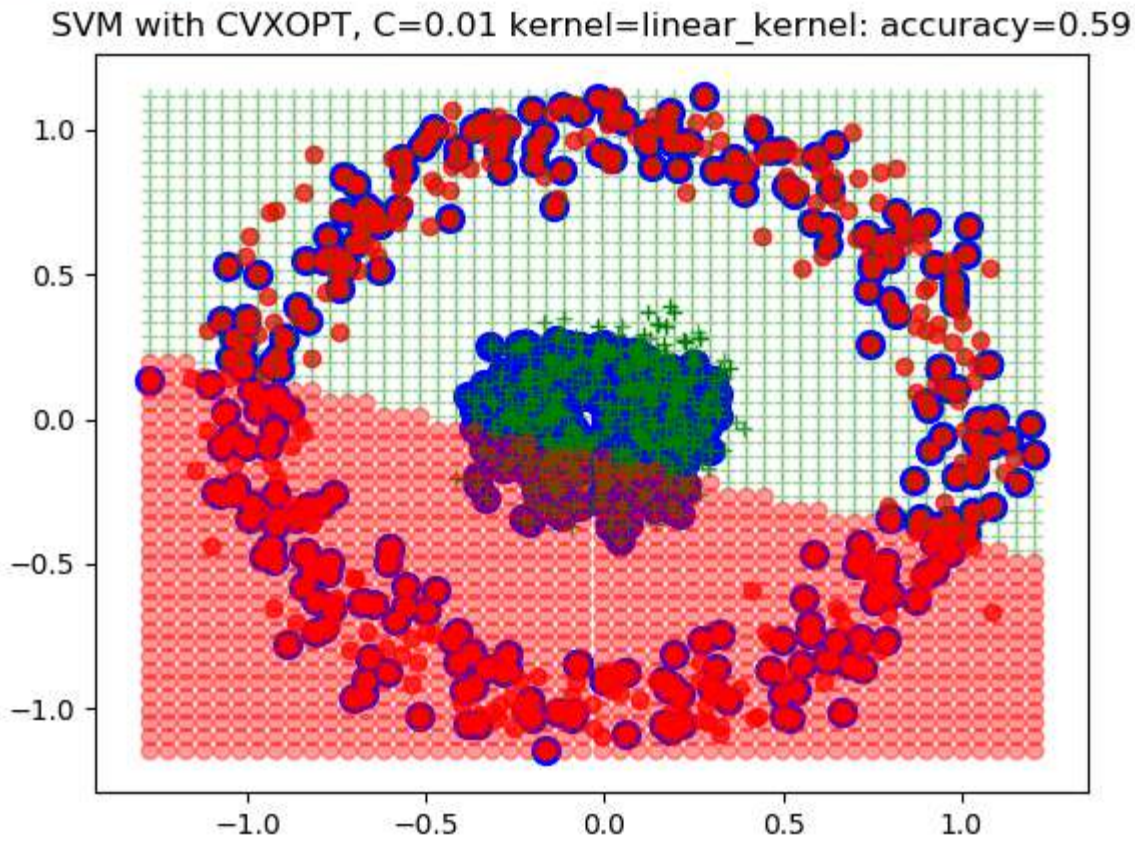
$$\min_{\substack{w \in \mathbb{R}^d \\ \xi \geq 0}} \quad \frac{1}{2} w^\top w + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i$$

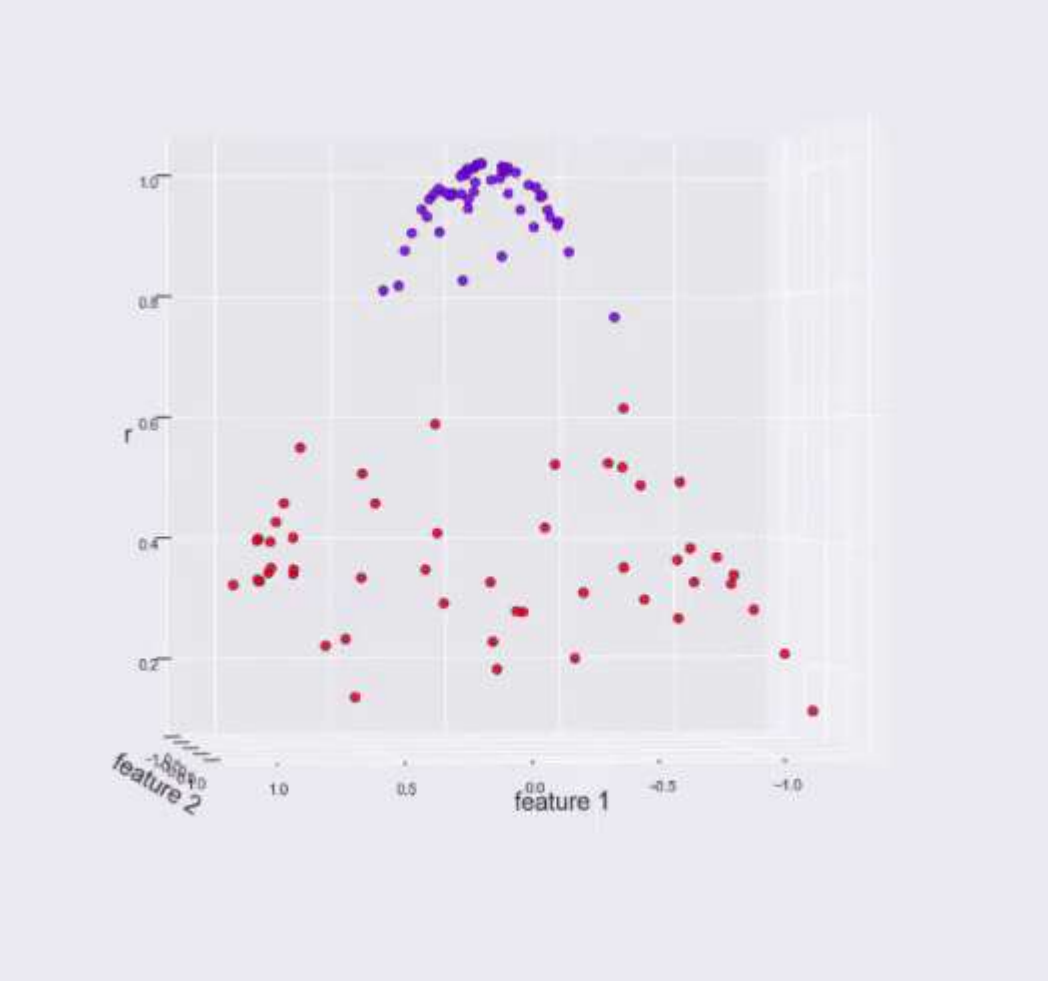


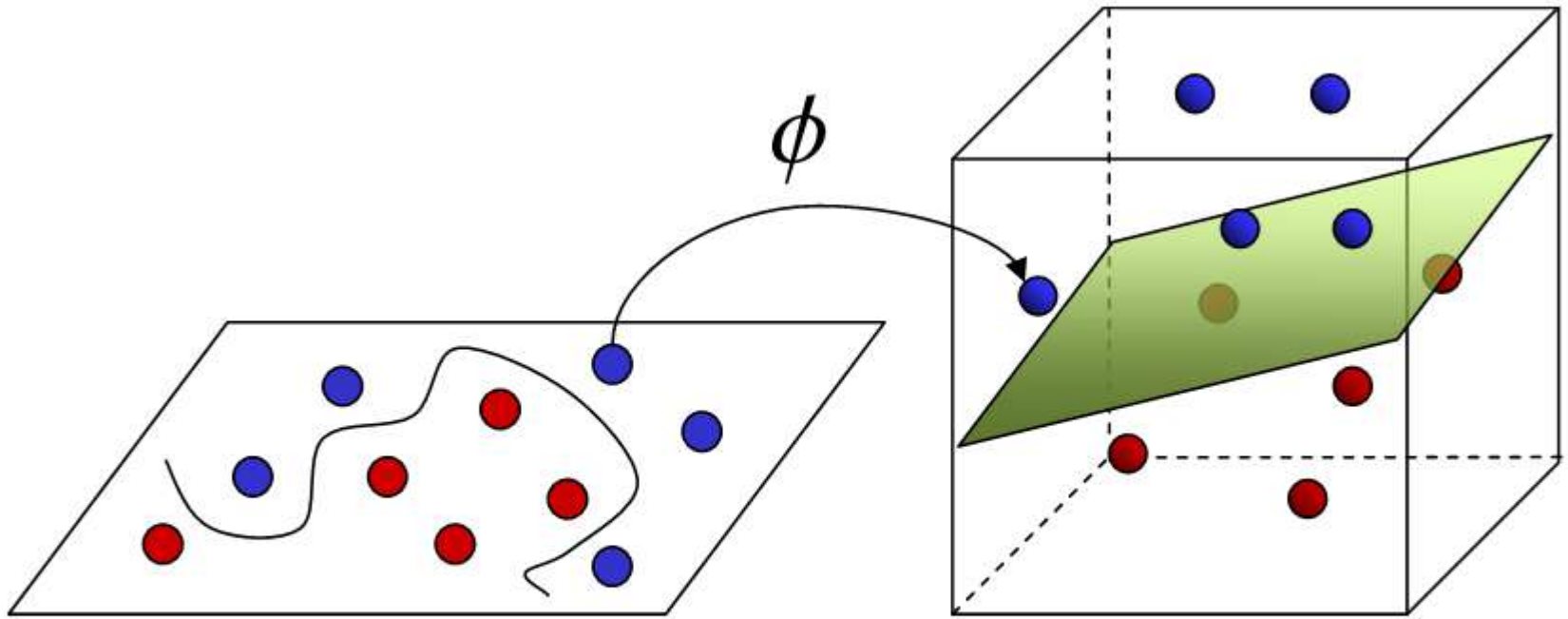
w is given by a linear combination only of the points that satisfy the inequality!

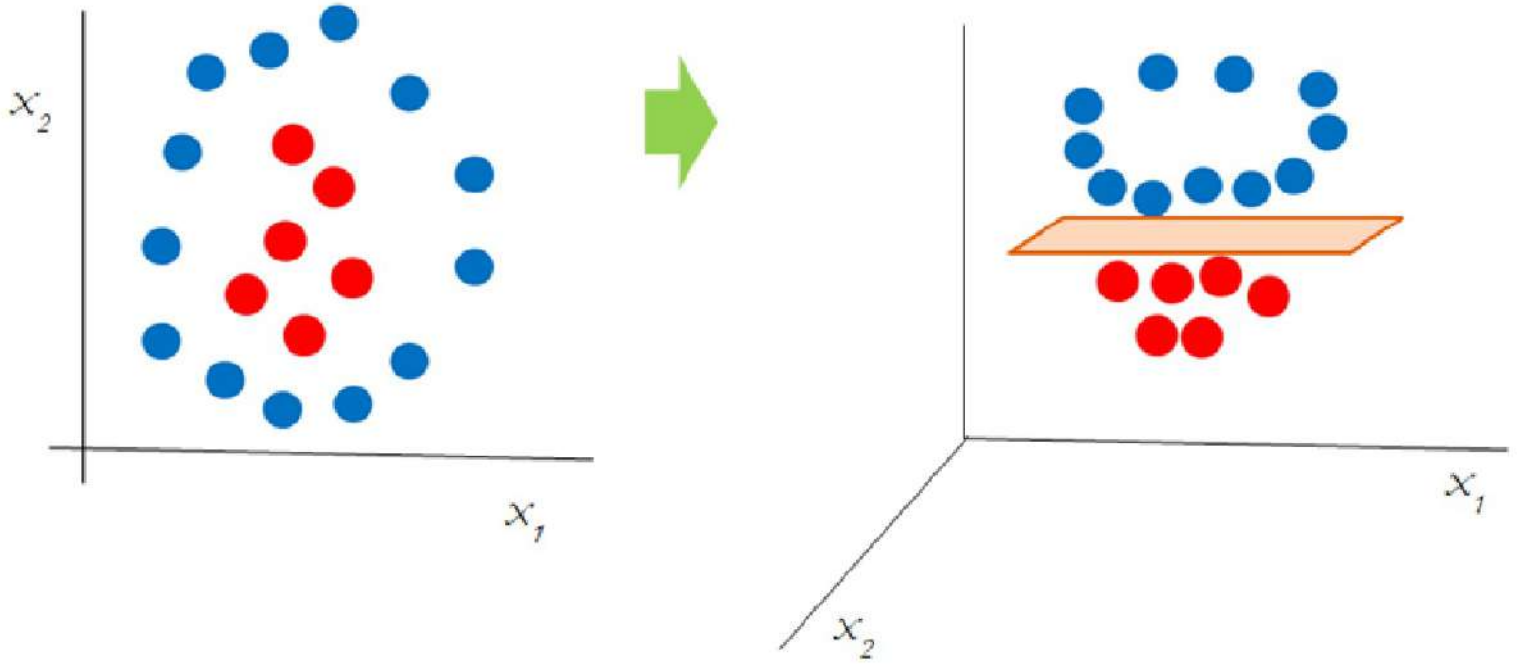
Problem of linear SVMs



What if…



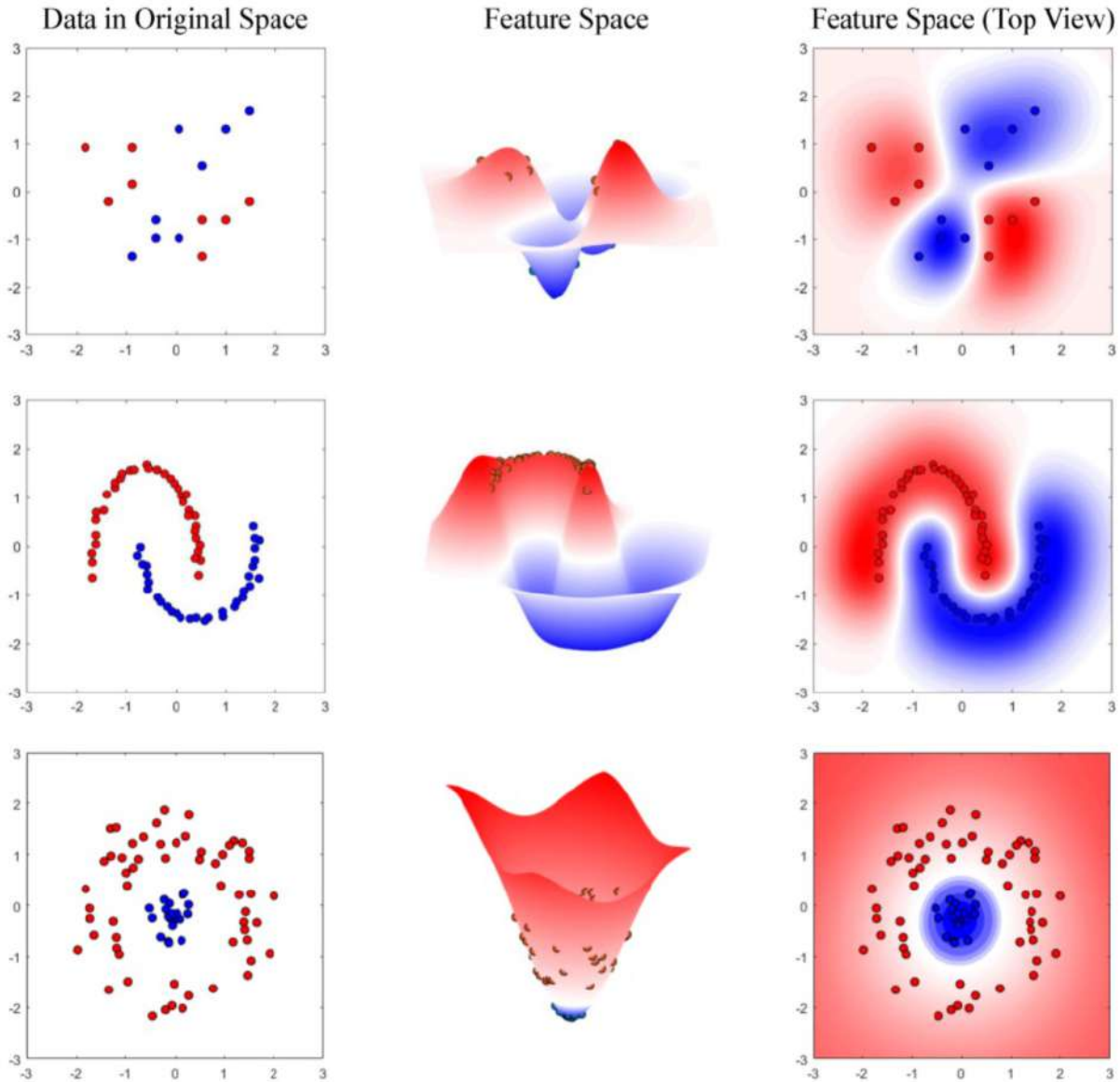

ϕ
Input Space
Feature Space



$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$

$$f(x) = w^T x \quad \rightarrow \quad f(x) = w^T \phi(x)$$

- A problem that is not linearly separable in 2 dim becomes separable in 3 dim.
- We know a good coordinate change; what's the problem with this?



What is the coordinate change here?

Features expansion

We can make linear classifiers non-linear by applying basis function (feature transformations) on the input feature vectors. Formally, for a data vector $\mathbf{x} \in \mathbb{R}^d$, we apply the transformation $\mathbf{x} \rightarrow \phi(\mathbf{x})$ where $\phi(\mathbf{x}) \in \mathbb{R}^D$. Usually $D \gg d$ because we add dimensions that capture non-linear interactions among the original features.

Advantage: It is simple, and your problem stays convex and well behaved. (i.e. you can still use your original gradient descent code, just with the higher dimensional representation)

Disadvantage: $\phi(\mathbf{x})$ might be very high dimensional.

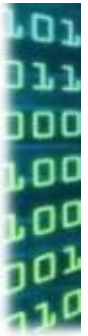
Consider the following example: $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$, and define $\phi(\mathbf{x}) = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \\ x_1 x_2 \\ \vdots \\ x_{d-1} x_d \\ \vdots \\ x_1 x_2 \cdots x_d \end{pmatrix}$.

How many features for a dimension d?

What's the problem of the approach?

- For example, how about using power series expansion?

$$(X, Y, Z) \rightarrow (X, Y, Z, X^2, Y^2, Z^2, XY, YZ, ZX, \ldots)$$



- But, many recent data are high-dimensional.
  e.g. microarray, images, etc...

The above expansion is intractable!

e.g. Up to 2nd moments, 10,000 dimension:

Dim of feature space: ${}_{10000}C_1 + {}_{10000}C_2 = 50{,}005{,}000$

However...

d=1

$$\phi(u).\phi(v) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} . \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = u_1 v_1 + u_2 v_2 = u.v$$

d=2

$$\phi(u).\phi(v) = \begin{pmatrix} u_1^2 \\ u_1 u_2 \\ u_2 u_1 \\ u_2^2 \end{pmatrix} . \begin{pmatrix} v_1^2 \\ v_1 v_2 \\ v_2 v_1 \\ v_2^2 \end{pmatrix} = u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2$$
$$= (u_1 v_1 + u_2 v_2)^2$$
$$= (u.v)^2$$

For any *d (we will skip proof):*

$$\phi(u).\phi(v) = (u.v)^d$$

Polynomials of degree **exactly** *d*

Why the kernel representation is convenient

$$x \in \mathbb{R},\ 0 < x < 1$$

$$\phi(x) = (1, x, x^2, \cdots)$$

$$K(x, y) = \phi(x)^T \phi(y) = \sum_{i=0}^{\infty} (xy)^i = \frac{1}{1 - xy}$$

Example: from linear regression to kernels

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 \quad \Rightarrow \quad \ell(\alpha) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j \mathbf{x}_j^\top \mathbf{x}_i - y_i \right)^2$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i.$$

$$h(\mathbf{x}_t) = \mathbf{w}^\top \mathbf{x}_t = \sum_{j=1}^{n} \alpha_j \mathbf{x}_j^\top \mathbf{x}_t.$$

Substitute the data with non-linear features
Rewrite the Loss with the explicit kernel

$$\ell(w) = \sum_{i=1}^{n} (w^T \phi(x_i) - y_i)^2 \quad \rightarrow \quad \ell(w) = \sum_{i=1}^{n} \left(\sum_{j=1} \alpha_j \phi(x_j)^T \phi(x_i) - y_i \right)^2$$

$$K_{ij} = K(x_i, x_j) = \phi(x_j)^T \phi(x_i)$$

A zoo of kernels: linear, polynomial, gaussian...

A zoo of kernels: linear, polynomial, gaussian

Name	Kernel Function (implicit dot product)	Feature Space (explicit dot product)
Linear	$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$	Same as original input space
Polynomial (v1)	$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z})^d$	All polynomials **of** degree d
Polynomial (v2)	$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + 1)^d$	All polynomials **up to** degree d
Gaussian	$K(\mathbf{x}, \mathbf{z}) = \exp(-\frcolor{\|\mathbf{x} - \mathbf{z}\|_2^2}{2\sigma^2})$	Infinite dimensional space
Hyperbolic Tangent (Sigmoid) Kernel	$K(\mathbf{x}, \mathbf{z}) = \tanh(\alpha\mathbf{x}^T\mathbf{z} + c)$	(With SVM, this is equivalent to a 2-layer neural network)

Polynomial kernel

For $x, z \in [0,1]$ and $0 < \alpha < 1$

$$k(x,z) = \frac{1}{1-\alpha^2 xz}$$

Proof

$$\frac{1}{1-\alpha^2 xz} = \sum_{s=0}^{\infty} (\alpha^2 xz)^s = \Phi(x)^\top \Phi(z)$$

with

$$\Phi(x) = (1, \alpha x, \alpha^2 x^2, \alpha^3 x^3, \ldots)$$

Gaussian kernel

$$K(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2}||x_i - x_j||^2\right) \qquad \sigma = 1/\sqrt{2}$$

$$= \exp\left(-||x_i - x_j||^2\right)$$

$$= \exp\left(-x_i^\top x_i\right) \exp\left(-x_j^\top x_j\right) \exp(2x_i^\top x_j) \qquad \exp(z) = \sum_{n=0}^{\infty} \frac{z^n}{n!}$$

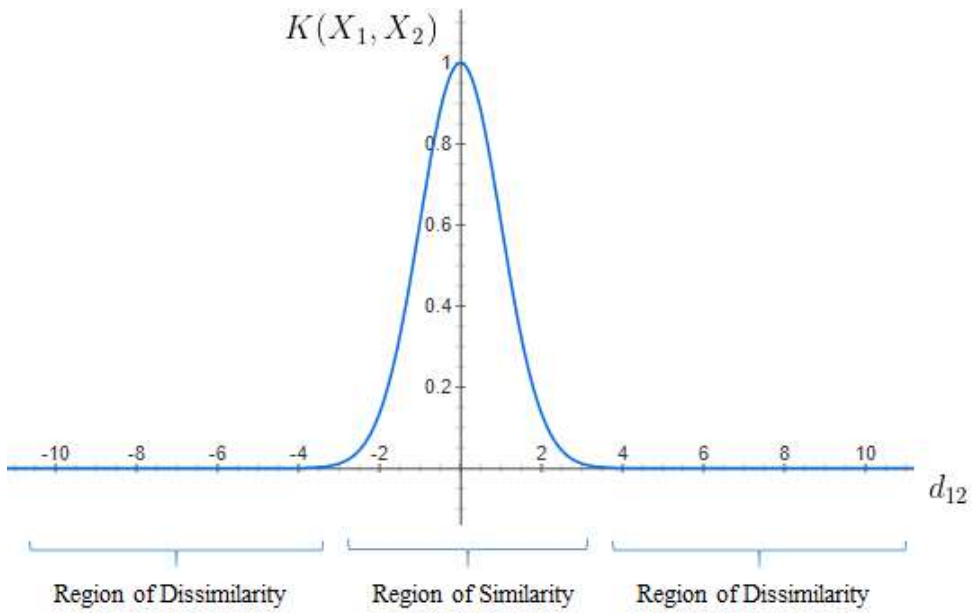
$$= \exp\left(-x_i^\top x_i\right) \exp\left(-x_j^\top x_j\right) \sum_{n=0}^{\infty} \frac{2^n (x_i^\top x_j)^n}{n!}$$

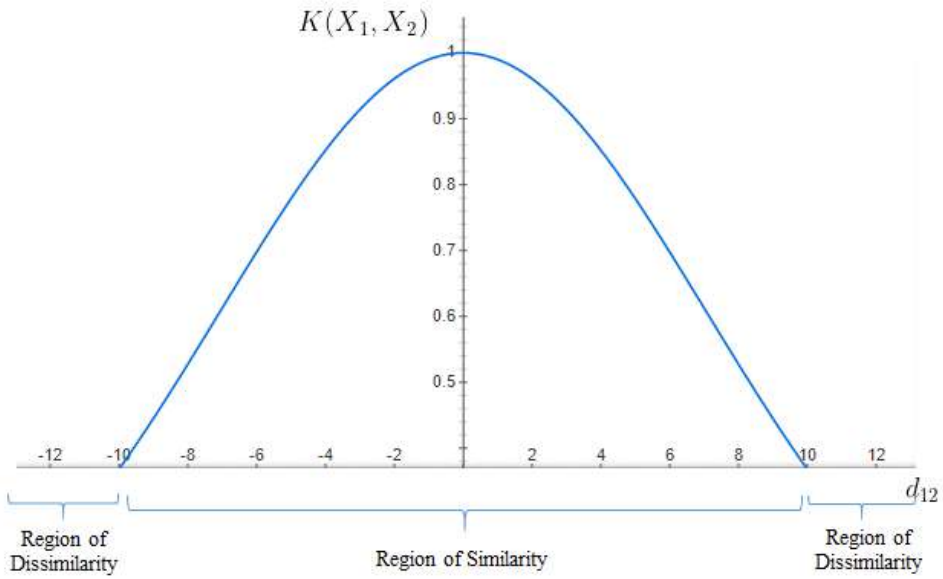
order-n polynomial kernel* $\Phi^n(x)$

$$\Phi^{\text{rbf}} = \exp(-x^\top x)\left[\Phi^1(x)^\top, \Phi^2(x)^\top, \ldots, \Phi^\infty(x)^\top\right]^\top$$

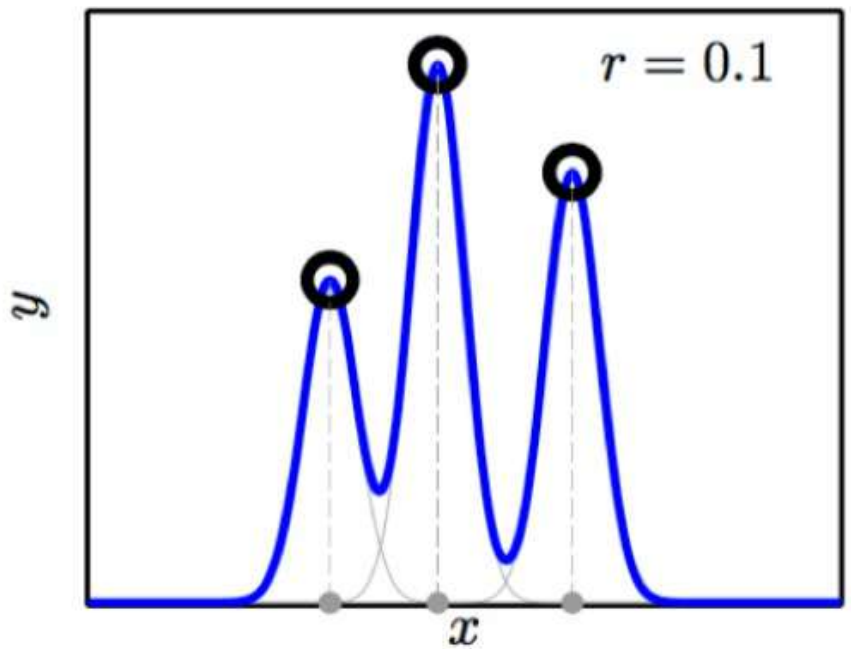
Gaussian kernel is a measure of data similarity

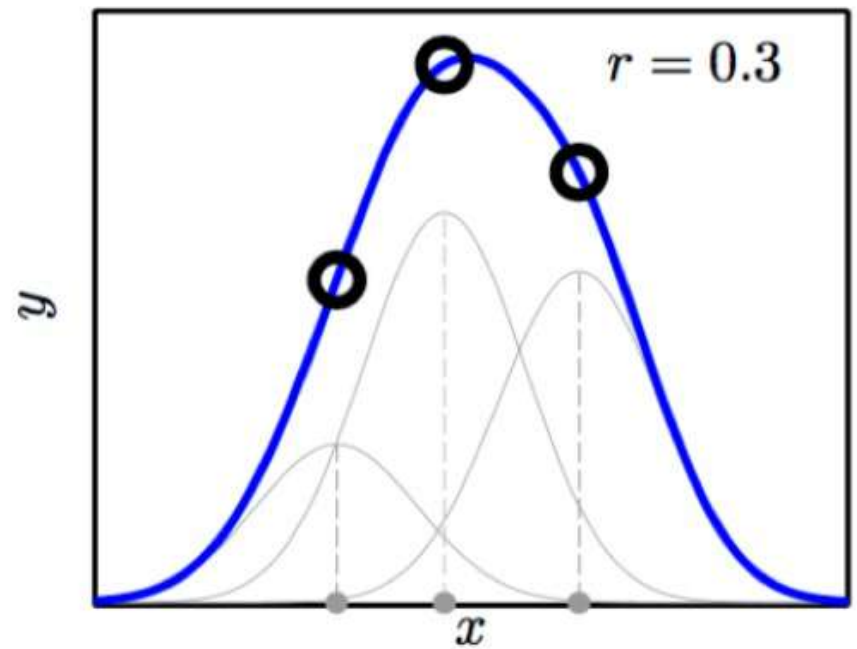
$$K(X_1, X_2) = e^{-\frac{\|X_1 - X_2\|_2^2}{r^2}}$$



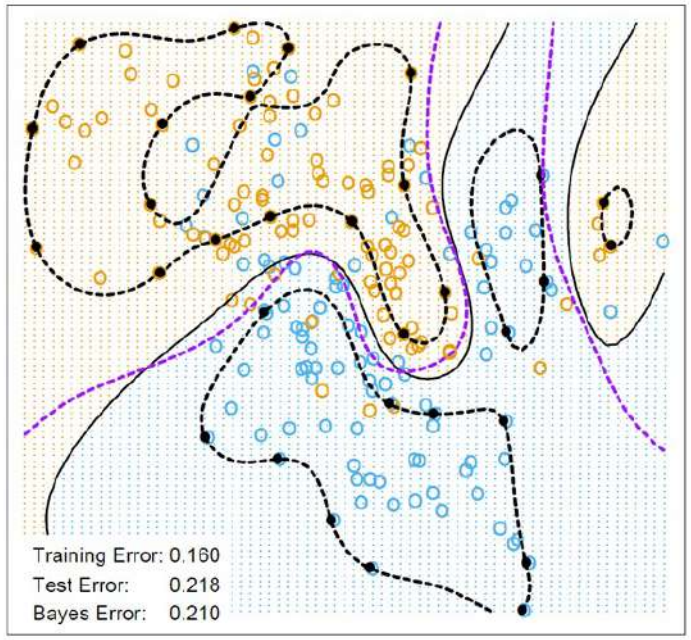


$$f(X) = \sum_i \alpha_i e^{-\frac{\|X - X_i\|_2^2}{r^2}}$$

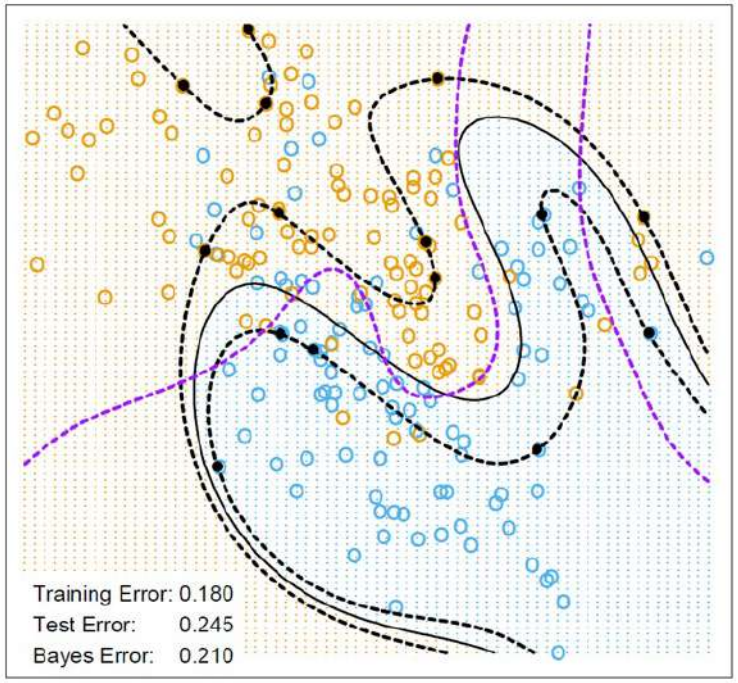




Gaussian Kernel SVM



Degree-4 Polynomial Kernel SVM



Importance of choosing the kernel

Making kernels

Kernels can be :

- Summed together
  - On the same space $k(x, y) = k_1(x, y) + k_2(x, y)$
  - On the tensor space $k(\mathbf{x}, \mathbf{y}) = k_1(x_1, y_1) + k_2(x_2, y_2)$
- Multiplied together
  - On the same space $k(x, y) = k_1(x, y) \times k_2(x, y)$
  - On the tensor space $k(\mathbf{x}, \mathbf{y}) = k_1(x_1, y_1) \times k_2(x_2, y_2)$
- Composed with a function
  - $k(x, y) = k_1(f(x), f(y))$

All these operations will preserve the positive definiteness.

Generating kernels

1. $k(u,v) = \alpha k_1(u,v) + \beta k_2(u,v)$, for $\alpha, \beta \geq 0$

   PROOF.

   Since $\alpha k_1(u,v) = \langle \sqrt{\alpha}\Phi_1(u), \sqrt{\alpha}\Phi_1(v) \rangle$ and $\beta k_2(u,v) = \langle \sqrt{\beta}\Phi_2(u), \sqrt{\beta}\Phi_2(v) \rangle$, then:

$$
\begin{aligned}
k(u,v) &= \alpha k_1(u,v) + \beta k_2(u,v) \\
&= \langle \sqrt{\alpha}\Phi_1(u), \sqrt{\alpha}\Phi_1(v) \rangle + \langle \sqrt{\beta}\Phi_2(u), \sqrt{\beta}\Phi_2(v) \rangle \\
&= \langle [\sqrt{\alpha}\Phi_1(u)\ \sqrt{\beta}\Phi_2(u)], [\sqrt{\alpha}\Phi_1(v)\ \sqrt{\beta}\Phi_2(v)] \rangle
\end{aligned}
$$

   and we see that $k(u,v)$ can be expressed as an inner product

Generating kernels

2. $k(u, v) = k_1(u, v)k_2(u, v)$

   PROOF.

   Note that the gram matrix K for k is the Hadamard product (or element-by-element product) of K_1 and K_2 ($K = K_1 \odot K_2$). Suppose that K_1 and K_2 are covariance matrices of $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_n)$ respectively. Then K is simply the covariance matrix of $(X_1Y_1, \ldots, X_nY_n)$, implying that it is symmetric and positive definite. $\square$

Kernels

Classic examples

- linear $k(x, \bar{x}) = x^\top \bar{x}$
- polynomial $k(x, \bar{x}) = (x^\top \bar{x} + 1)^s$
- Gaussian $k(x, \bar{x}) = e^{-\|x-\bar{x}\|^2 \gamma}$

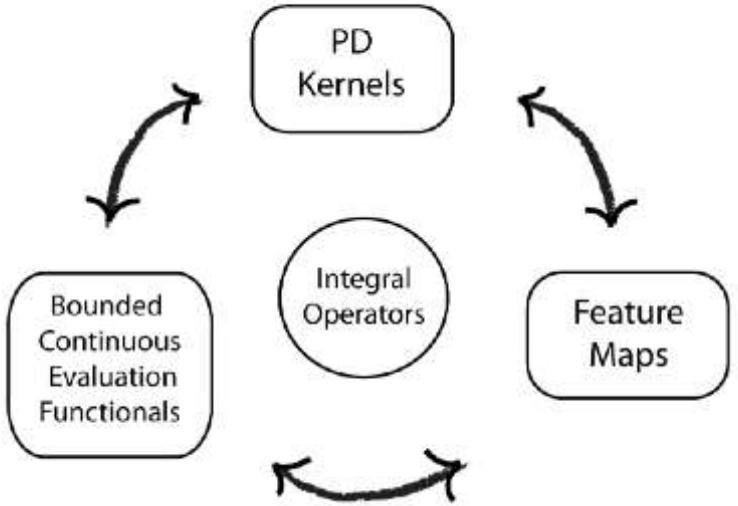
But one can consider

- kernels on probability distributions
- kernels on strings
- kernels on functions
- kernels on groups
- kernels graphs
- ...

It is natural to think of a kernel as a measure of similarity.

Class 3: Math of kernel methods

- Aronszajn theorem (positive definite maps and features)
- Reproducing property. Reproducing kernel Hilbert spaces (RKHS)
- Pointwise continuity



Some slides from : Julien Mairal

https://members.cbio.mines-paristech.fr/~jvert/svn/kernelcourse/slides/master2017/master2017.pdf

Can we start from the kernel instead of features? What defines a good kernel?

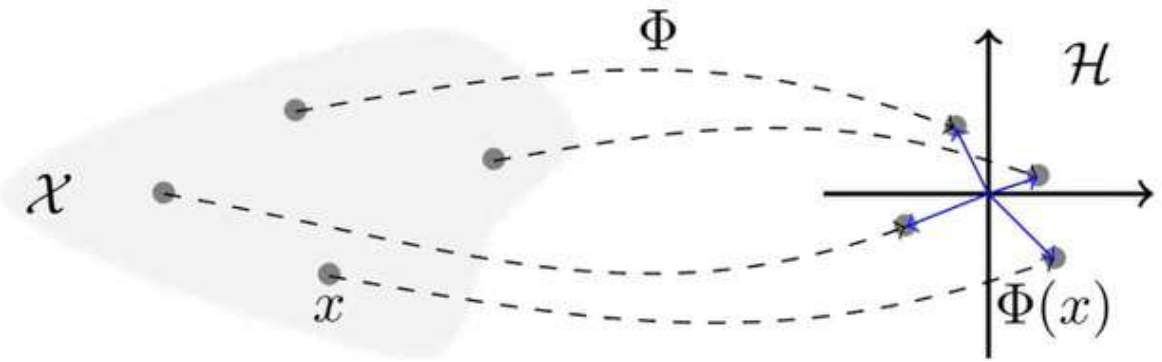
Theorem (Aronszajn, 1950)

K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

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

such that, for any $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$:

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}} .$$



Def1: Hilbert space

- An inner product on an $\mathbb{R}$-vector space $\mathcal{H}$ is a mapping $(f, g) \mapsto \langle f, g \rangle_{\mathcal{H}}$ from $\mathcal{H}^2$ to $\mathbb{R}$ that is bilinear, symmetric and such that $\langle f, f \rangle_{\mathcal{H}} > 0$ for all $f \in \mathcal{H} \backslash \{0\}$.
- A vector space endowed with an inner product is called pre-Hilbert. It is endowed with a norm defined as $\| f \|_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}^{\frac{1}{2}}$.
- A Cauchy sequence $(f_n)_{n \geq 0}$ is a sequence whose elements become progressively arbitrarily close to each other:

$$\lim_{N \to +\infty} \sup_{n,m \geq N} \|f_n - f_m\|_{\mathcal{H}} = 0.$$

- A Hilbert space is a pre-Hilbert space complete for the norm $\|.\|_{\mathcal{H}}$. That is, any Cauchy sequence in $\mathcal{H}$ converges in $\mathcal{H}$.

Completeness is necessary to keep "good" convergence properties of Euclidean spaces in an infinite-dimensional context.

Def2: Positive definite kernel

Definition. Ω: set. $k: \Omega \times \Omega \to \mathbf{R}$ is a positive definite kernel if

1) (symmetry) $k(x, y) = k(y, x)$

2) (positivity) for arbitrary $x_1, \ldots, x_n \in \Omega$

$$\begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}$$ is positive semidefinite,

(Gram matrix)

i.e., $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$ for any $c_i \in \mathbf{R}$ $\qquad c^T K c$

Spectral Theorem

For Finite-Dimensional Spaces (Matrices)

Given a **normal matrix** A (such as a symmetric matrix in the real case or a Hermitian matrix in the complex case), the spectral theorem states that A can be expressed as:

$$A = \sum_{i=1}^{n} \lambda_i P_i,$$

where:

- λ_i are the eigenvalues of A.
- P_i are **orthogonal projection matrices** onto the eigenspaces corresponding to each eigenvalue λ_i. These projection matrices satisfy:
  - $P_i P_j = \delta_{ij} P_i$ (orthogonality for $i \neq j$),
  - $P_i^2 = P_i$ (idempotence),
  - $\sum_{i=1}^{n} P_i = I$ (they sum to the identity matrix).

The matrix A can therefore be "decomposed" into a sum of its eigenvalues scaled by their corresponding projectors.

pd $\Longrightarrow$ feats

- Assume $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is finite of size N.
- Any p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is entirely defined by the $N \times N$ symmetric positive semidefinite matrix $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$.
- It can therefore be diagonalized on an orthonormal basis of eigenvectors $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$, with non-negative eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_N$, i.e.,

Spectral theorem

Sum over the eigenvectors
Fixed coordinates

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{l=1}^{N} \lambda_l \mathbf{u}_l \mathbf{u}_l^\top\right]_{ij} = \sum_{l=1}^{N} \lambda_l [\mathbf{u}_l]_i [\mathbf{u}_l]_j = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathbb{R}^N},$$

with

$$\Phi(\mathbf{x}_i) = \begin{pmatrix} \sqrt{\lambda_1}[\mathbf{u}_1]_i \\ \vdots \\ \sqrt{\lambda_N}[\mathbf{u}_N]_i \end{pmatrix}. \quad \square$$

$$c^T K c$$

pd $\Longleftarrow$ feats

- $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathbb{R}^d} = \langle \Phi(\mathbf{x}'), \Phi(\mathbf{x}) \rangle_{\mathbb{R}^d}$
- $\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathbb{R}^d} = \| \sum_{i=1}^{N} a_i \Phi(\mathbf{x}_i) \|_{\mathbb{R}^d}^2 \geq 0$

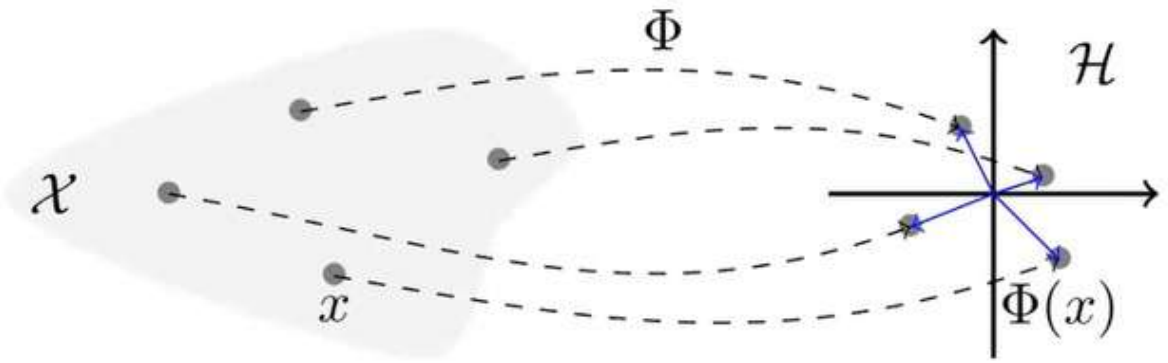
Theorem (Aronszajn, 1950)

K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

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

such that, for any $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$:

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}} .$$



The mapping is not unique!

Example 37. Consider $\mathcal{X} = \mathbb{R}$, and

$$k(x,y) = xy = \begin{bmatrix} \frac{x}{\sqrt{2}} & \frac{x}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{y}{\sqrt{2}} \\ \frac{y}{\sqrt{2}} \end{bmatrix},$$

where we defined the feature maps $\phi(x) = x$ and $\tilde{\phi}(x) = \begin{bmatrix} \frac{x}{\sqrt{2}} & \frac{x}{\sqrt{2}} \end{bmatrix}$, and where the feature spaces are respectively, $\mathcal{H} = \mathbb{R}$, and $\tilde{\mathcal{H}} = \mathbb{R}^2$.

What is a good definition of the restriction of the H space?

The reproducing property guarantees the uniqueness

Reproducing kernels Hilbert spaces (RKHS)

$$f(x) = \int f(y)K(x,y)dy$$

Definition

Let $\mathcal{X}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a class of functions forming a (real) Hilbert space with inner product $\langle .,. \rangle_{\mathcal{H}}$. The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ is called a reproducing kernel (r.k.) of $\mathcal{H}$ if

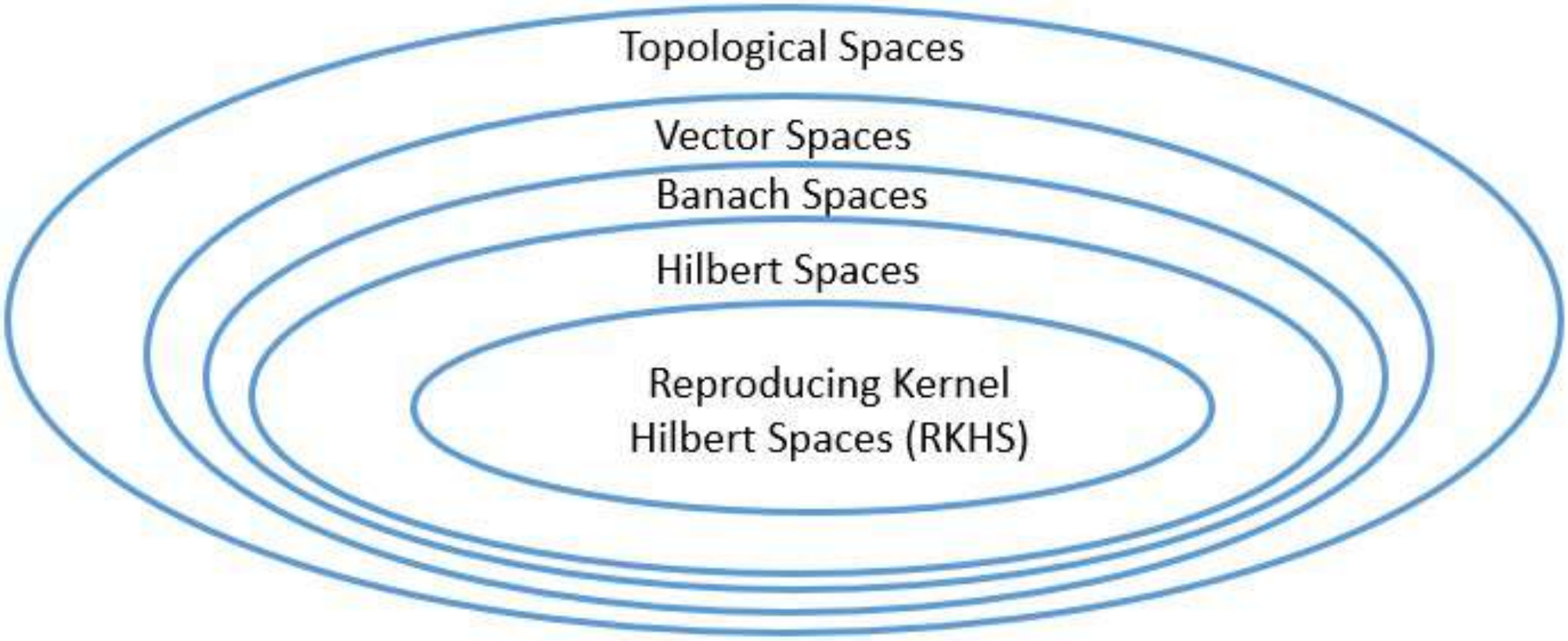
1. $\mathcal{H}$ contains all functions of the form

$$\forall \mathbf{x} \in \mathcal{X}, \quad K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t}) .$$

2. For every $\mathbf{x} \in \mathcal{X}$ and $f \in \mathcal{H}$ the reproducing property holds:

$$f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}} .$$

If a r.k. exists, then $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS).



Kernels and pd are one to one if the Hilbert space where they are defined has the *reproducing property*.

Let's see why.

First a definition (from matrices to operators):

Fix a symmetric function $k : \mathcal{X}^2 \to \mathbb{R}$ on a compact set $\mathcal{X} \subset \mathbb{R}^d$, and consider the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ defined as

$$T_k f(\cdot) = \int_{\mathcal{X}} k(\cdot, x) f(x)\, dx.$$

We say T_k is positive semidefinite if, for all $f \in L_2(\mathcal{X})$, $\langle f, T_k f \rangle_{L_2(\mathcal{X})} \geq 0$, that is,

$$\int_{\mathcal{X}^2} k(u, v) f(u) f(v)\, du\, dv \geq 0.$$

Mercer's Theorem

Theorem: If k is continuous and T_k is positive semidefinite, then T_k has eigenfunctions $\psi_i \in L_2(\mathcal{X})$ (say $\|\psi_i\|_{L_2} = 1$) with eigenvalues $\lambda_i \geq 0$, and for all $u, v \in \mathcal{X}$, we can write

$$k(u, v) = \sum_{i=1}^{\infty} \lambda_i \psi_i(u) \psi_i(v).$$

Furthermore, this series converges uniformly.

- A positive semidefinite operator defines a kernel
- Eigenfunctions play the role of features/basis (not normalized)

We skip the demonstration (Spectral Theorem)

Reproducing property

Under certain conditions (Mercer's theorem and extensions), we can write

$$k(x,x') = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(x'), \qquad \int_{\mathcal{X}} e_i(x) e_j(x) d\mu(x) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

where this sum is guaranteed to converge whatever the x and x'.

Infinite dimensional feature map: $\phi(x) = \begin{bmatrix} \vdots \\ \sqrt{\lambda_i} e_i(x) \\ \vdots \end{bmatrix} \in \ell_2.$

Reproducing property when we have pd: more concrete

$$f(x) = \sum_i \alpha_i e_i(x)$$

$$K(x, x') = \sum_j \lambda_j e_j(x) e_j(x')$$

$$\begin{aligned}
\langle f, K_x \rangle &= \int K(x, x') f(x') dx' \\
&= \int \Big(\sum_i \lambda_i e_i(x) e_i(x') \Big) f(x') dx' \\
&= \sum_i \Big(\int \sqrt{\lambda_i} e_i(x') f(x') dx' \Big) \sqrt{\lambda_i} e_i(x) \\
&= \sum_i \alpha_i e_i(x) = f(x)
\end{aligned}$$

How does the reproducing property guarantees uniqueness?
We divide the demonstration into steps

$$\text{Kernel} \leftrightarrow \text{PD} \leftrightarrow \text{r.k.} \leftrightarrow \text{RKHS}$$

Kernel ↔ PD ↔ r.k. ↔ RKHS

Mercer theorem

PD ↔ r.k. ↔ RKHS

A function $k : X \times X \to \mathbb{R}$ defines a positive definite kernel if and only if there exists a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ of functions on X such that:

1. For every $x \in X$, the function $k_x(\cdot) = k(x, \cdot)$ belongs to $\mathcal{H}$.
2. The reproducing property holds: for every $f \in \mathcal{H}$ and every $x \in X$,

$$f(x) = \langle f, k_x \rangle_{\mathcal{H}}.$$

(⇒) If k is a positive definite kernel, then it induces an RKHS with the reproducing property.

Construct an inner product on the space of functions generated by the kernel k and show that this construction leads to an RKHS that inherently satisfies the reproducing property.

1. **Construct the RKHS:** Given a positive definite kernel $k : X \times X \to \mathbb{R}$, we define the linear space $\mathcal{H}_0$ consisting of finite linear combinations of kernel functions:

$$\mathcal{H}_0 = \left\{ f = \sum_{i=1}^{n} c_i k_{x_i} : c_i \in \mathbb{R}, x_i \in X, n \in \mathbb{N} \right\}.$$

Define an inner product on this space by:

$$\left\langle \sum_{i=1}^{n} c_i k_{x_i}, \sum_{j=1}^{m} d_j k_{y_j} \right\rangle_{\mathcal{H}_0} = \sum_{i=1}^{n} \sum_{j=1}^{m} c_i d_j k(x_i, y_j).$$

2. **Completion to an RKHS:** The space $\mathcal{H}_0$ can be completed to form a Hilbert space $\mathcal{H}$. By construction, every function in this Hilbert space can be expressed (or approximated arbitrarily closely) as a linear combination of the kernel functions k_x.

3. **Reproducing Property:** For each $x \in X$ and any function $f = \sum_{i=1}^{n} c_i k_{x_i} \in \mathcal{H}_0$,

$$f(x) = \sum_{i=1}^{n} c_i k(x_i, x) = \left\langle \sum_{i=1}^{n} c_i k_{x_i}, k_x \right\rangle_{\mathcal{H}_0}.$$

Since $k_x \in \mathcal{H}$ by construction, this property extends to the entire RKHS $\mathcal{H}$. Therefore, k satisfies the reproducing property.

The reproducing property **emerges** naturally from how the inner product was defined, due to the kernel's structure. This is the key of the proof: defining the space and inner product using the kernel, then completing this space to obtain the RKHS where the reproducing property holds.

(⇐) If there exists an RKHS with the reproducing property, then k is a positive definite kernel.

1. **Existence of the Kernel Function:** Suppose $\mathcal{H}$ is an RKHS of functions on X with the reproducing property. For every $x \in X$, there exists a function $k_x \in \mathcal{H}$ such that for any $f \in \mathcal{H}$,

$$f(x) = \langle f, k_x \rangle_{\mathcal{H}}.$$

2. **Define the Kernel:** Define the function $k : X \times X \to \mathbb{R}$ by

$$k(x, y) = \langle k_y, k_x \rangle_{\mathcal{H}}.$$

3. **Positive Definiteness:** To show that k is a positive definite kernel, take any finite set of points $\{x_1, x_2, \ldots, x_n\} \subset X$ and any real coefficients $c_1, c_2, \ldots, c_n$. Consider the linear combination in the RKHS:

$$f = \sum_{i=1}^{n} c_i k_{x_i}.$$

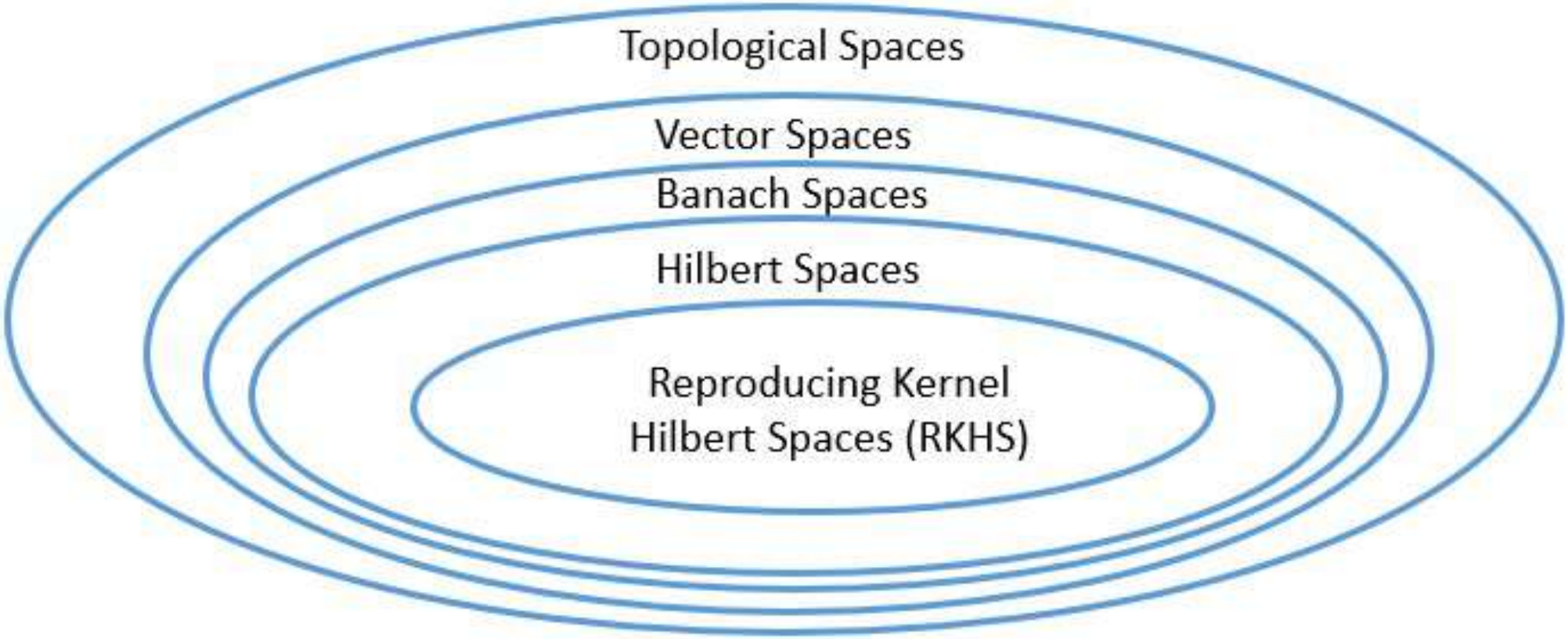
Then, the norm squared of f in the RKHS is given by:

$$\|f\|_{\mathcal{H}}^2 = \left\langle \sum_{i=1}^{n} c_i k_{x_i}, \sum_{j=1}^{n} c_j k_{x_j} \right\rangle_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \langle k_{x_i}, k_{x_j} \rangle_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j).$$

Since $\|f\|_{\mathcal{H}}^2 \geq 0$ (as it is the squared norm of a function in a Hilbert space), it follows that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0. \qquad c^T K c$$

Therefore, k is a positive definite kernel.



Functions that are in the span of the kernel (by the reproducing property) define the RKHS

Uniqueness of RKHS and RK

Theorem

- If $\mathcal{H}$ is a RKHS, then it has a unique r.k.
- Conversely, a function K can be the r.k. of at most one RKHS.

$\Longrightarrow$

If a r.k. exists then it is unique

Let K and K' be two r.k. of a RKHS $\mathcal{H}$. Then for any $\mathbf{x} \in \mathcal{X}$:

$$\begin{aligned} \| K_{\mathbf{x}} - K'_{\mathbf{x}} \|^2_{\mathcal{H}} &= \langle K_{\mathbf{x}} - K'_{\mathbf{x}}, K_{\mathbf{x}} - K'_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= \langle K_{\mathbf{x}} - K'_{\mathbf{x}}, K_{\mathbf{x}} \rangle_{\mathcal{H}} - \langle K_{\mathbf{x}} - K'_{\mathbf{x}}, K'_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= K_{\mathbf{x}}(\mathbf{x}) - K'_{\mathbf{x}}(\mathbf{x}) - K_{\mathbf{x}}(\mathbf{x}) + K'_{\mathbf{x}}(\mathbf{x}) \\ &= 0 . \end{aligned}$$

This shows that $K_{\mathbf{x}} = K'_{\mathbf{x}}$ as functions, i.e., $K_{\mathbf{x}}(\mathbf{y}) = K'_{\mathbf{x}}(\mathbf{y})$ for any $\mathbf{y} \in \mathcal{X}$. In other words, K=K'. □

Another way of characterizing RKHS is:
reproducing property and pointwise continuity

If $\mathcal{H}$ is a RKHS then $f \mapsto f(\mathbf{x})$ is continuous

If a r.k. K exists, then for any $(\mathbf{x}, f) \in \mathcal{X} \times \mathcal{H}$:

$$
\begin{aligned}
|f(\mathbf{x})| &= |\langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}}| \\
&\leq \|f\|_{\mathcal{H}} . \|K_{\mathbf{x}}\|_{\mathcal{H}} \\
&\leq \|f\|_{\mathcal{H}} . K(\mathbf{x}, \mathbf{x})^{\frac{1}{2}} ,
\end{aligned}
$$

$$|\langle \mathbf{u}, \mathbf{v} \rangle|^2 \leq \langle \mathbf{u}, \mathbf{u} \rangle \cdot \langle \mathbf{v}, \mathbf{v} \rangle$$

(Cauchy-Schwarz)

because $\|K_{\mathbf{x}}\|^2_{\mathcal{H}} = \langle K_{\mathbf{x}}, K_{\mathbf{x}} \rangle_{\mathcal{H}} = K(\mathbf{x}, \mathbf{x})$. Therefore $f \in \mathcal{H} \mapsto f(\mathbf{x}) \in \mathbb{R}$ is a continuous linear mapping. □

Reproducing property implies the fact that is pointwise continuous

Reproducing property and pointwise continuity

$\Longleftarrow$

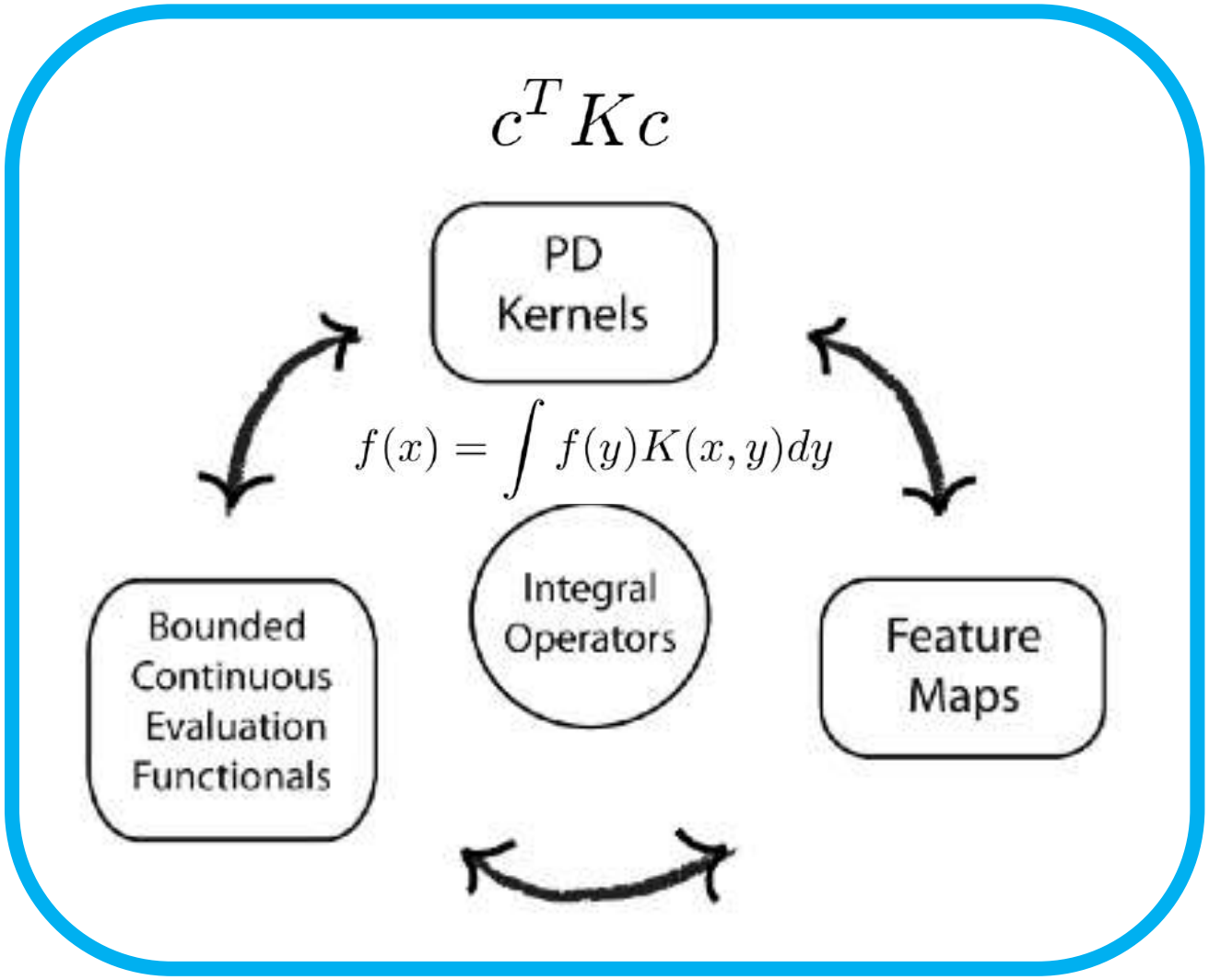
If $f \mapsto f(\mathbf{x})$ is continuous then $\mathcal{H}$ is a RKHS

Conversely, let us assume that for any $\mathbf{x} \in \mathcal{X}$ the linear form $f \in \mathcal{H} \mapsto f(\mathbf{x})$ is continuous.
Then by Riesz representation theorem (general property of Hilbert spaces) there exists a unique $g_{\mathbf{x}} \in \mathcal{H}$ such that:

$$f(\mathbf{x}) = \langle f, g_{\mathbf{x}} \rangle_{\mathcal{H}} .$$

The function $K(\mathbf{x}, \mathbf{y}) = g_{\mathbf{x}}(\mathbf{y})$ is then a r.k. for $\mathcal{H}$. $\square$

The picture is:



Summarizing

- Feature maps define kernels and PD operators but the relation is not one to one
- Feature maps defined by the reproducing kernel Hilbert spaces property are one to one with kernels and pd
- To have the reproducing property we consider operators that satisfy the Mercer theorem
- Pointwise continuity is one to one with RKHS

Class 4

- Representer theorem
- Regularization
- Examples: Kernel ridge regression, Kernel SVM, Kernel PCA

Summarizing

- Feature maps define kernels and PD but the relation is not one to one
- Feature maps defined by the reproducing kernel Hilbert spaces property are one to one with kernels and pd
- To have the reproducing property we consider operators that satisfy the Mercer theorem
- Pointwise continuity is one to one with RKHS

Why do we care?

(Representer Theorem): *The optimal solution to any vector valued function learning problem of the form.*

$$f^* = \text{argmin}_{f \in \mathcal{H}_K} L(f(\mathbf{x}_1) \dots f(\mathbf{x}_l)) + \lambda \|f\|^2_{\mathcal{H}_K}$$

is a sum of matrix-vector products of the form

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} K(\mathbf{x}, \mathbf{x}_i)\boldsymbol{\alpha}_i$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^n, \quad i = 1 \dots l$, *L is an arbitrary loss function (which can also be an indicator function encoding arbitrary constraints) and* $\lambda > 0$ *is a regularization parameter.*

- **We reduced an infinite dimensional non-linear optimization to a linear finite one!!!**
- **Choosing a kernel we choose a regularization!!!**

Motivation

- An RKHS is a space of (potentially nonlinear) functions, and $\| f \|_{\mathcal{H}}$ measures the smoothness of f.
- Given a set of data $(\mathbf{x}_i \in \mathcal{X}, y_i \in \mathbb{R})_{i=1,\dots,n}$, a natural way to estimate a regression function $f : \mathcal{X} \to \mathbb{R}$ is to solve something like:

$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \|f\|_{\mathcal{H}}^2}_{\text{regularization}} . \tag{1}$$

  for a loss function ℓ such as $\ell(y, t) = (y - t)^2$.
- How to solve in practice this problem, potentially in infinite dimension?

Proof

f in the subspace spanned by the kernel → The loss is minimized

Suppose we project f onto the subspace:

$$\text{span}\{k(x_i, \cdot)\colon\ 1 \leq i \leq n\}$$

obtaining f_s (the component along the subspace) and $f_\perp$ (the component perpendicular to the subspace). We have:

$$f = f_s + f_\perp \Rightarrow ||f||^2 = ||f_s||^2 + ||f_\perp||^2 \geq ||f_s||^2$$

Since Ω is non-decreasing,

$$\Omega(||f||_H^2) \geq \Omega(||f_s||_H^2)$$

implying that $\Omega(\cdots)$ is minimized if f lies in the subspace. Furthermore, since the kernel k has the reproducing property, we have:

$$f(x_i) = \langle f, k(x_i, \cdot)\rangle = \langle f_s, k(x_i, \cdot)\rangle + \langle f_\perp, k(x_i, \cdot)\rangle = \langle f_s, k(x_i, \cdot)\rangle = f_s(x_i)$$

Implying that:

$$L(f(x_1), \ldots, f(x_n)) = L(f_s(x_1), \ldots, f_s(x_n))$$

Hence, $L(\cdots)$ depends only on the component of f lying in the subspace: $\text{span}\{k(x_i, \cdot)\colon 1 \leq i \leq n\}$, and $\Omega(\cdots)$ is minimized if f lies in that subspace. Hence, $J(f)$ is minimized if f lies in that subspace, and we can express the minimizer as:

$$f^*(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$$

Practical use of the representer theorem (1/2)

- When the representer theorem holds, we know that we can look for a solution of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) , \quad \text{for some } \boldsymbol{\alpha} \in \mathbb{R}^n.$$

- For any $j = 1, \ldots, n$, we have

$$f(\mathbf{x}_j) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) = [\mathbf{K}\boldsymbol{\alpha}]_j.$$

- Furthermore,

$$\| f \|_{\mathcal{H}}^2 = \left\| \sum_{i=1}^{n} \alpha_i K_{\mathbf{x}_i} \right\|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}.$$

Practical use of the representer theorem (2/2)

- Therefore, a problem of the form

$$\min_{f \in \mathcal{H}} \Psi \left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right), \| f \|_{\mathcal{H}}^2 \right)$$

is equivalent to the following n-dimensional optimization problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \Psi \left([\mathbf{K}\boldsymbol{\alpha}]_1, \cdots, [\mathbf{K}\boldsymbol{\alpha}]_n, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right).$$

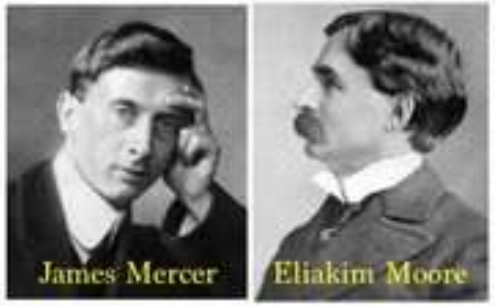
Summary

Feature map: $\varphi : x \in \mathbb{R}^d \longmapsto \mathcal{H}$ Hilbert space.

Reproducing Hilbert space: $f : \mathbb{R}^d \to \mathbb{R}$

$$\|f\|_R = \min \{\|h\|_{\mathcal{H}} \; ; \; h \in \mathcal{H}, f = \langle h, \varphi(\cdot)\rangle_{\mathcal{H}}\}$$

Reproducing kernel: $k(x,y) \overset{\text{def.}}{=} \langle \varphi(x), \varphi(y)\rangle_{\mathcal{H}}$



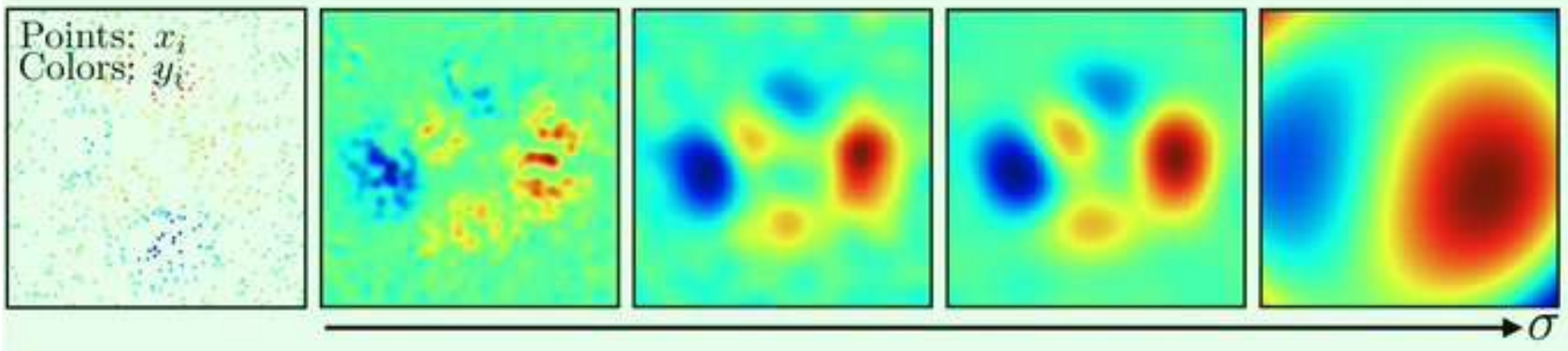
Representer theorem: the solution of $\min_{f:\mathbb{R}^d \to \mathbb{R}} \mathcal{E}((f(x_i))_i) + \|f\|_R^2$

satisfies $f(x) = \sum_i w_i K(x_i, x)$.

Example: regression

$$\mathcal{E}((f_i)_i) = \frac{1}{n}\sum_{i=1}^n |f_i - y_i|^2$$

$$k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$



- Kernel Ridge Regression
- Kernel SVM

Kernel Ridge Regression (KRR)

- Let us now consider a RKHS $\mathcal{H}$, associated to a p.d. kernel K on $\mathcal{X}$.
- KRR is obtained by regularizing the MSE criterion by the RKHS norm:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}}^2 \qquad (2)$$

- *1st effect = prevent overfitting by penalizing non-smooth functions.*
- By the representer theorem, any solution of (2) can be expanded as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right).$$

- *2nd effect = simplifying the solution.*

Solving KRR

- Let $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$
- Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$
- Let $\mathbf{K}$ be the $n \times n$ Gram matrix: $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$
- We can then write:

$$\left(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n)\right)^\top = \mathbf{K}\boldsymbol{\alpha}$$

- The following holds as usual:

$$\| \hat{f} \|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

- The KRR problem (2) is therefore equivalent to:

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})^\top (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}) + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

Solving KRR

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n}\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)^\top\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)+\lambda\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}$$

- This is a convex and differentiable function of $\boldsymbol{\alpha}$. Its minimum can therefore be found by setting the gradient in $\boldsymbol{\alpha}$ to zero:

$$\begin{aligned} 0 &= \frac{2}{n}\mathbf{K}\left(\mathbf{K}\boldsymbol{\alpha}-\mathbf{y}\right)+2\lambda\mathbf{K}\boldsymbol{\alpha} \\ &= \mathbf{K}\left[\left(\mathbf{K}+\lambda n\mathbf{I}\right)\boldsymbol{\alpha}-\mathbf{y}\right] \end{aligned}$$

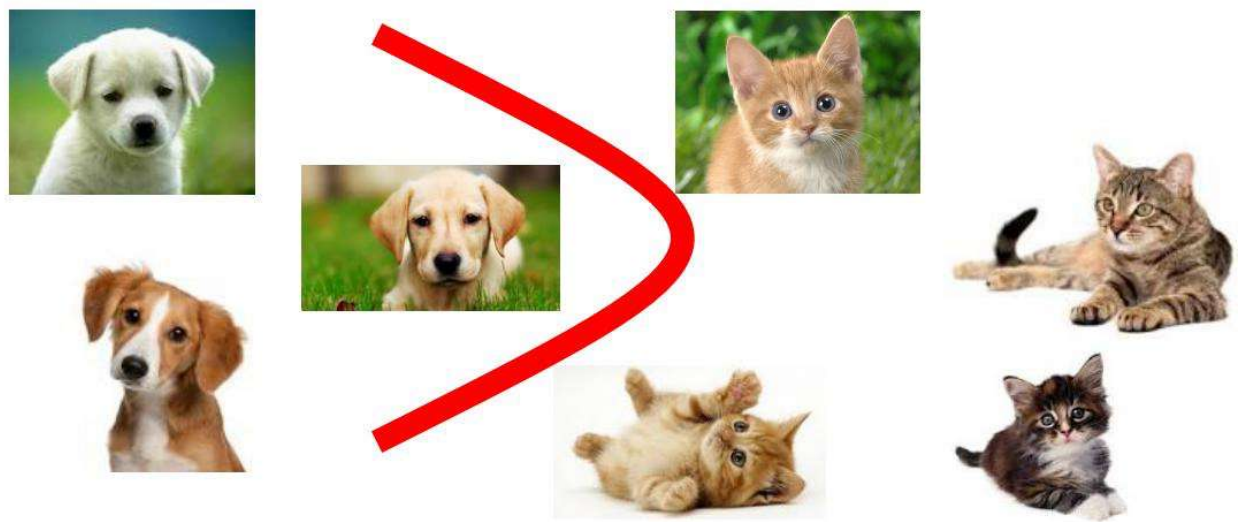
- For $\lambda > 0$, $\mathbf{K}+\lambda n\mathbf{I}$ is invertible (because $\mathbf{K}$ is positive semidefinite) so one solution is to take:

$$\boldsymbol{\alpha} = \left(\mathbf{K}+\lambda n\mathbf{I}\right)^{-1}\mathbf{y}.$$

Binary classification

Setup

- $\mathcal{X}$ set of inputs
- $\mathcal{Y} = \{-1, 1\}$ binary outputs
- $\mathcal{S}_n = (\mathbf{x}_i, y_i)_{i=1,\dots,n} \in (\mathcal{X} \times \mathcal{Y})^n$ a training set of n pairs
- Goal = find a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $sign(f(\mathbf{x}))$



The 0/1 loss

- The 0/1 loss measures if a prediction is correct or not:

$$\ell_{0/1}\left(f(\mathbf{x}), y\right)) = \mathbf{1}\left(yf(\mathbf{x}) < 0\right) = \begin{cases} 0 & \text{if } y = sign\left(f(\mathbf{x})\right) \\ 1 & \text{otherwise.} \end{cases}$$

- It is then tempting to learn f by solving:

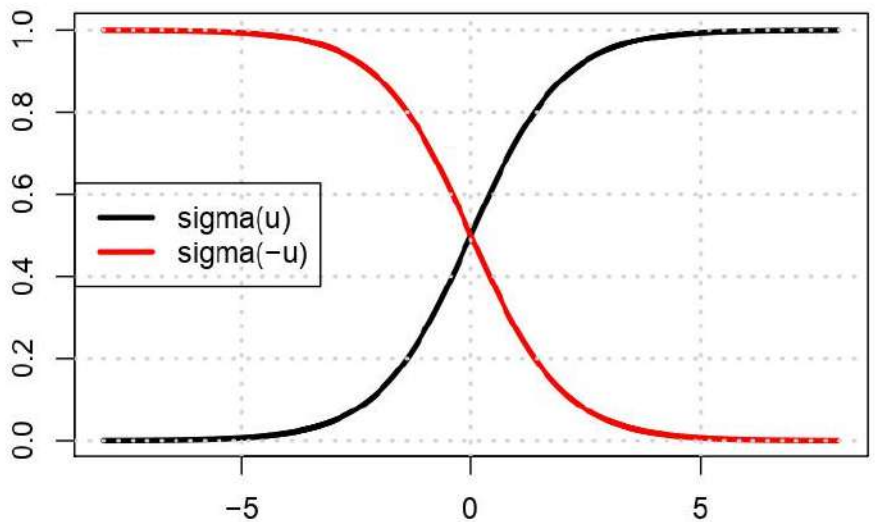
$$\min_{f \in \mathcal{H}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell_{0/1}\left(f\left(\mathbf{x}_i\right), y_i\right)}_{\text{misclassification rate}} + \underbrace{\lambda \| f \|_{\mathcal{H}}^2}_{\text{regularization}}$$

- However:
  - The problem is non-smooth, and typically NP-hard to solve

The logistic loss

- An alternative is to define a probabilistic model of y parametrized by $f(\mathbf{x})$, e.g.:

$$\forall \mathbf{y} \in \{-1, 1\}, \quad p(y \mid f(\mathbf{x})) = \frac{1}{1 + e^{-yf(\mathbf{x})}} = \sigma(yf(\mathbf{x}))$$



- The logistic loss is the negative conditional likelihood:

$$\ell_{logistic}(f(\mathbf{x}), y) = -\ln p(y \mid f(\mathbf{x})) = \ln\left(1 + e^{-yf(\mathbf{x})}\right)$$

Kernel logistic regression (KLR)

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell_{logistic}\left(f(\mathbf{x}_i), y_i\right) + \frac{\lambda}{2} \| f \|_{\mathcal{H}}^2$$

$$= \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i f(\mathbf{x}_i)}\right) + \frac{\lambda}{2} \| f \|_{\mathcal{H}}^2$$

- Can be interpreted as a regularized conditional maximum likelihood estimator
- No explicit solution, but smooth convex optimization problem that can be solved numerically

Solving KLR

- By the representer theorem, any solution of KLR can be expanded as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

and as always we have:

$$\left(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n)\right)^\top = \mathbf{K}\boldsymbol{\alpha} \quad \text{and} \quad \| \hat{f} \|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

- To find $\boldsymbol{\alpha}$ we therefore need to solve:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i [\mathbf{K}\boldsymbol{\alpha}]_i}\right) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

Given:

- $f(x_i) = \sum_{j=1}^{n} \alpha_j K(x_i, x_j)$, where $K(x_i, x_j)$ is the kernel function.
- The probability in logistic regression is $\sigma(f(x_i)) = \frac{1}{1+e^{-\sum_{j=1}^{n} \alpha_j K(x_i, x_j)}}$.

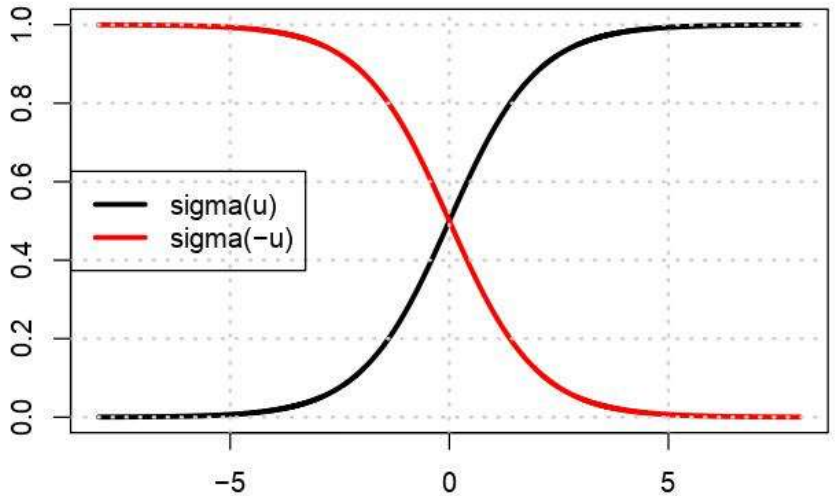
1. Logistic Regression Cost Function

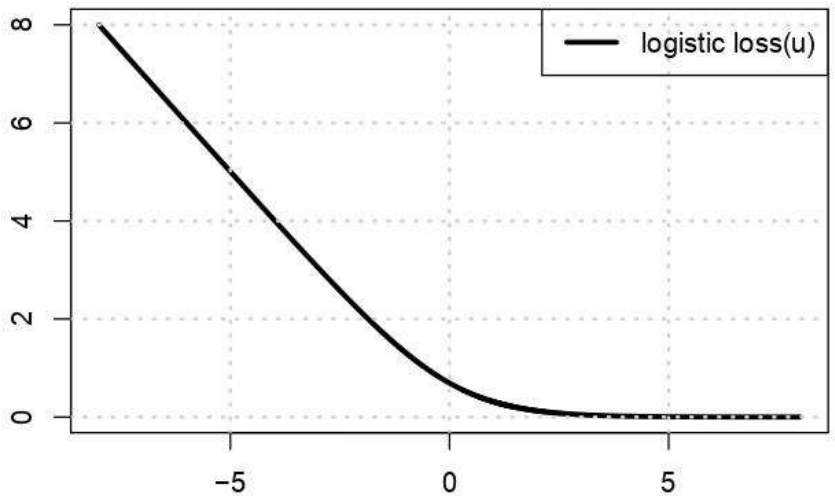
The cost function for logistic regression is:

$$J(\alpha) = -\sum_{i=1}^{n} [y_i \log(\sigma(f(x_i))) + (1 - y_i) \log(1 - \sigma(f(x_i)))] + \frac{\lambda}{2} \sum_{j,k=1}^{n} \alpha_j \alpha_k K(x_j, x_k).$$

Here, λ is the regularization parameter.

Technical facts





Sigmoid:

- $\sigma(u) = \frac{1}{1+e^{-u}}$
- $\sigma(-u) = 1 - \sigma(u)$
- $\sigma'(u) = \sigma(u)\sigma(-u) \geq 0$

Logistic loss:

- $\ell_{logistic}(u) = \ln\left(1 + e^{-u}\right)$
- $\ell'_{logistic}(u) = -\sigma(-u)$
- $\ell''_{logistic}(u) = \sigma(u)\sigma(-u) \geq 0$

Step 1: Derivative of the Logistic Loss

The logistic loss for a single sample x_i is:

$$L_i = -\left[y_i \log(\sigma(f(x_i))) + (1 - y_i) \log(1 - \sigma(f(x_i)))\right].$$

Since $f(x_i) = \sum_{j=1}^{n} \alpha_j K(x_i, x_j)$, the derivative of the logistic loss with respect to $f(x_i)$ is:

$$\frac{\partial L_i}{\partial f(x_i)} = \sigma(f(x_i)) - y_i.$$

Now, we need to express this in terms of α_j. The derivative of $f(x_i)$ with respect to α_j is:

$$\frac{\partial f(x_i)}{\partial \alpha_j} = K(x_i, x_j).$$

Using the chain rule, the derivative of L_i with respect to α_j is:

$$\frac{\partial L_i}{\partial \alpha_j} = \frac{\partial L_i}{\partial f(x_i)} \frac{\partial f(x_i)}{\partial \alpha_j} = (\sigma(f(x_i)) - y_i) K(x_i, x_j).$$

Step 2: Derivative of the Regularization Term

The regularization term is:

$$\frac{\lambda}{2} \sum_{j,k=1}^{n} \alpha_j \alpha_k K(x_j, x_k).$$

The derivative of this term with respect to α_j is:

$$\frac{\partial}{\partial \alpha_j} \left(\frac{\lambda}{2} \sum_{k=1}^{n} \alpha_k K(x_j, x_k) \right) = \lambda \sum_{k=1}^{n} \alpha_k K(x_j, x_k).$$

3. Combining Both Parts

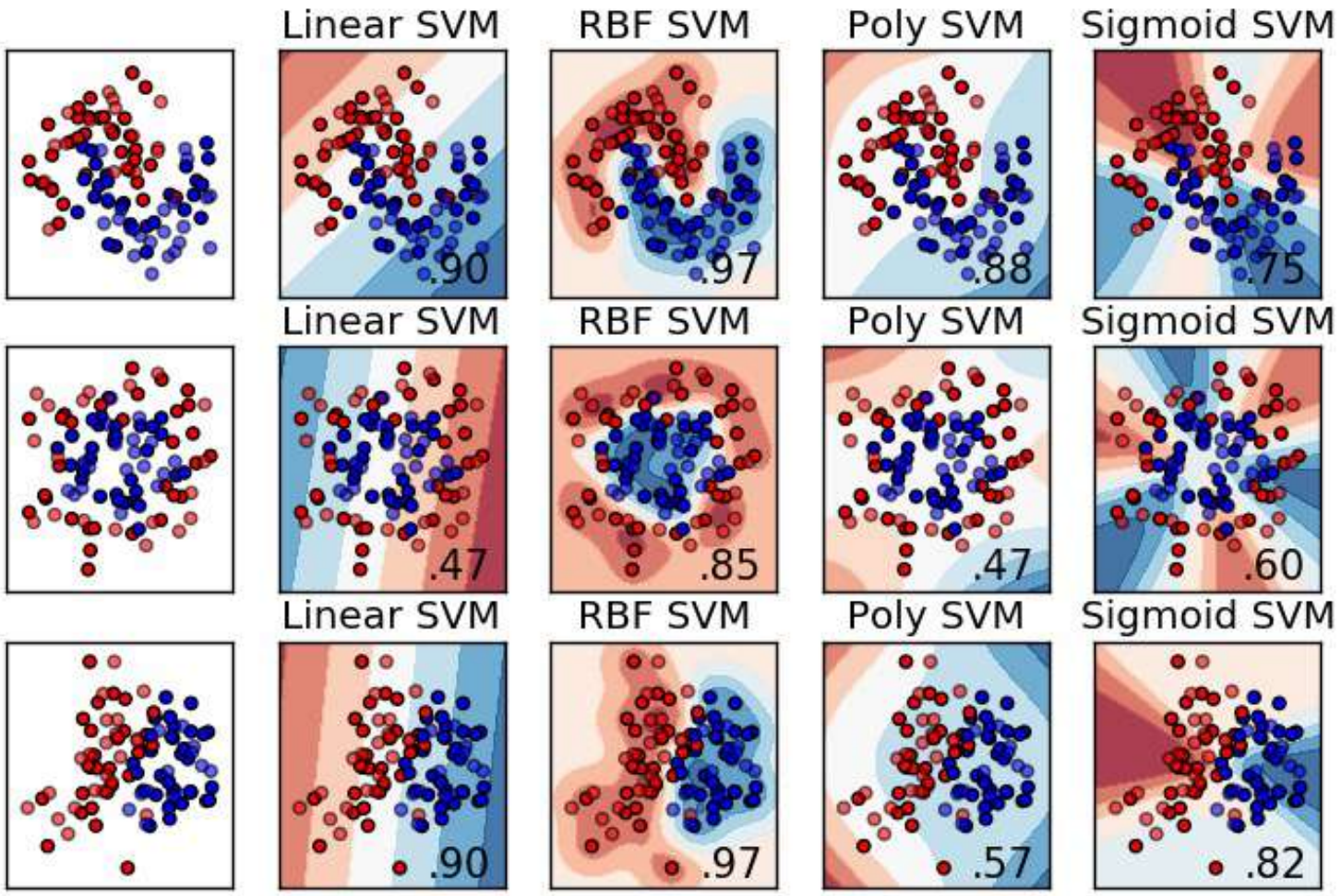
Now, we combine the derivatives from both parts (logistic loss and regularization):

$$\frac{\partial J(\alpha)}{\partial \alpha_j} = \sum_{i=1}^{n} (\sigma(f(x_i)) - y_i) K(x_i, x_j) + \lambda \sum_{k=1}^{n} \alpha_k K(x_j, x_k).$$

Final Expression

The derivative of the logistic regression cost function in the kernelized form with respect to the coefficient α_j is:

$$\frac{\partial J(\alpha)}{\partial \alpha_j} = \sum_{i=1}^{n} (\sigma(\sum_{k=1}^{n} \alpha_k K(x_i, x_k)) - y_i) K(x_i, x_j) + \lambda \sum_{k=1}^{n} \alpha_k K(x_j, x_k).$$



Kernel SVM example

Different kernels induce different regularizations

The norm depends on the kernel

Changing the kernel we change the regularization

Since $f = \sum_{j=1}^{n} c_j K_{x_j}$, then

$$\begin{aligned} \|f\|_{\mathcal{H}}^2 &= \langle f, f \rangle_{\mathcal{H}} \\ &= \langle \sum_{i=1}^{n} c_i K_{x_i}, \sum_{j=1}^{n} c_j K_{x_j} \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \langle K_{x_i}, K_{x_j} \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) = c^t \mathbf{K} c \end{aligned}$$

$$\|f\|_{\mathcal{H}} = c^T K c = c^T U^T \Lambda U c = \tilde{c}^T \Lambda \tilde{c}$$

Smoothness functional

A simple inequality

- By Cauchy-Schwarz we have, for any function $f \in \mathcal{H}$ and any two points $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

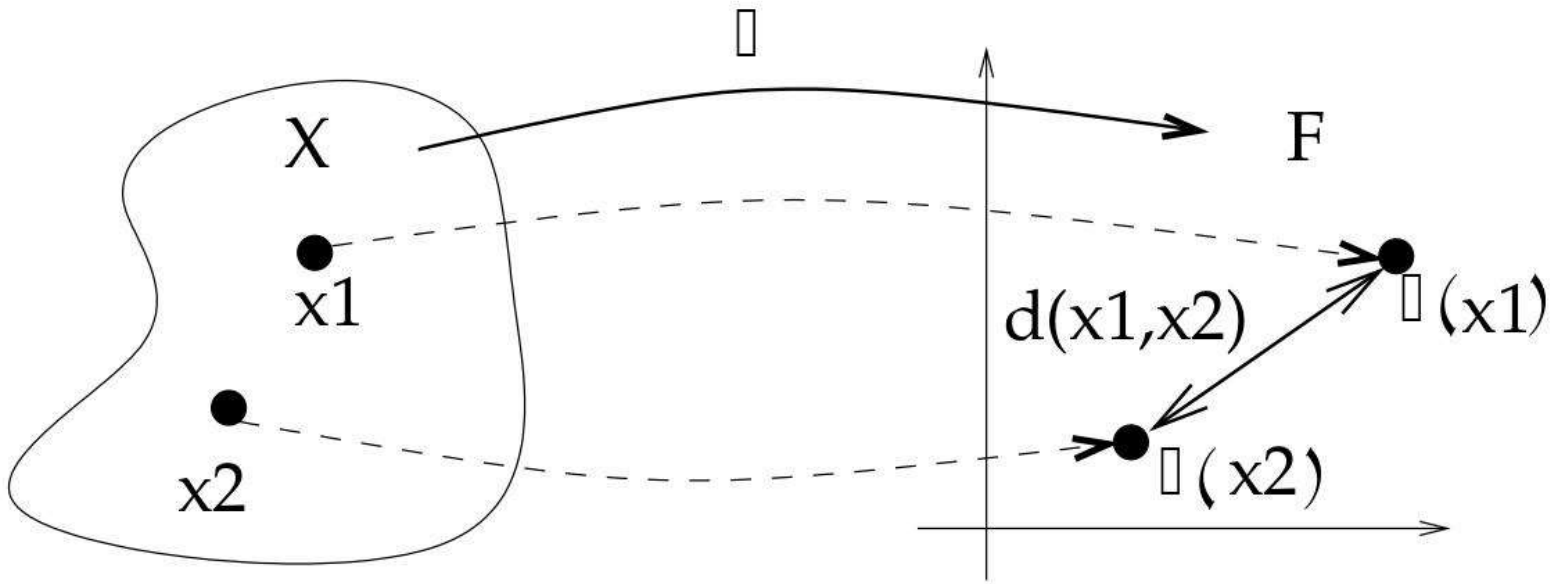
$$\begin{aligned} \left| f(\mathbf{x}) - f(\mathbf{x}') \right| &= \left| \langle f, K_{\mathbf{x}} - K_{\mathbf{x}'} \rangle_{\mathcal{H}} \right| \\ &\leq \| f \|_{\mathcal{H}} \times \| K_{\mathbf{x}} - K_{\mathbf{x}'} \|_{\mathcal{H}} \\ &= \| f \|_{\mathcal{H}} \times d_K(\mathbf{x}, \mathbf{x}') \ . \end{aligned}$$

- The norm of a function in the RKHS controls how fast the function varies over $\mathcal{X}$ with respect to the geometry defined by the kernel (Lipschitz with constant $\| f \|_{\mathcal{H}}$).

Important message

Small norm $\Longrightarrow$ slow variations.

Example 1: computing distances in the feature space



$$d_K(\mathbf{x}_1, \mathbf{x}_2)^2 = \| \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \|^2_{\mathcal{H}}$$
$$= \langle \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}}$$
$$= \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_1) \rangle_{\mathcal{H}} + \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} - 2 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}}$$

$$d_K(\mathbf{x}_1, \mathbf{x}_2)^2 = K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2)$$

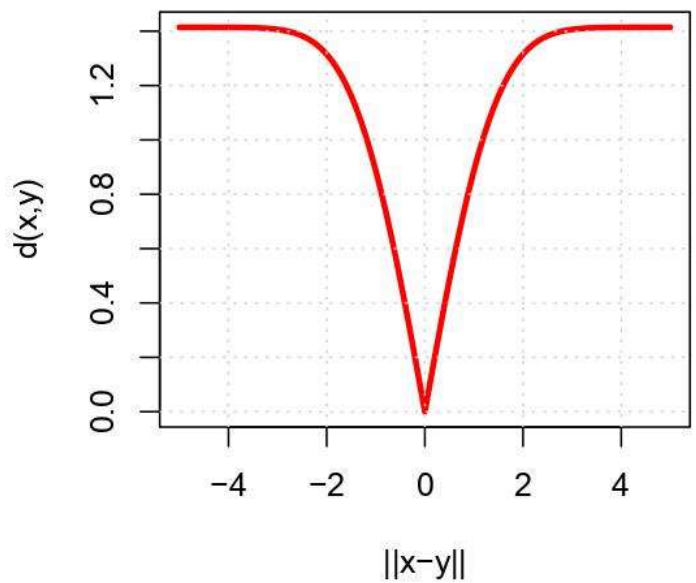
Distance for the Gaussian kernel

- The Gaussian kernel with bandwidth σ on $\mathbb{R}^d$ is:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}},$$

- $K(\mathbf{x}, \mathbf{x}) = 1 = \|\Phi(\mathbf{x})\|^2_{\mathcal{H}}$, so all points are on the unit sphere in the feature space.
- The distance between the images of two points $\mathbf{x}$ and $\mathbf{y}$ in the feature space is given by:

$$d_K(\mathbf{x}, \mathbf{y}) = \sqrt{2\left[1 - e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}\right]}$$



Example 2: distance between a point and a set

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$.
- How to define and compute the similarity between any point $\mathbf{x}$ in $\mathcal{X}$ and the set $\mathcal{S}$?

Where do you use this quantity?

Example 2: distance between a point and a set

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$.
- How to define and compute the similarity between any point $\mathbf{x}$ in $\mathcal{X}$ and the set $\mathcal{S}$?

Where do you use this quantity?

K-means

Example 2: distance between a point and a set

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$.
- How to define and compute the similarity between any point $\mathbf{x}$ in $\mathcal{X}$ and the set $\mathcal{S}$?

A solution:

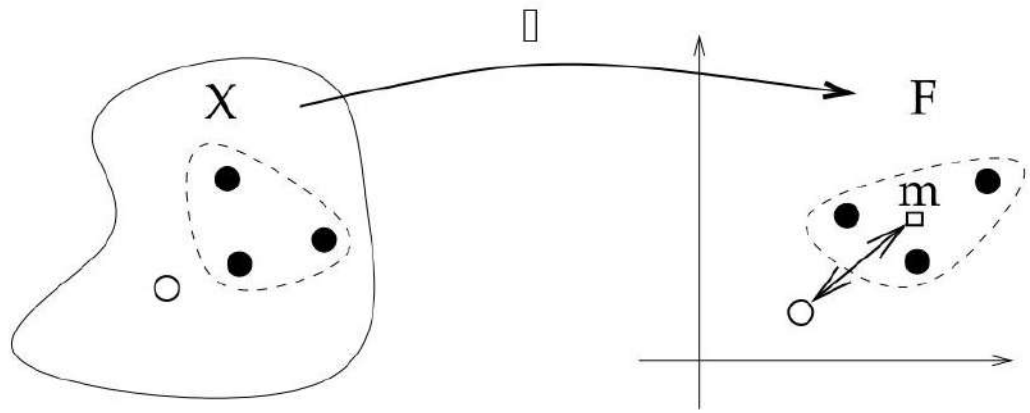
- Map all points to the feature space.
- Summarize $\mathcal{S}$ by the barycenter of the points:

$$\boldsymbol{\mu} := \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i) \ .$$

- Define the distance between $\mathbf{x}$ and $\mathcal{S}$ by:

$$d_K(\mathbf{x}, \mathcal{S}) := \| \Phi(\mathbf{x}) - \boldsymbol{\mu} \|_{\mathcal{H}} \ .$$

Computation

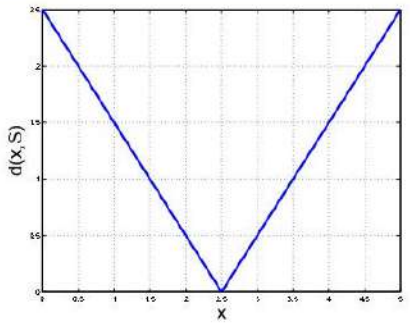


$$d_K(\mathbf{x}, \mathcal{S}) = \left\| \Phi(\mathbf{x}) - \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i) \right\|_{\mathcal{H}}$$

$$= \sqrt{K(\mathbf{x}, \mathbf{x}) - \frac{2}{n} \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)}.$$

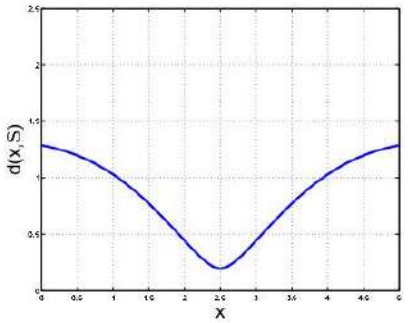
1D illustration

- $\mathcal{S} = \{2, 3\}$
- Plot $f(x) = d(x, \mathcal{S})$



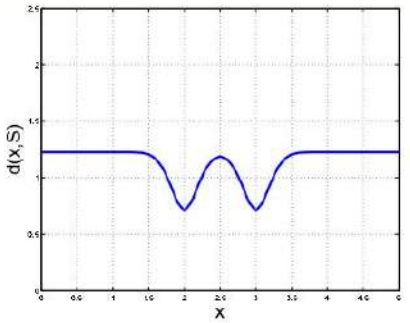
$K(x, y) = xy.$

(linear)



$K(x, y) = e^{-\frac{(x-y)^2}{2\sigma^2}}.$

with $\sigma = 1.$



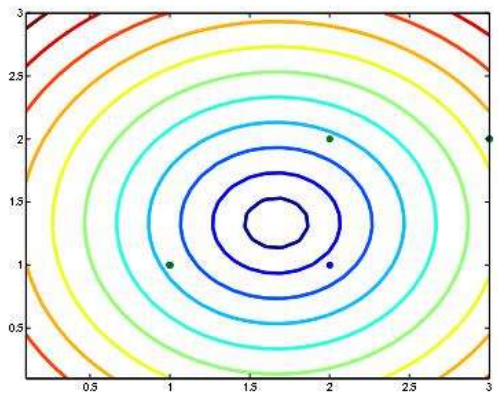
$K(x, y) = e^{-\frac{(x-y)^2}{2\sigma^2}}.$

with $\sigma = 0.2.$

With the Gaussian kernel we separate the two points!

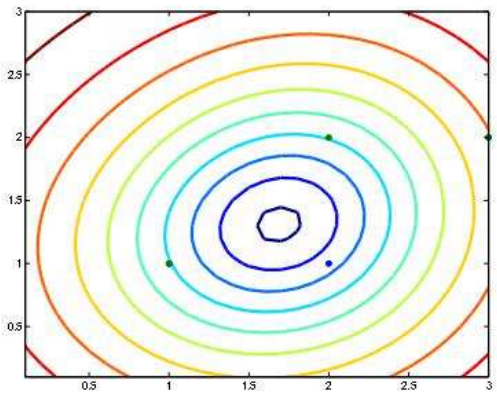
2D illustration

- $\mathcal{S} = \{(1,1)', (1,2)', (2,2)'\}$
- Plot $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S})$



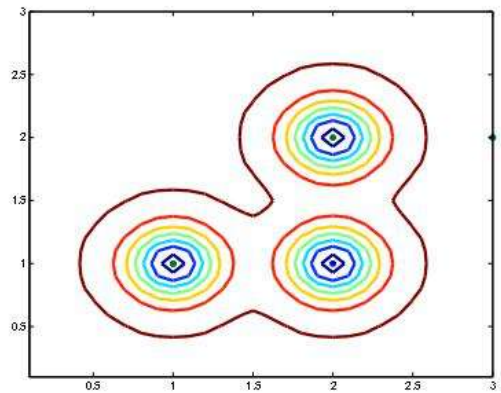
$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$

(linear)



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

with $\sigma = 1$.

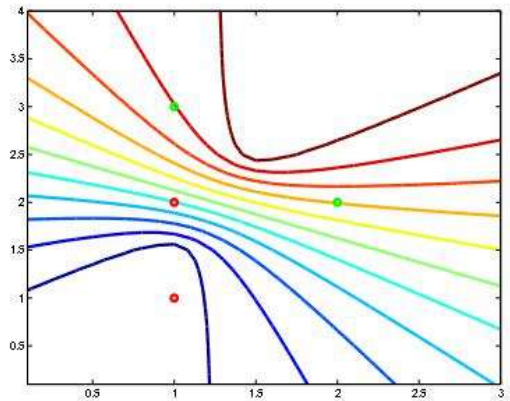


$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

with $\sigma = 0.2$.

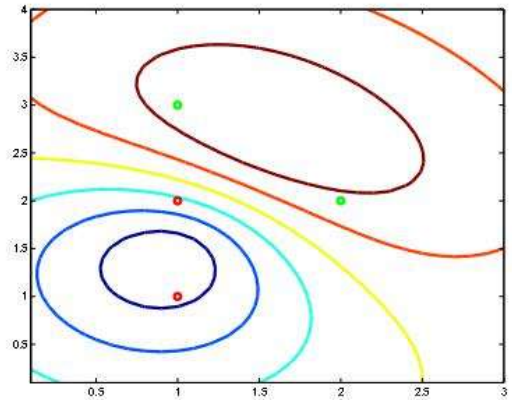
Basic application in discrimination

- $\mathcal{S}_1 = \{(1,1)', (1,2)'\}$ and $\mathcal{S}_2 = \{(1,3)', (2,2)'\}$
- Plot $f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S}_1)^2 - d(\mathbf{x}, \mathcal{S}_2)^2$



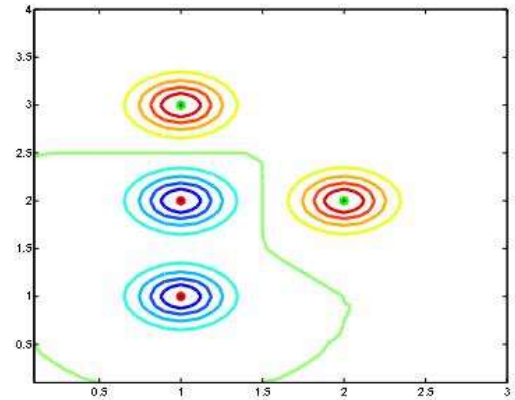
$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$

(linear)



$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

with $\sigma = 1$.



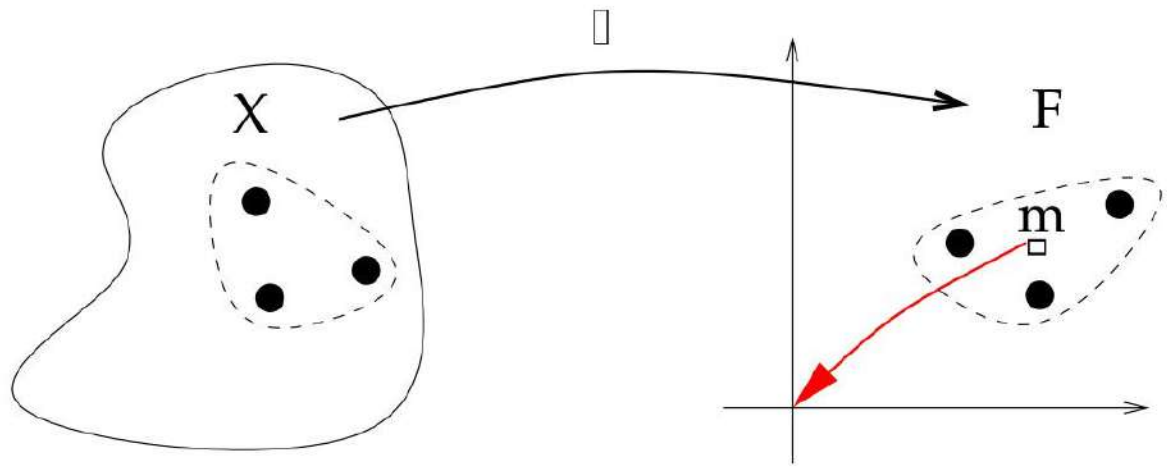
$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

with $\sigma = 0.2$.

Example 3: Centering data in the feature space

Problem

- Let $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be a finite set of points in $\mathcal{X}$ endowed with a p.d. kernel K. Let $\mathbf{K}$ be their $n \times n$ Gram matrix: $[\mathbf{K}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
- Let $\boldsymbol{\mu} = 1/n \sum_{i=1}^{n} \Phi(\mathbf{x}_i)$ their barycenter, and $\mathbf{u}_i = \Phi(\mathbf{x}_i) - \boldsymbol{\mu}$ for $i = 1, \ldots, n$ be centered data in $\mathcal{H}$.
- How to compute the centered Gram matrix $[\mathbf{K}^c]_{i,j} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle_{\mathcal{H}}$?



Computation

- A direct computation gives, for $0 \leq i, j \leq n$:

$$\begin{aligned}
\mathbf{K}^c_{i,j} &= \langle \Phi(\mathbf{x}_i) - \boldsymbol{\mu}, \Phi(\mathbf{x}_j) - \boldsymbol{\mu} \rangle_{\mathcal{H}} \\
&= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}} - \langle \boldsymbol{\mu}, \Phi(\mathbf{x}_i) + \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}} + \langle \boldsymbol{\mu}, \boldsymbol{\mu} \rangle_{\mathcal{H}} \\
&= \mathbf{K}_{i,j} - \frac{1}{n} \sum_{k=1}^{n} (\mathbf{K}_{i,k} + \mathbf{K}_{j,k}) + \frac{1}{n^2} \sum_{k,l=1}^{n} \mathbf{K}_{k,l} \,.
\end{aligned}$$

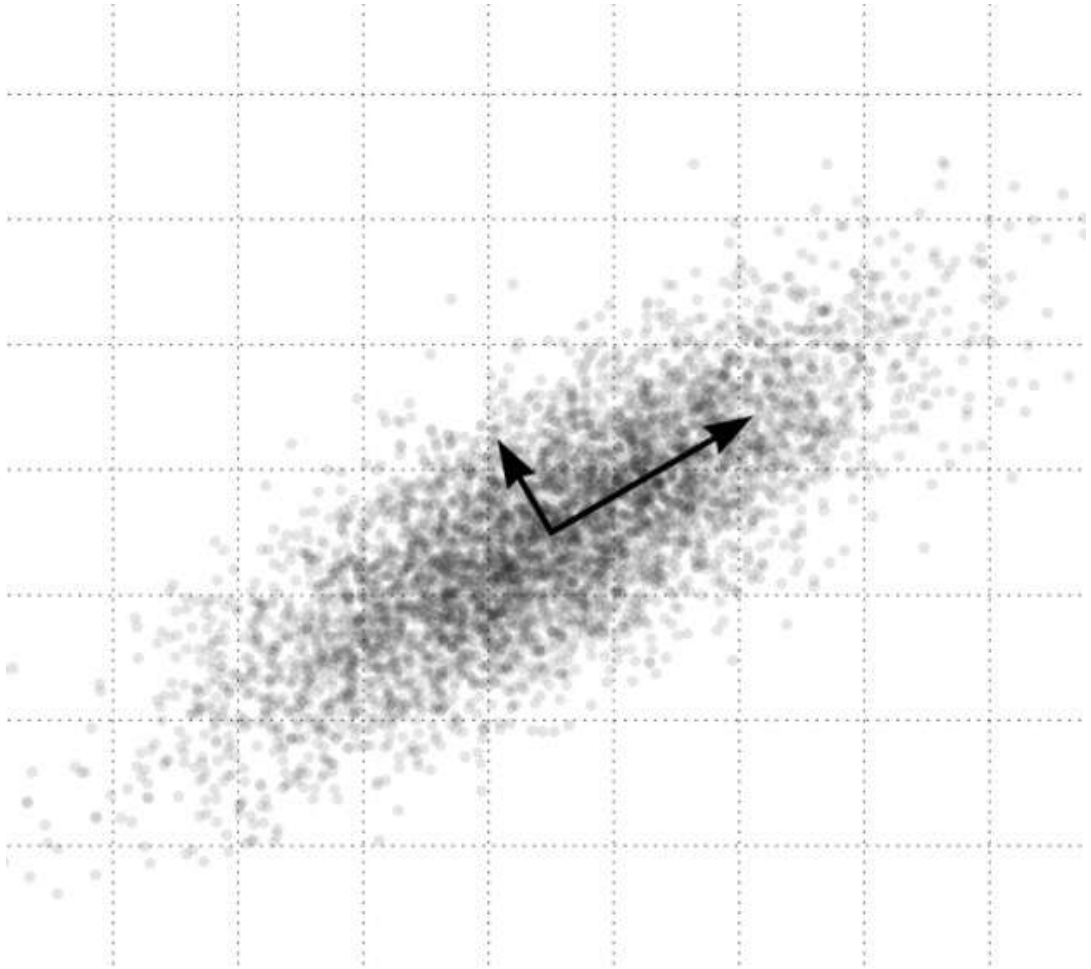
- This can be rewritten in matricial form:

$$\mathbf{K}^c = \mathbf{K} - \mathbf{U}\mathbf{K} - \mathbf{K}\mathbf{U} + \mathbf{U}\mathbf{K}\mathbf{U} = (\mathbf{I} - \mathbf{U})\,\mathbf{K}\,(\mathbf{I} - \mathbf{U}) \,,$$

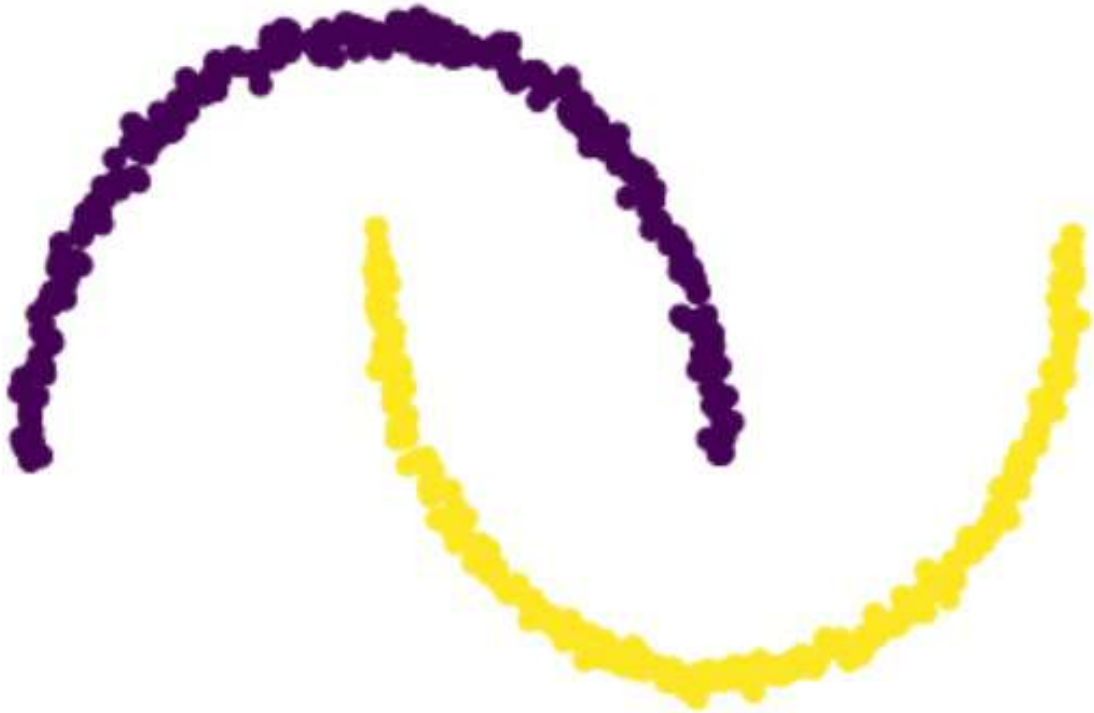
where $\mathbf{U}_{i,j} = 1/n$ for $1 \leq i, j \leq n$.

Kernel PCA

PCA



What if?



Kernel PCA

We want to extend the PCA in the features space.

- Calculate the covariance matrix in the feats space:

$$C_F = \frac{1}{N} \sum_{i}^{N} \phi(x_i) \phi^T(x_i)$$

- We want to solve $C_F v = \lambda v$

Eigenvectors can be expressed as a linear combination of the features i.e.

$$v = \sum_{i=1}^{N} \alpha_i \phi(x_i)$$

- We have

$$C_F v = \lambda v = \frac{1}{N} \sum_{i}^{N} \phi(x_i) \phi^T(x_i) v$$

- Thus

$$v = \frac{1}{N\lambda} \sum_{i}^{N} \phi(x_i) \phi^T(x_i) v = \sum_{i}^{N} \phi(x_i) \frac{\beta_i}{N\lambda} = \sum_{i=1}^{N} \alpha_i \phi(x_i)$$

Let's multiply both sides by $\phi^T(x_i)$

- Expanding

$$\lambda \phi^T(x_k) v = \phi^T(x_k) C_F v$$

$$\lambda \sum_i \alpha_i \phi^T(x_k) \phi(x_i) = \frac{1}{N} \sum_i^N \alpha_i (\phi^T(x_k) \sum_j^N \phi(x_j) (\phi^T(x_j) \phi(x_i))$$

- Define $K_{ij} = \phi^T(x_j)\phi(x_i)$ we have

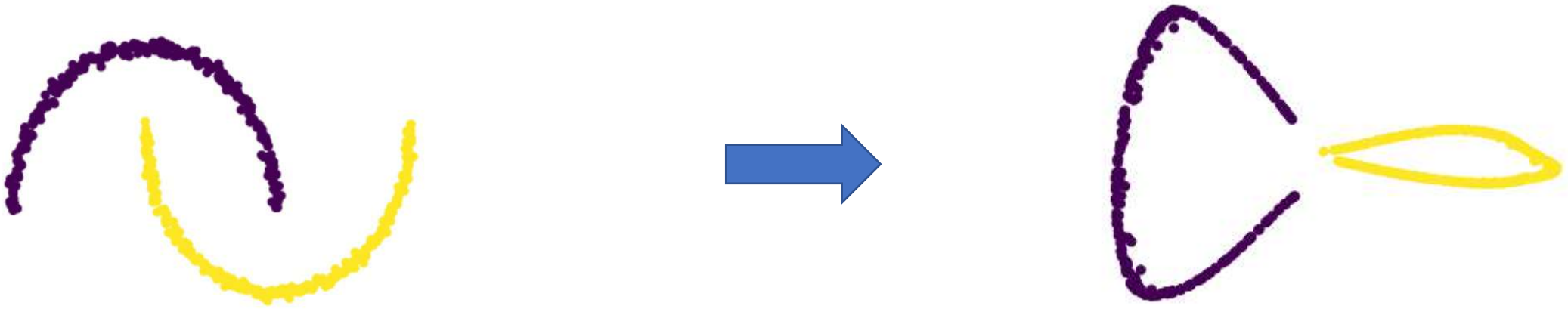
$$N\lambda K\alpha = K^2\alpha \quad \rightarrow \quad N\lambda\alpha = K\alpha$$

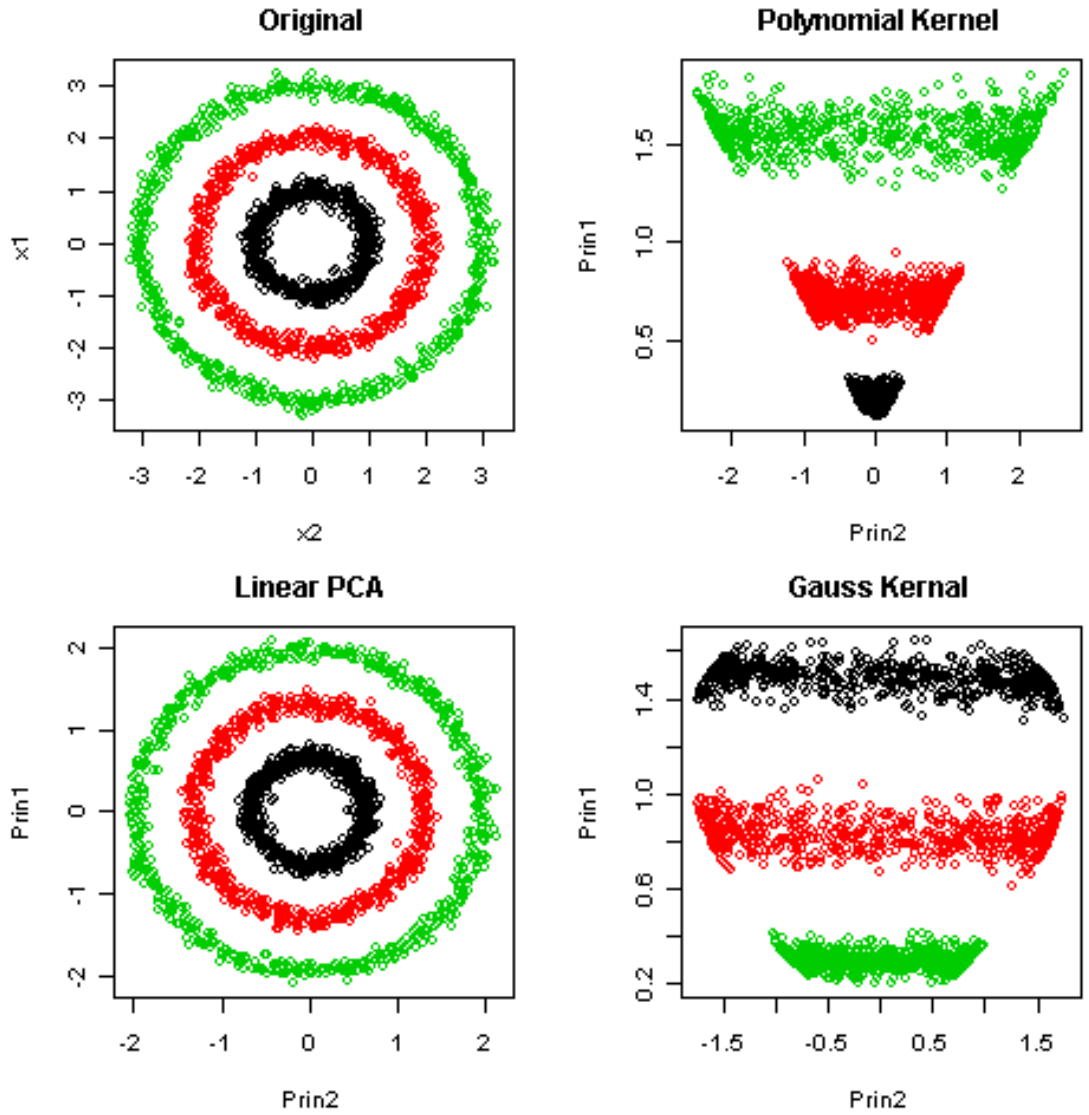
Normalization

$$\hat{\phi}(x_i) = \phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_k)$$

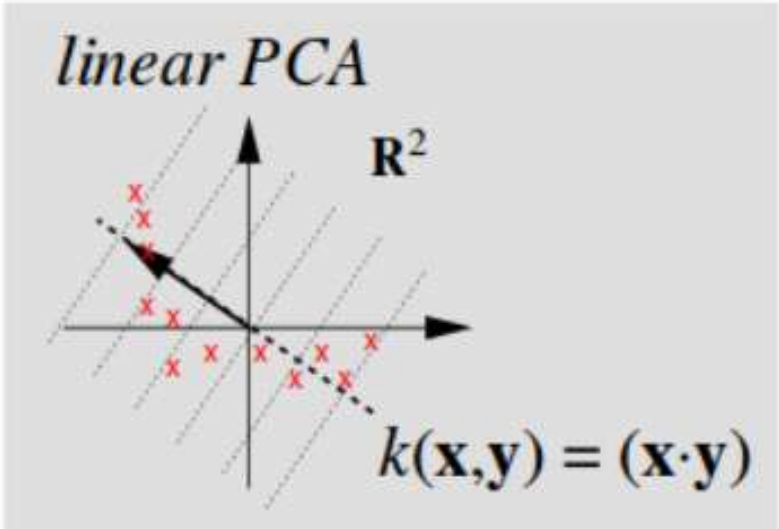
$$\begin{aligned}
\hat{K}_{ij} &= \left(\phi(x_i) - \frac{1}{N}\sum_{k=1}^{N}\phi(x_k)\right)\left(\phi(x_j) - \frac{1}{N}\sum_{l=1}^{N}\phi(x_l)\right) \\
&= K_{ij} - \frac{1}{N}\sum_{k=1}^{N}\phi^T(x_i)\phi(x_k) - \frac{1}{N}\sum_{l=1}^{N}\phi^T(x_j)\phi(x_l) + \frac{1}{N^2}\sum_{kl=1}^{N}\phi^T(x_k)\phi(x_l) \\
&= K_{ij} - \frac{1}{N}\sum_{k=1}^{N}K_{i,k} - \frac{1}{N}\sum_{k=1}^{N}K_{j,k} + \frac{1}{N^2}\sum_{k,l=1}^{N}K_{k,l}
\end{aligned}$$

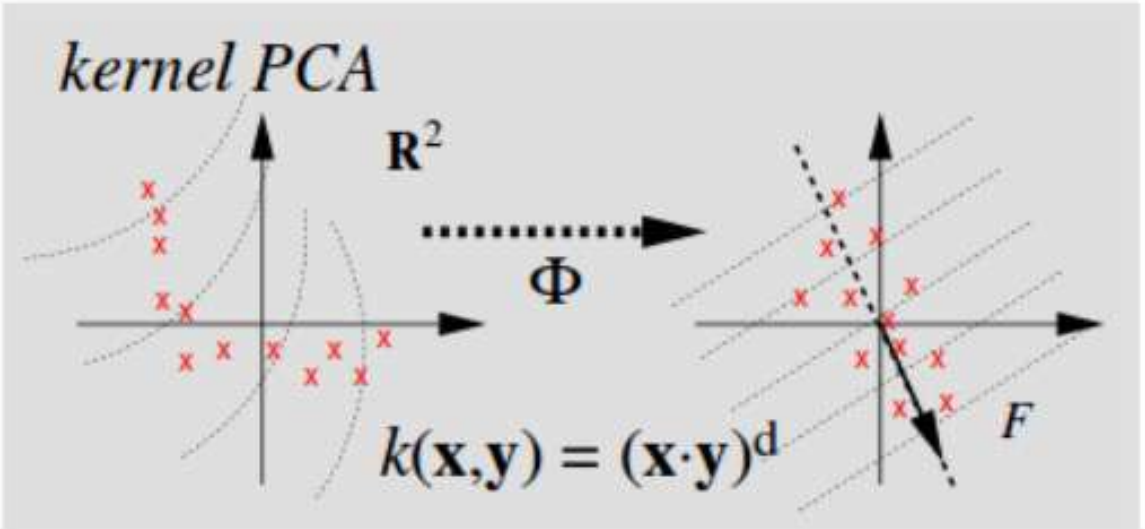
Kernel PCA example





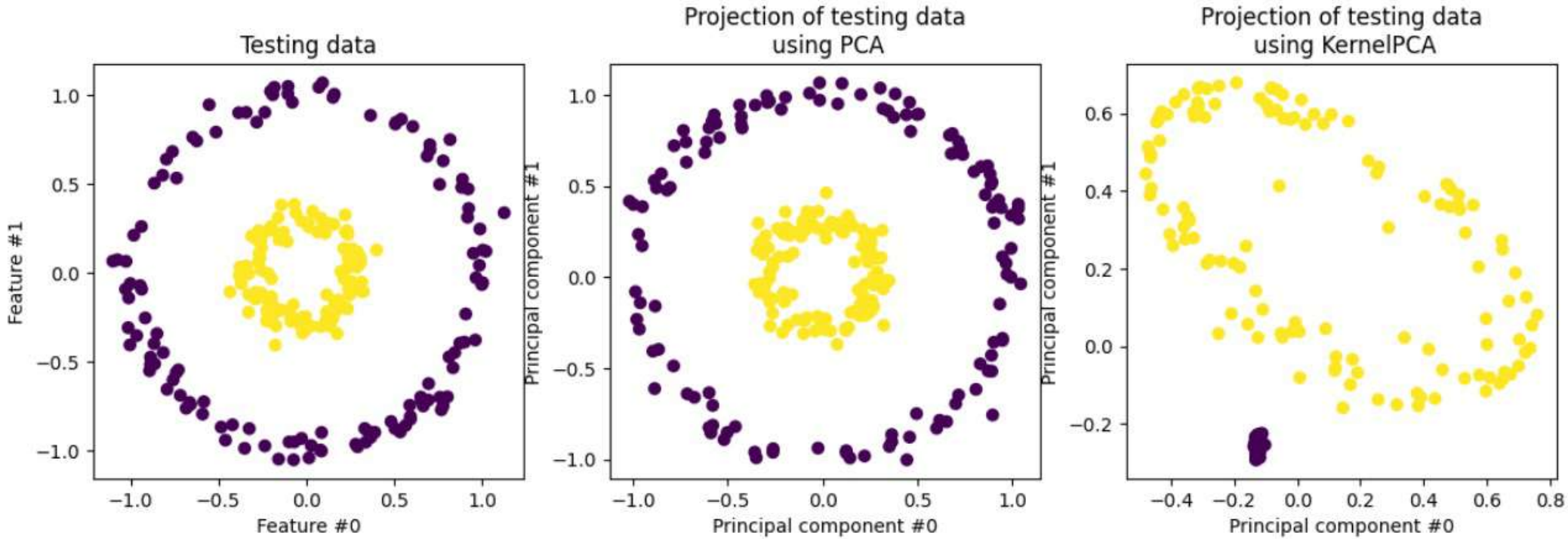
Kernel PCA example





Both PCA and KPCA will reduce the dimensionality but KPCA will also find “non-linear directions”

Kernel PCA


Testing data
Feature #1
Feature #0
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
Projection of testing data
using PCA
Principal component #1
Principal component #0
Projection of testing data
using KernelPCA
Principal component #1
Principal component #0
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

Different kernels different regularizations

The norm depends on the kernel

Changing the kernel we change the regularization

Since $f = \sum_{j=1}^{n} c_j K_{x_j}$, then

$$\begin{aligned} \|f\|_{\mathcal{H}}^2 &= \langle f, f \rangle_{\mathcal{H}} \\ &= \langle \sum_{i=1}^{n} c_i K_{x_i}, \sum_{j=1}^{n} c_j K_{x_j} \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \langle K_{x_i}, K_{x_j} \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) = c^t \mathbf{K} c \end{aligned}$$

$$\|f\|_{\mathcal{H}} = c^T K c = c^T U^T \Lambda U c = \tilde{c}^T \Lambda \tilde{c}$$

Regularization due to kernel choice

Gaussian kernel

$$K(x, y) = e^{-\frac{(x-y)^2}{2\sigma^2}}$$

corresponds to:

$$\varphi(t) = e^{-\frac{t^2}{2\sigma^2}}$$

$$\hat{\varphi}(\omega) = e^{-\frac{\sigma^2\omega^2}{2}}$$

and

$$\mathcal{H} = \left\{ f : \int \left| \hat{f}(\omega) \right|^2 e^{\frac{\sigma^2\omega^2}{2}} d\omega < \infty \right\}.$$

In particular, all functions in $\mathcal{H}$ are infinitely differentiable with all derivatives in L^2.

Laplace kernel

$$K(x,y) = \frac{1}{2} e^{-\gamma|x-y|}$$

corresponds to:

$$\varphi(to) = \frac{1}{2} e^{-\gamma|t|}$$

$$\hat{\varphi}(\omega) = \frac{\gamma}{\gamma^2 + \omega^2}$$

and

$$\mathcal{H} = \left\{ f : \int \left| \hat{f}(\omega) \right|^2 \frac{(\gamma^2 + \omega^2)}{\gamma} d\omega < \infty \right\},$$

the set of functions L^2 differentiable with derivatives in L^2 (Sobolev norm).

Compare with Gaussian Kernel

Low-frequency filter

$$K(x,y) = \frac{\sin(\Omega(x-y))}{\pi(x-y)}$$

corresponds to:

$$\varphi(t) = \frac{\sin(\Omega t)}{\pi t}$$
$$\hat{\varphi}(\omega) = 1_{[-\Omega,\Omega]}(\omega)$$

and

$$\mathcal{H} = \left\{ f : \int_{|\omega|>\Omega} \left| \hat{f}(\omega) \right|^2 d\omega = 0 \right\},$$

the set of functions whose spectrum is included in $[-\Omega, \Omega]$.

Compare with Gaussian Kernel