MACHINE LEARNING II

Kernels

Aude Billard

Understanding Kernels

A Fundamental Concept in Machine Learning and Signal Processing

What is a Kernel?

A kernel can be thought of as **function** that **measures similarity** *between two data points*.

Pairwise measure of distance.

What are Kernel used for?

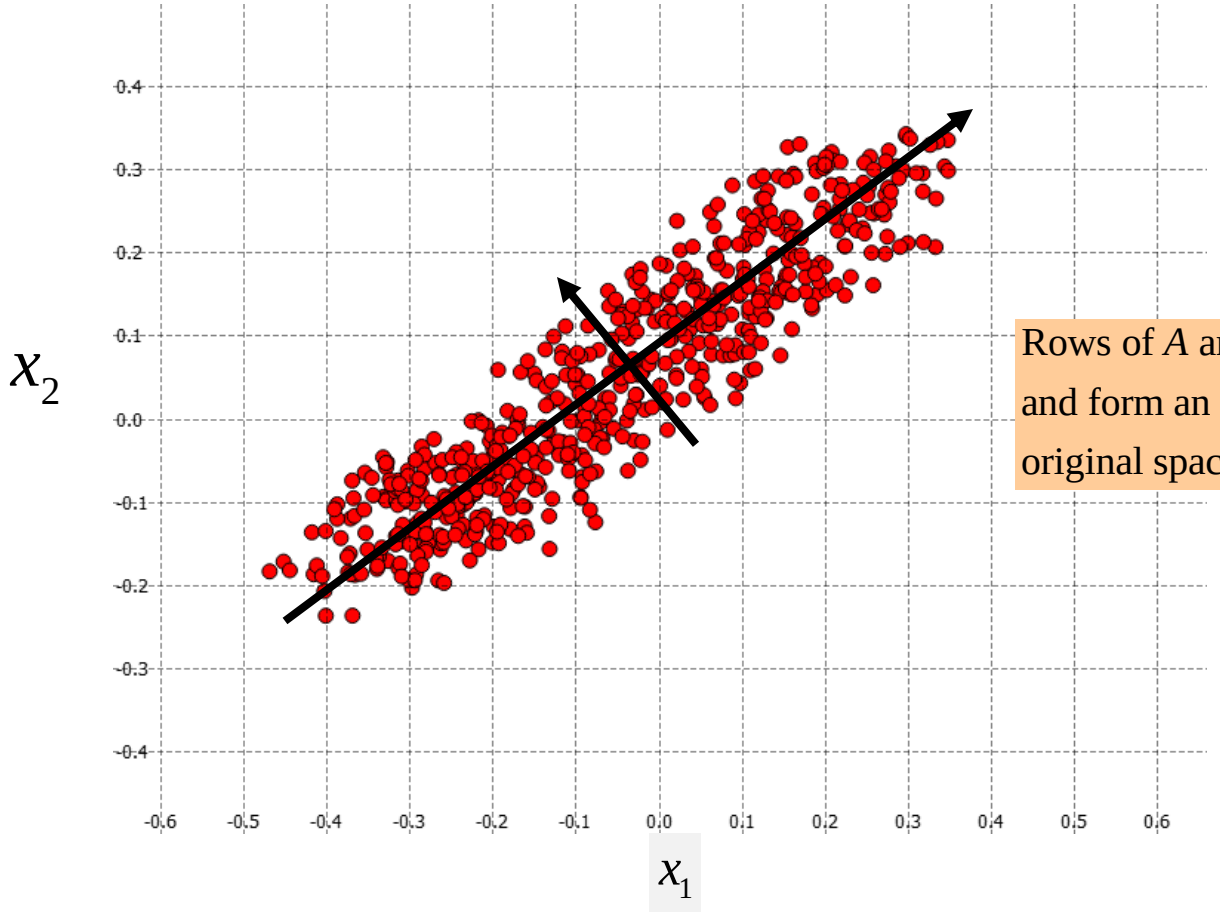
Kernels are used extensively in machine learning to compute ***nonlinear*** problems.

They reduce the problem to a linear problem (Kernel Trick)

From linear to nonlinear transforms

Many of the traditional techniques for dimensionality reduction are *linear.*

Principal component analysis



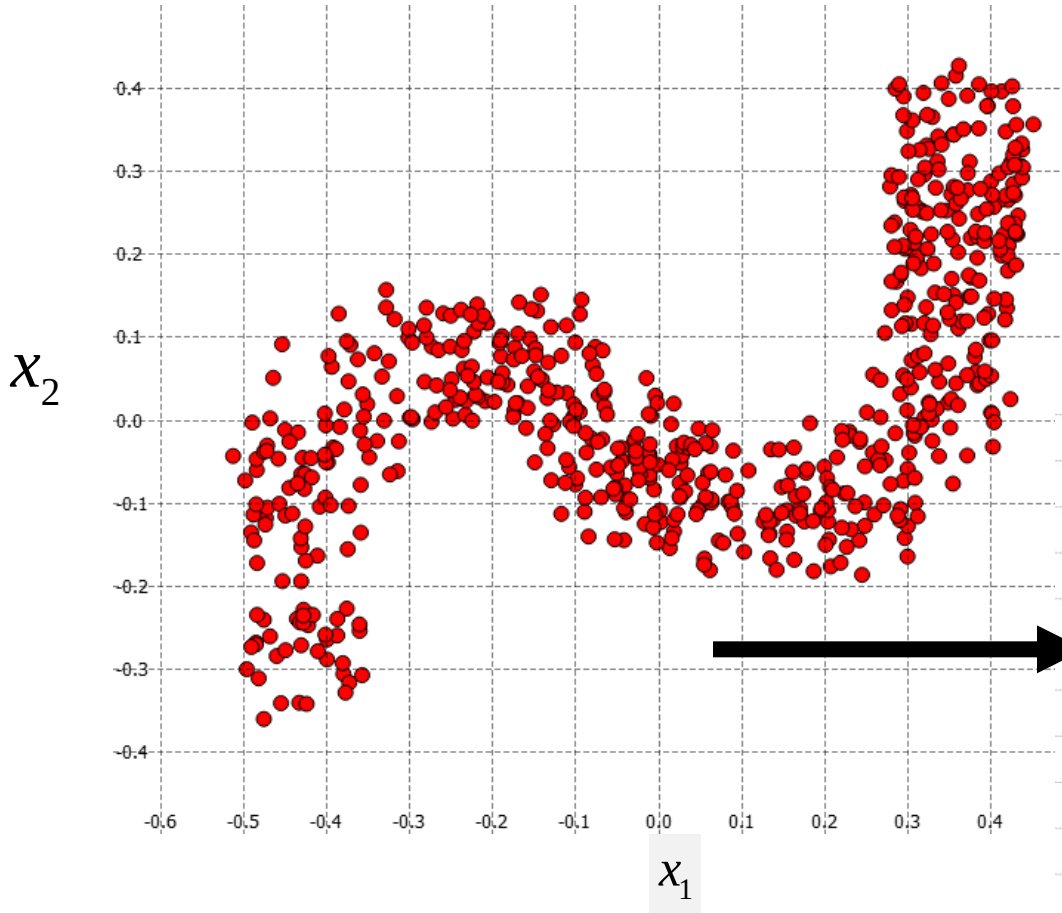
$$x' = Ax$$

Rows of A are projection vectors and form an orthonormal basis of the original space.

From linear transforms to nonlinear ones

What if the problem is nonlinear?
Can we find an **embedding** in which the data appears linear?

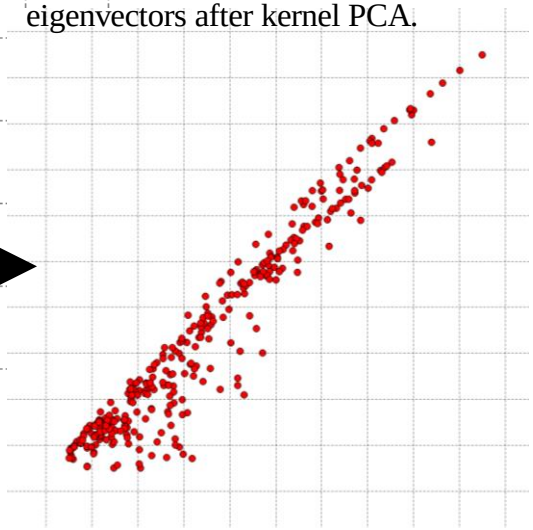
Nonlinear Principal component analysis – kernel PCA



$$x' = A\phi(x),$$

$\phi(x)$: Nonlinear transformation

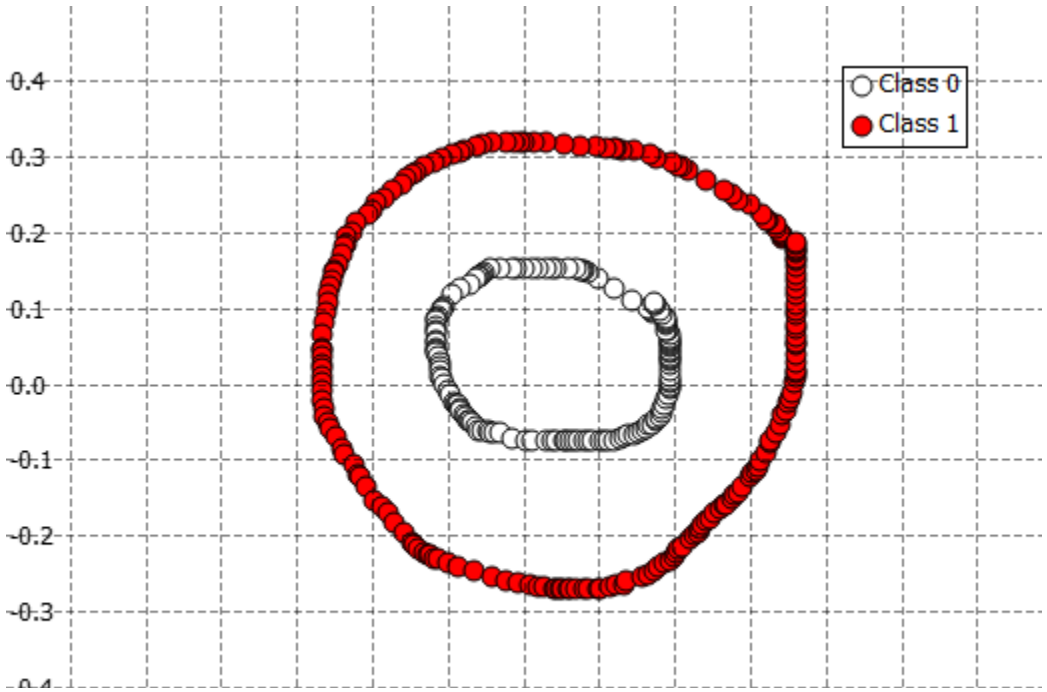
Projections onto two well-chosen eigenvectors after kernel PCA.



Use of kernels: example

Key idea:

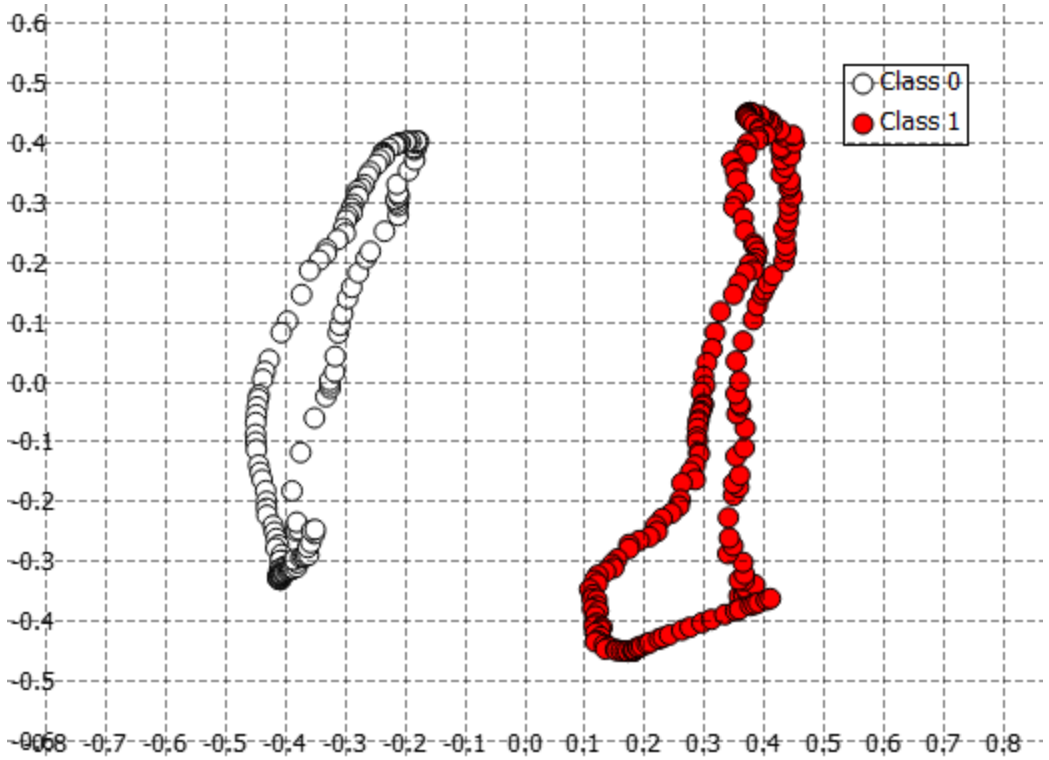
Some problems are made simpler
if you change the representation of the data



Which representation of the data allows to separate *linearly* the two groups of datapoints?

Result after kernel PCA

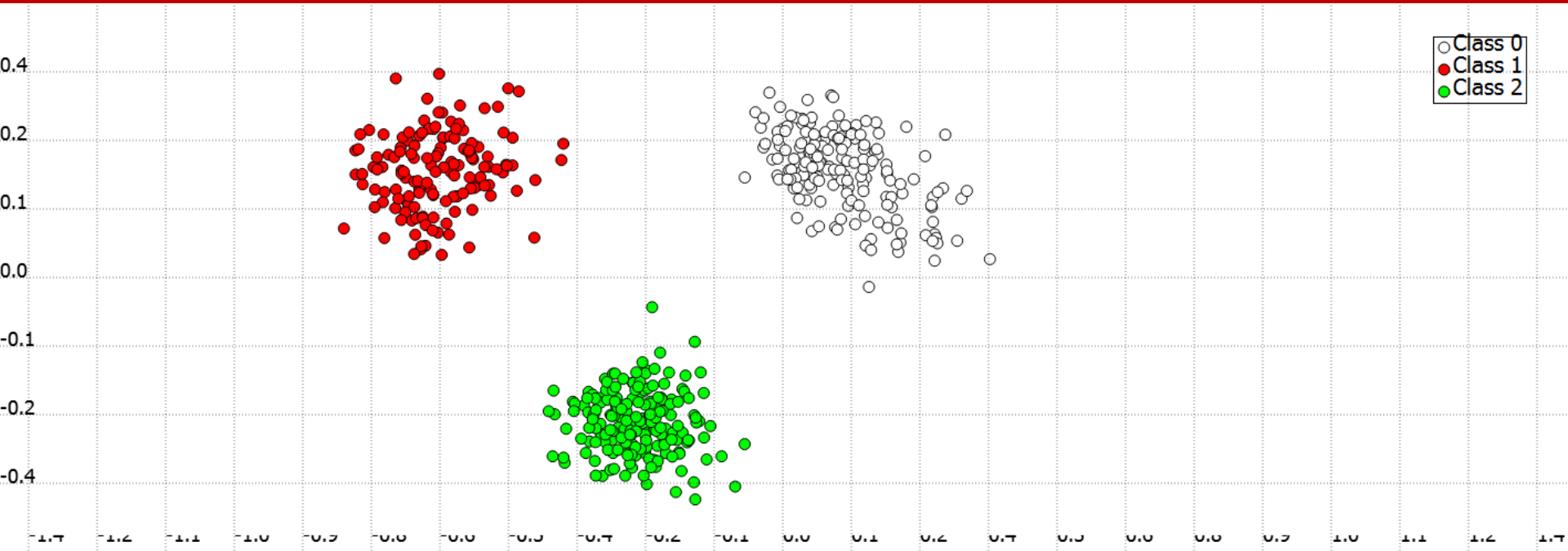
Data becomes linearly separable when projected onto two first principal components of kernel PCA with RBF kernel (see next lecture)



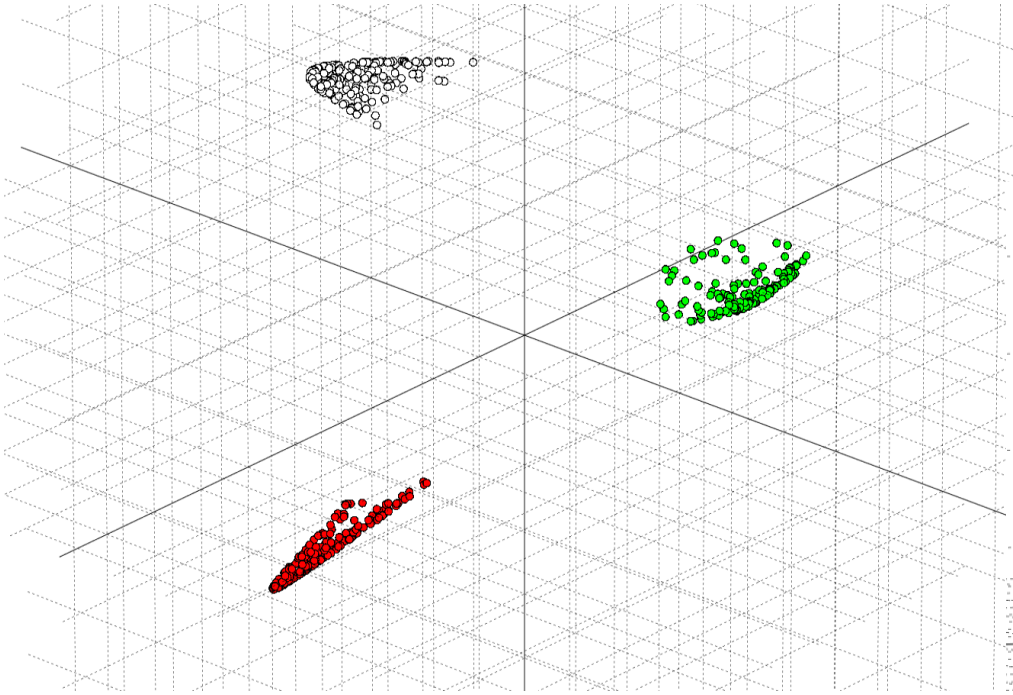
Use of kernels: example

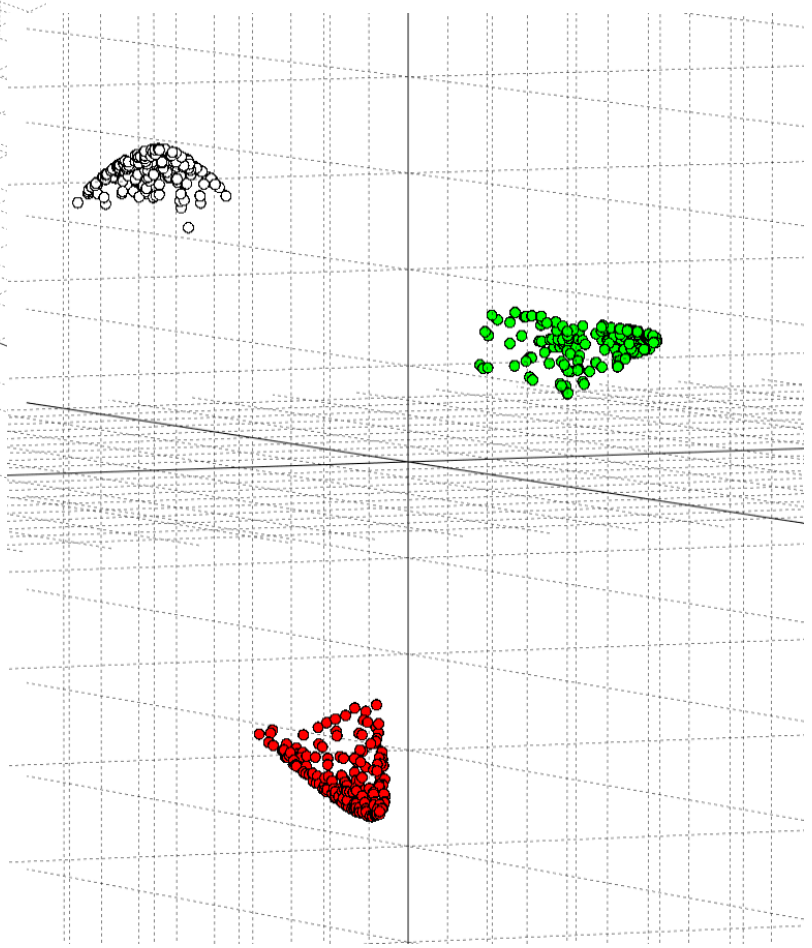
Key idea:

Some problems are made simpler
if you change the representation of the data



Result after kernel PCA

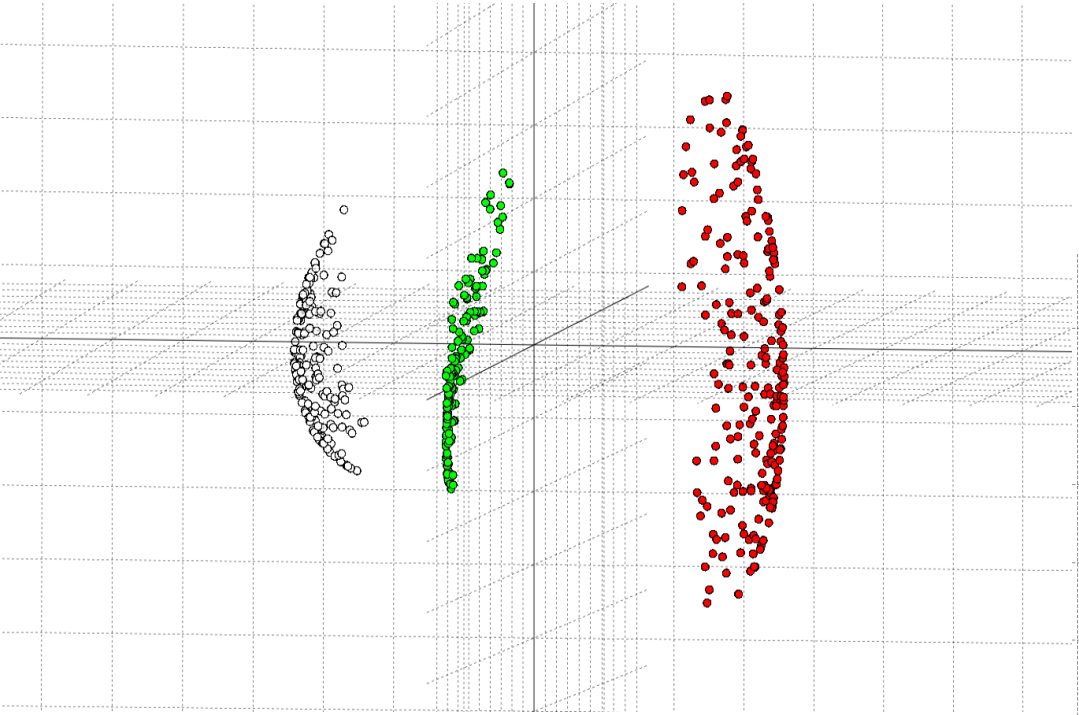




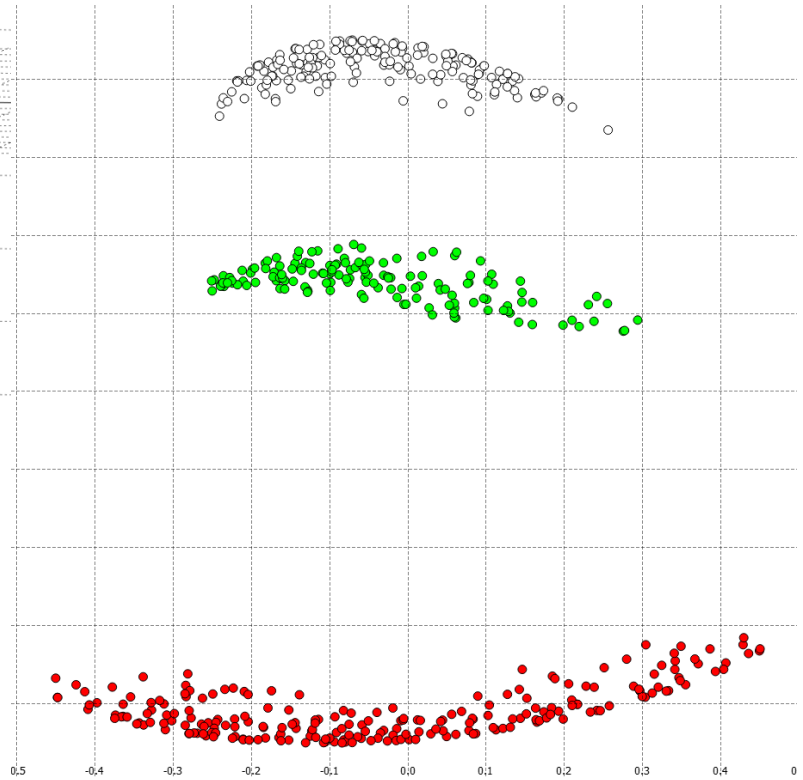
3D view on 3 eigenvectors

Places each group in a quadran

Result after kernel PCA

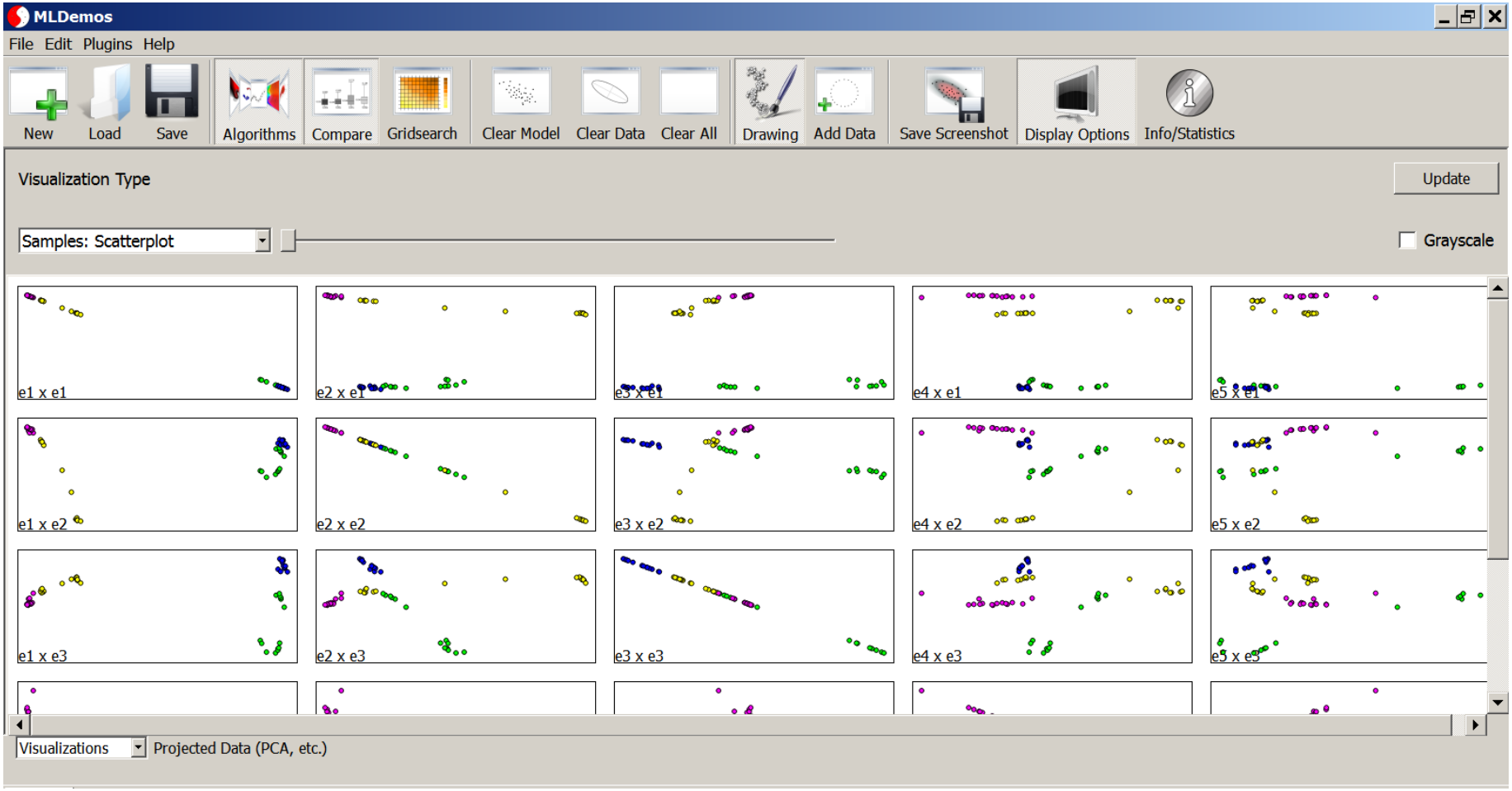


3D view on 3 eigenvectors



2D projections

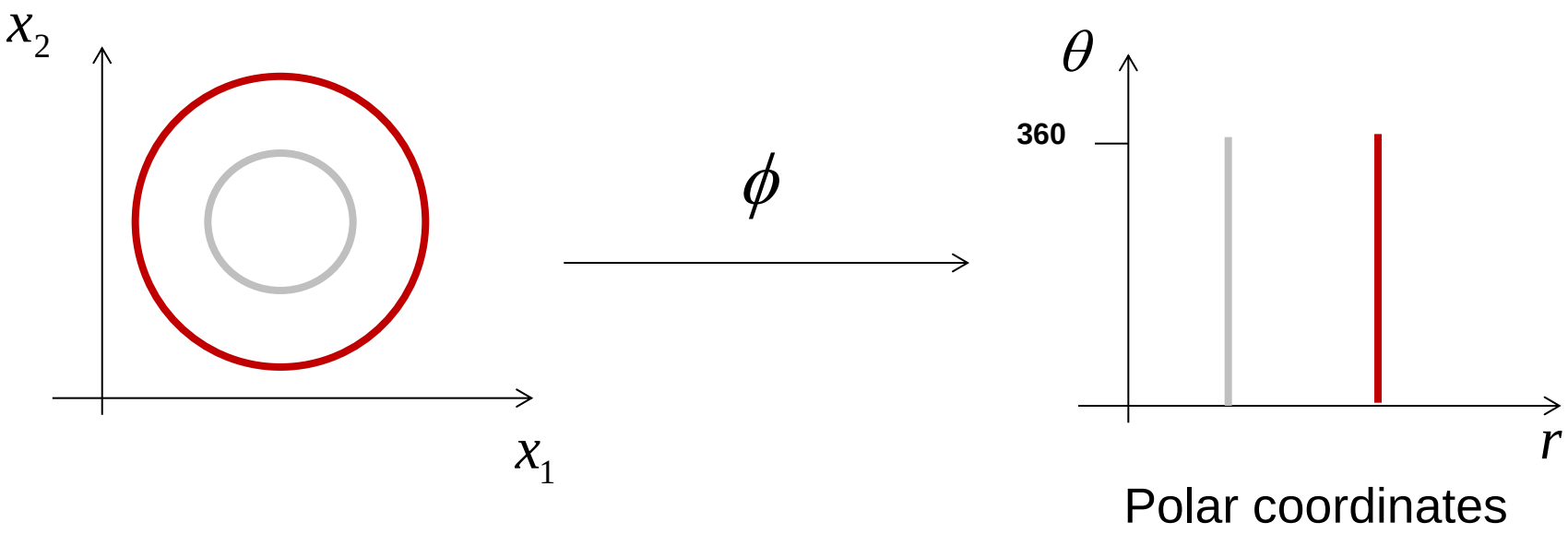
In large dimension?



What if we have many groups and they live in *N dimensions*, with *N>>1?*
Grouping may require many combination of projections
and can no longer be visualized

Kernels: intuition

How to separate the red class from the grey class?

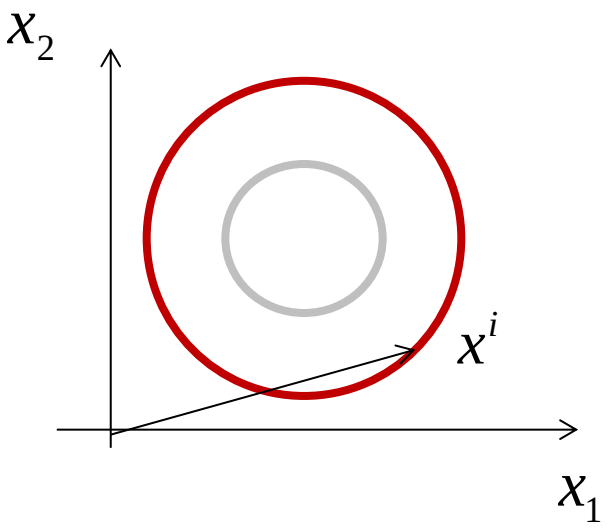


Polar coordinates

Data become linearly separable

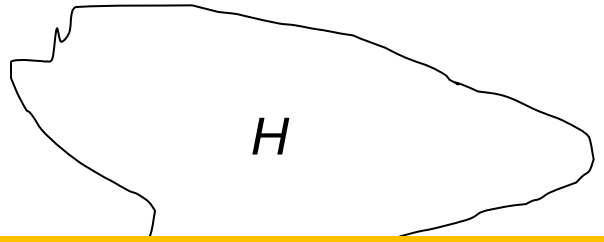
Kernels: formalism

Original Space



ϕ

Feature Space H



While the dimension of the original space is N, the dimension of the feature space may be greater than N! ➔ *X is lifted onto H*

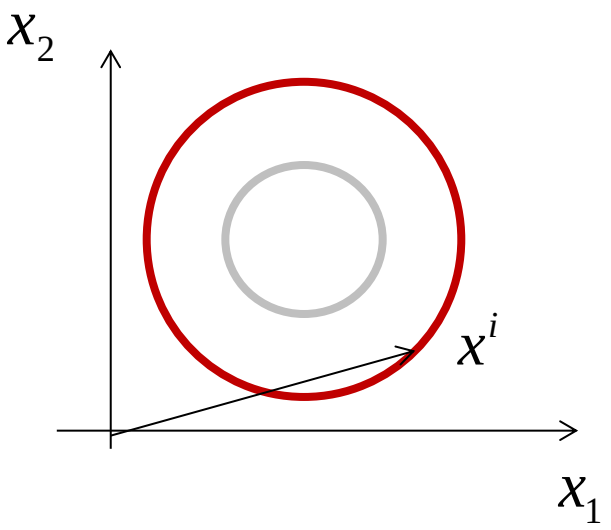
Send the data *X* into a *feature* space *H* through the nonlinear map ϕ.

$$X = \left\{ x^i \in \mathbb{R}^N \right\}^{i=1...M} \mapsto \phi(X) = \left(\phi\left(x^1\right), ..., \phi\left(x^M\right) \right)$$

Idea: In feature space, computation is simpler (it becomes a linear problem)

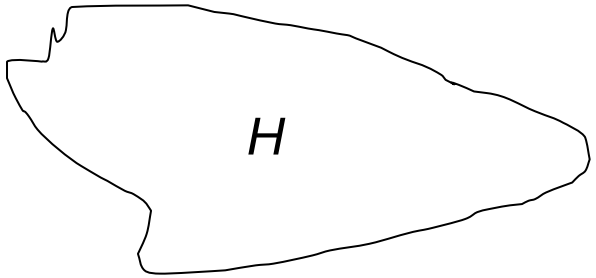
Kernels: formalism

Original Space



ϕ

Feature Space H



Determining ϕ is difficult → Kernel Trick

What is the kernel trick?

- Most algorithms in ML only require to compare relative distance across datapoints.

  → They do not need explicit coordinates of the datapoints.

- The relative distance relies often on computing the inner product: $\left\langle x^i, x^j \right\rangle = x^{iT} \cdot x^j$

  → No need to compute the transformation ϕ, if one expresses everything as a function of the inner product in feature space.

Define a *kernel function*: $k : X \times X \to \mathbb{R}$

$$k\left(x^i, x^j\right) \to \left\langle \phi\left(x^i\right), \phi\left(x^j\right) \right\rangle.$$

Apply linear transformation (PCA, linear regression, K-means) by using the kernel in place of the inner product.

The kernel function

- $k\left(x^i, x^j\right)$ defines a measure of similarity / distance across datapoints in feature space.
- It can extract features that are either common or that distinguish groups of datapoints.
- There exist several popular kernel functions in machine learning.
- To build an understanding of what feature they can extract, we will do a few exercises next.

Popular kernels

- **Gaussian / RBF Kernel (translation-invariant):**

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}, \quad \sigma \in \mathbb{R}$$

- **Homogeneous Polynomial Kernels:**

$$k(x, x') = \langle x, x' \rangle^p, \quad p \in \mathbb{N};$$

- **Inhomogeneous Polynomial Kernels:**

$$k(x, x') = \left(\langle x, x' \rangle + c\right)^p, \quad p \in \mathbb{N}, \; c \geq 0$$

Popular kernels

❖ **Linear Kernel:** $k(x, x') = x^T x'.$

❖ **Exponential/Laplacian Kernels:**

$$k(x, x') = e^{-\frac{\|x-x'\|}{2\sigma^2}}, \quad k(x, x') = e^{-\frac{\|x-x'\|}{\sigma}}, \quad \sigma \in \mathbb{R}.$$

❖ **Sigmoid kernel:**

$$k(x, x') = \tanh\left(ax^T x' + c\right), \quad a, c \in \mathbb{R};$$

See supplement posted on moodle for more examples of kernels.

Kernels: properties

The kernel function is a real-valued function with two arguments:

$k(x, x') : X \times X \to \mathbb{R}$

It is symmetric:

$k(x, x') = k(x', x)$

In some cases, it is non-negative:

$k(x, x') \geq 0.$

RBF, exponential and Laplacian kernels and even order Polynomial kernels

When k is non-negative and symmetric, then there is a Hilbert space of function on X for which k is a reproducing kernel. This is known as a *Mercer* kernel.

Kernels: properties

Sending data into feature space can increase the dimension of data features.

Consider inhomogeneous polynomial kernel with $p = 2$.

If $x \in \mathbb{R}^2$, we have $k(x, x') = \left(1 + x_1 x_1' + x_2 x_2'\right)^2$

$$= 1 + \left(x_1 x_1'\right)^2 + \left(x_2 x_2'\right)^2 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_1' x_2 x_2'$$

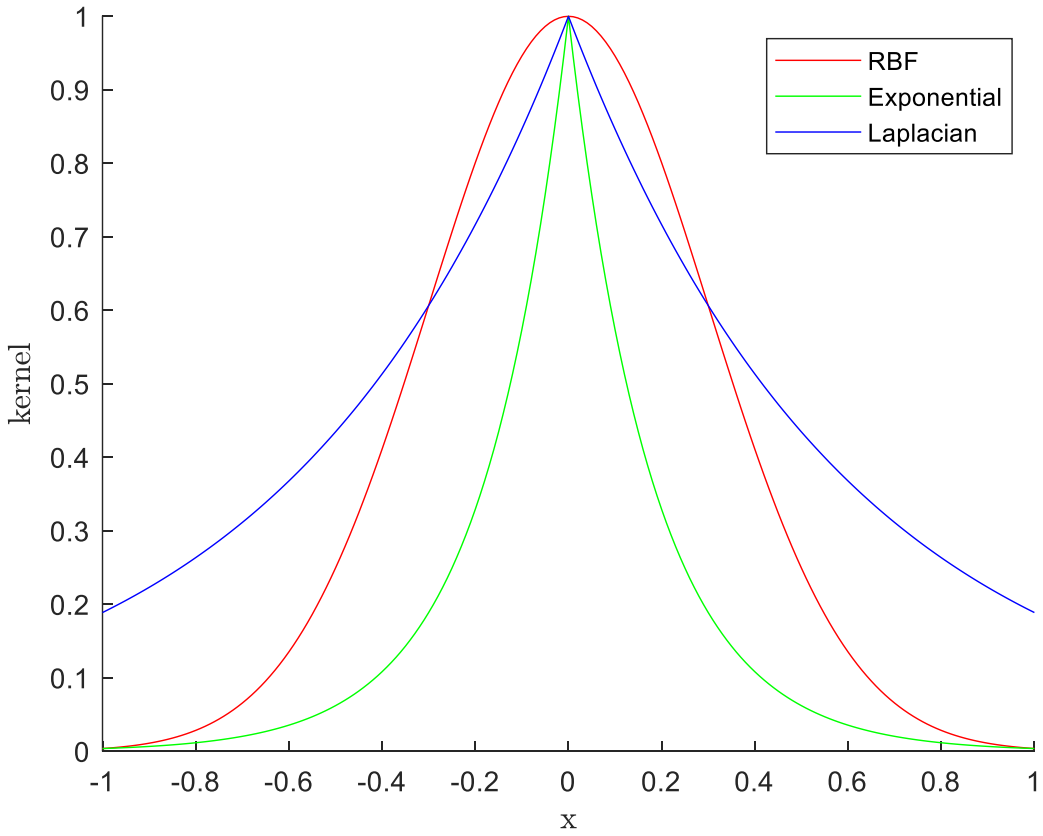
$$k(x, x') = \phi(x)^T \phi(x')$$

$$\phi(x) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2\right]^T \quad \phi(x) \in \mathbb{R}^6$$

Kernels: properties

RBF, exponential and Laplacian kernels

RBF $k(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$, $\sigma \in \mathbb{R}$ Exp: $k(x,x') = e^{-\frac{\|x-x'\|}{2\sigma^2}}$, Laplacian: $k(x,x') = e^{-\frac{\|x-x'\|}{\sigma}}$ or $e^{-\frac{|x-x'|}{\sigma}}$, $\sigma \in \mathbb{R}$.



Kernels: properties

Are the RBF, exponential and Laplacian kernels metrics?

$$k(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}, \quad \sigma \in \mathbb{R} \qquad k(x,x') = e^{-\frac{\|x-x'\|}{2\sigma^2}}, \quad k(x,x') = e^{-\frac{\|x-x'\|}{\sigma}}, \quad \sigma \in \mathbb{R}.$$

Condition 1: $k(x,x') = 0$ if and only if $x = x'$.

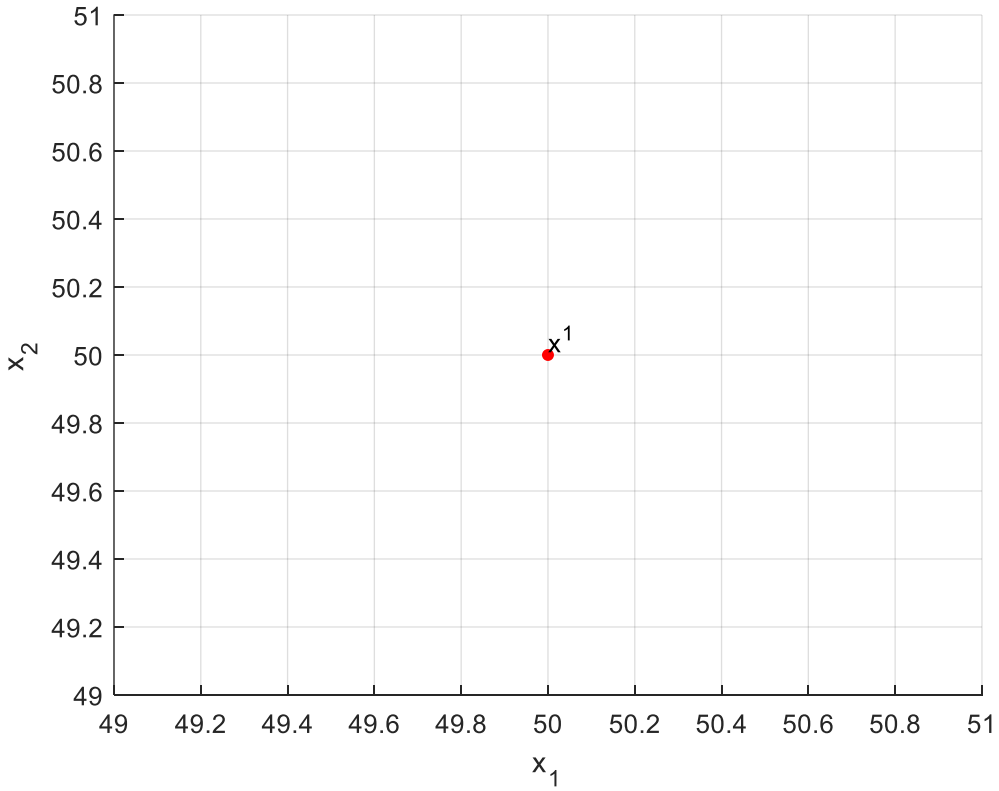
Not satisfied, but if we write m(x,x)=1- k(x,x), we get a metric.

Condition 2 (symmetry): $m(x^1, x^2) = m(x^2, x^1)$

Condition 3 (triangle inequality): $m(x^1, x^2) + m(x^2, x^3) \geq m(x^1, x^3)$

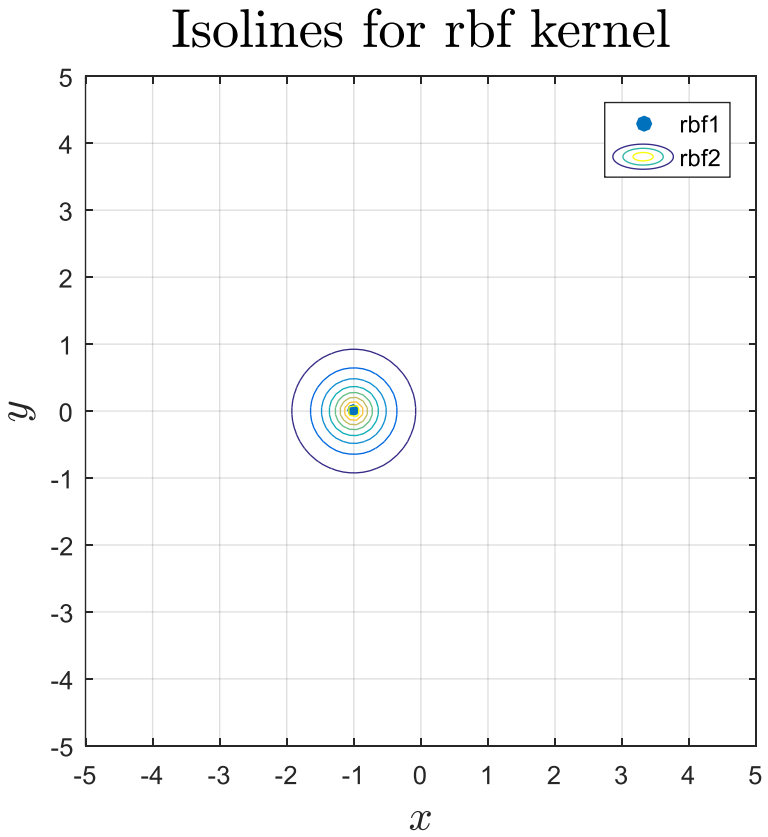
Kernels: Exercise 1.1

Using the RBF kernel: $k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$, $\sigma \in \mathbb{R}$, draw the isolines of the kernel for one datapoint x^1. Find all x, s.t. $k(x, x^1) = cst$.



Kernels: Solutions Exercise 1.1

RBF Kernel; M=1, i.e. 1 data point



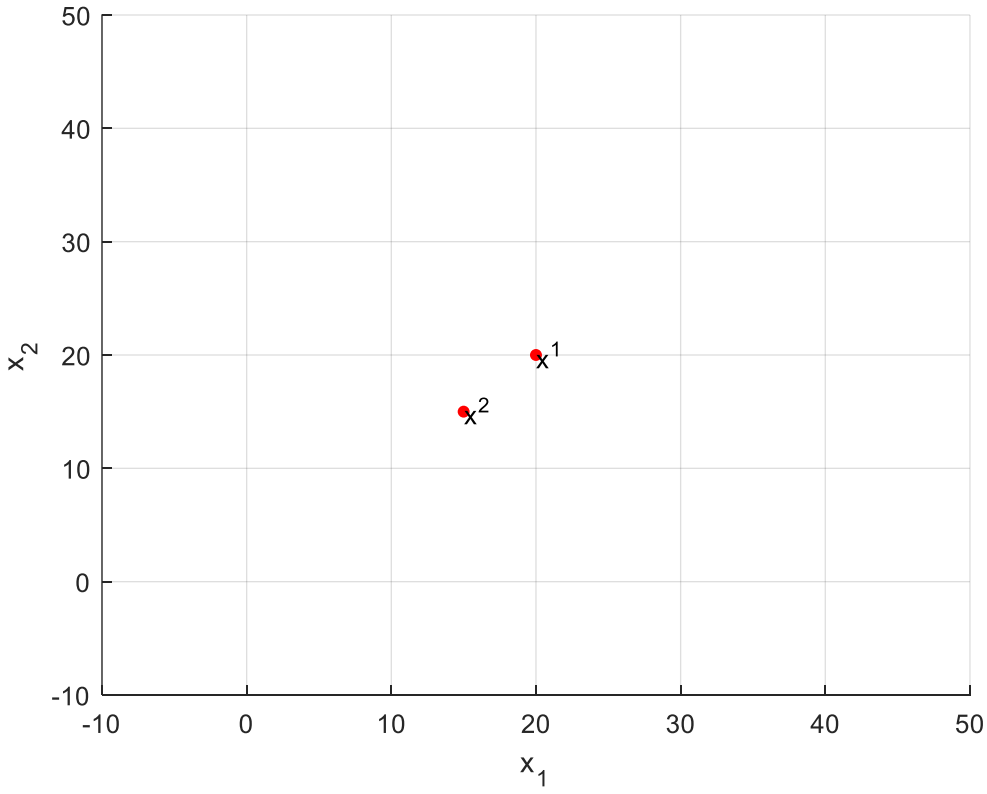
Kernels: Exercise 1.2

Using the RBF kernel: $k\left(x,x'\right)=e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$, $\sigma\in\mathbb{R}$, draw the isolines of the kernel for two datapoints, x^1, x^2:

a) Find all x, s.t. $k\left(x,x^1\right)+k\left(x,x^2\right)=cst.$

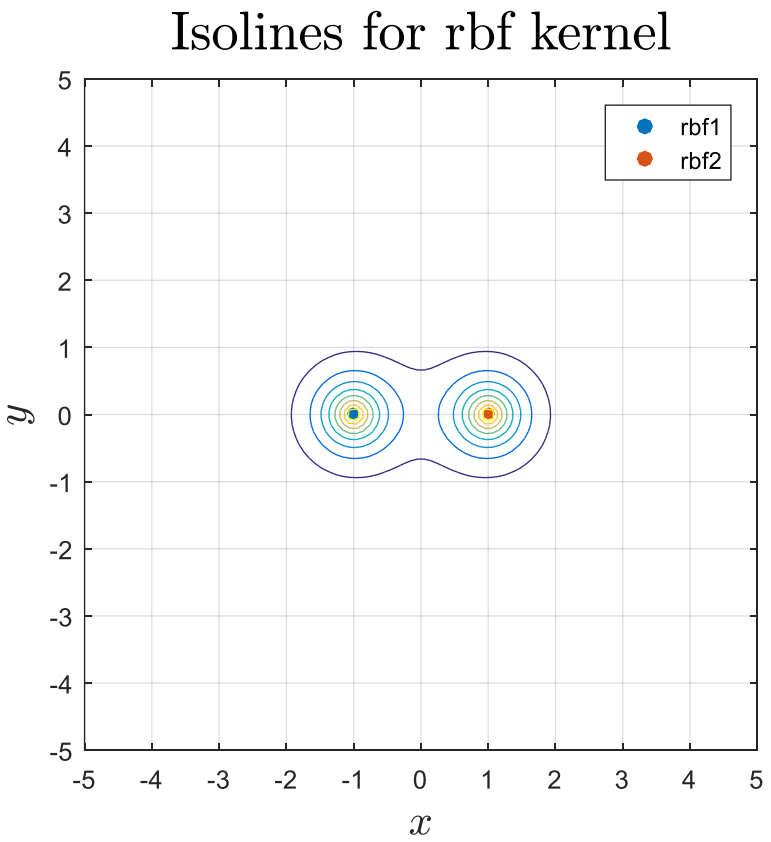
b) Find all x, s.t. $k\left(x,x^1\right)-k\left(x,x^2\right)=cst.$

c) Discuss the effect of σ on the isolines.

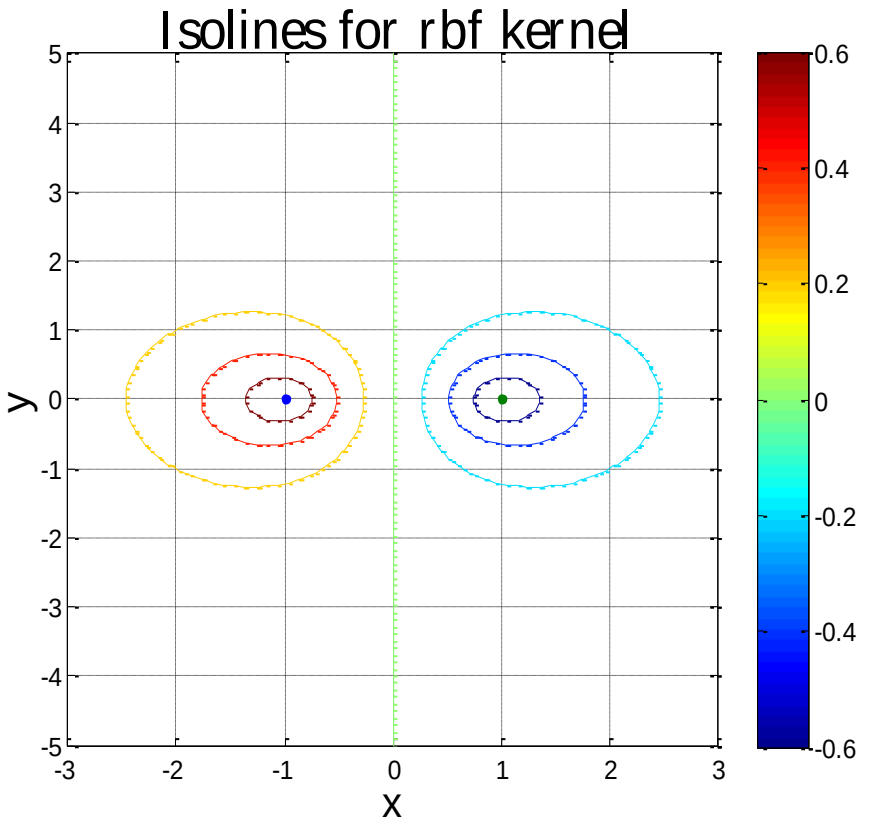


Kernels: Solutions Exercise 1.2

Gaussian Kernel; M=2, i.e. 2 data points



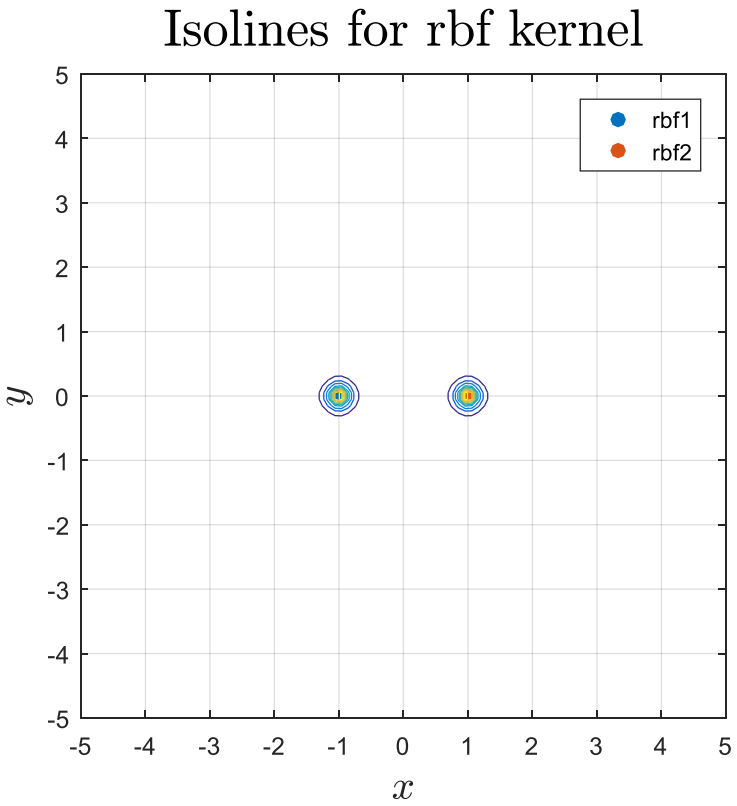
Solution when taking the sum of kernels



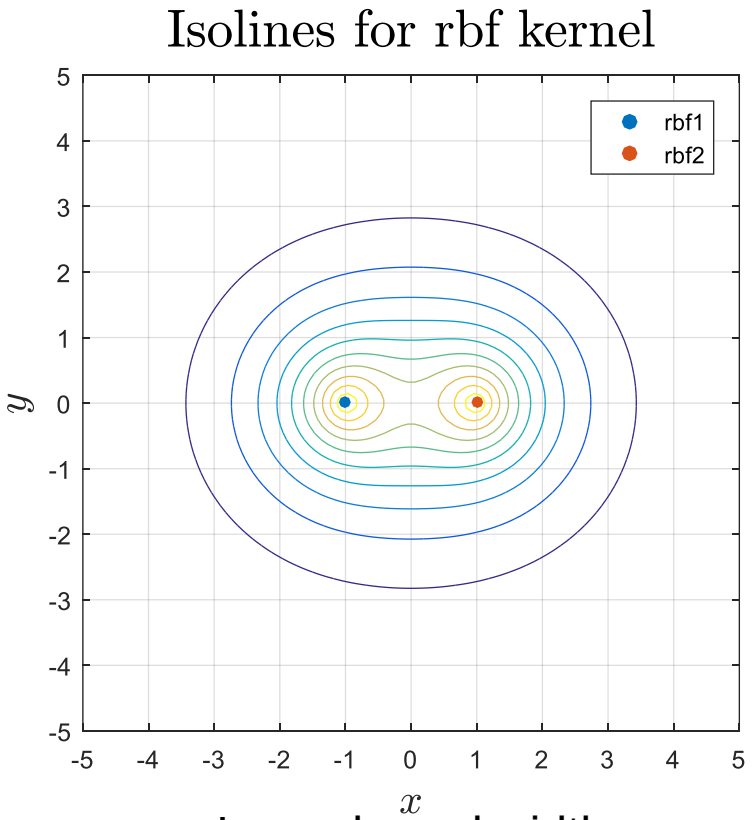
Solution when taking the difference of kernels

Kernels: Solutions Exercise 1.2

Gaussian Kernel; M=2, i.e. 2 data points



Small kernel width



Large kernel width

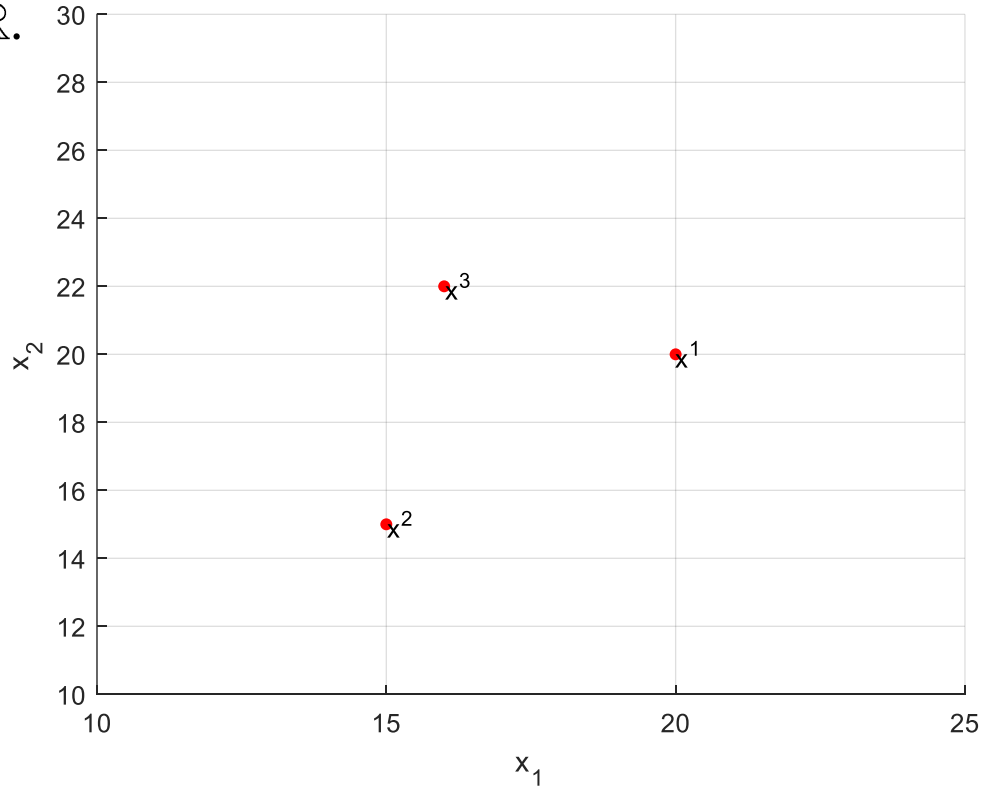
Solution when taking the sum of kernels

Kernels: Exercise 1.3

Using the RBF kernel: $k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$, $\sigma \in \mathbb{R}$, draw the isolines of the kernel for three datapoints

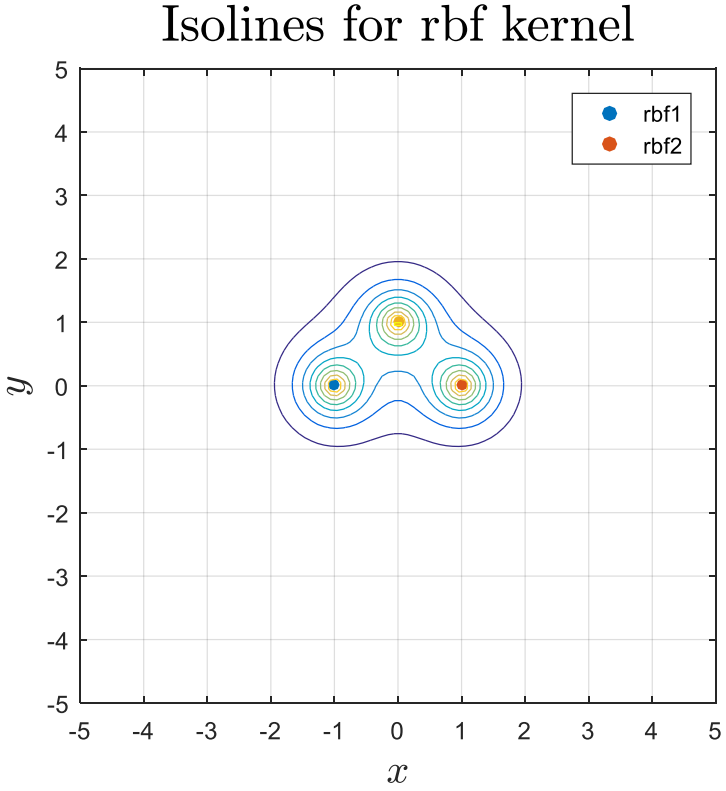
Compare RBF to Exponential & Laplacian kernels:

$k(x, x') = e^{-\frac{\|x-x'\|}{2\sigma^2}}$, $k(x, x') = e^{-\frac{|x-x'|}{\sigma}}$, $\sigma \in \mathbb{R}$.

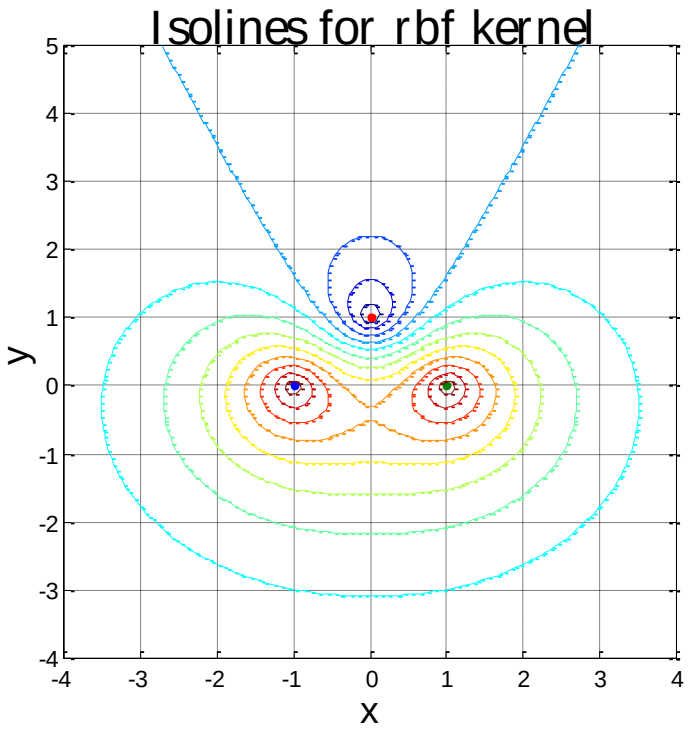


Kernels: Solutions Exercise 1.3

Gaussian Kernel; M=3, i.e. 3 data points



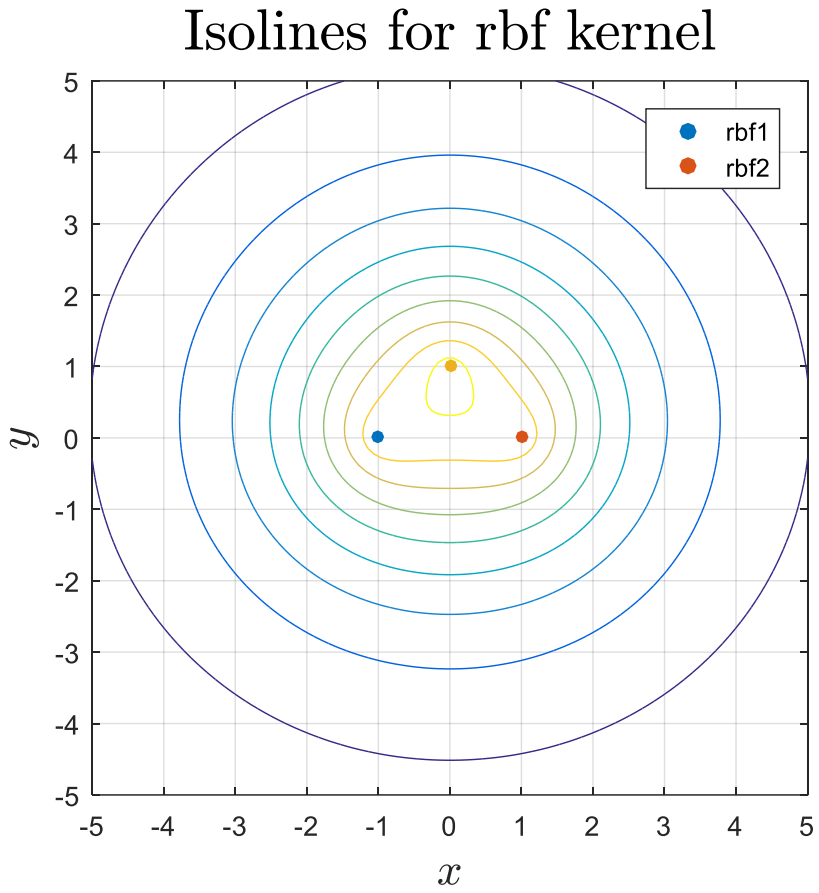
Solution when taking the sum of kernels



Solution when taking the sum for the two points below and the difference with the 3rd point.

Kernels: Solutions Exercise 1.3

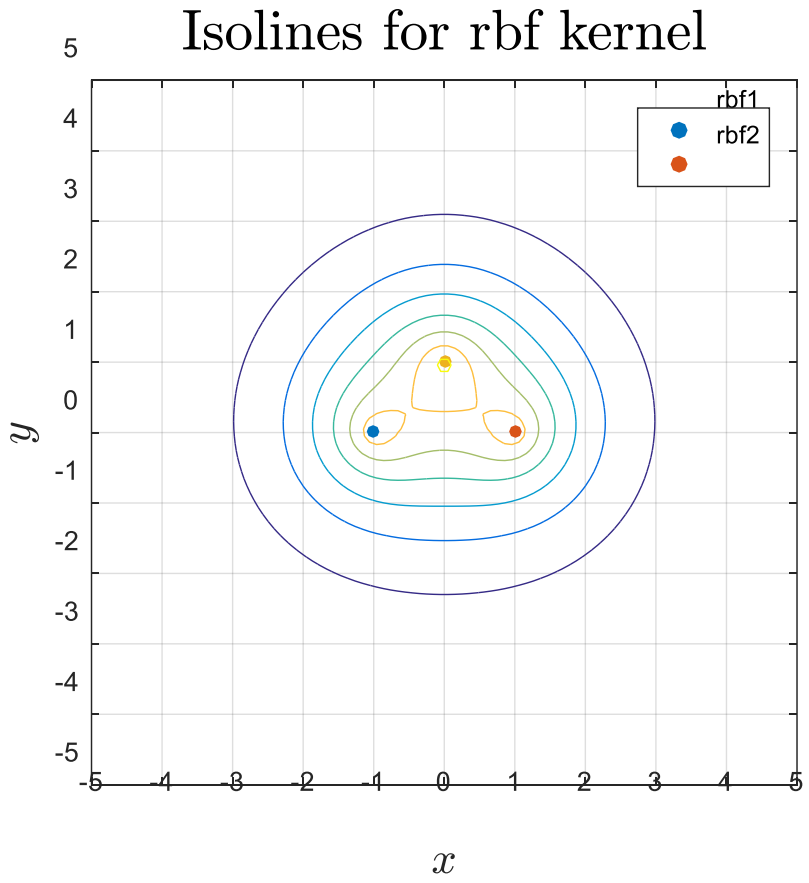
Gaussian Kernel; M=3, i.e. 3 data points



Effect of increasing the kernel width

Kernels: Solutions Exercise 1.3

Gaussian Kernel; M=3, i.e. 3 data points



Effect of increasing the kernel width

Kernels: Exercise 2.1

Using the homogeneous polynomial kernel:

$$k\left(x, x'\right) = \left\langle x, x' \right\rangle^{p}, \quad p \in \mathbb{N},$$

draw the isolines as in previous exercise for:

a) one datapoint

b) two datapoints

Discuss the effect of p on the isolines.

Kernels: Solutions Exercise 2.1

Given a datapoint *x'*, the polynomial kernel is given by:

$$\langle x, x' \rangle^p = \|x\| \|x'\| \cos(\theta)$$

This is the equation of a **projection** onto the vector *x'*.

The set of points *x,* solutions of the equation:

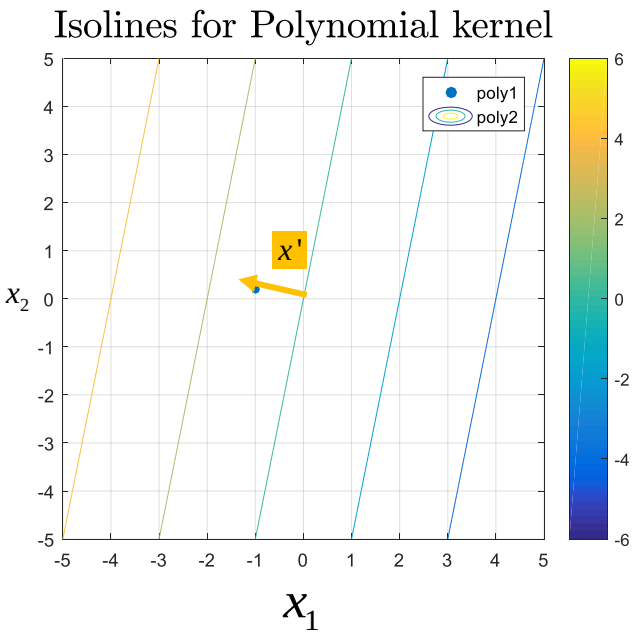
$$\langle x, x' \rangle^p = \|x\| \|x'\| \cos(\theta) = cst.$$

is an infinite set of lines perpendicular to the vector *x'.*

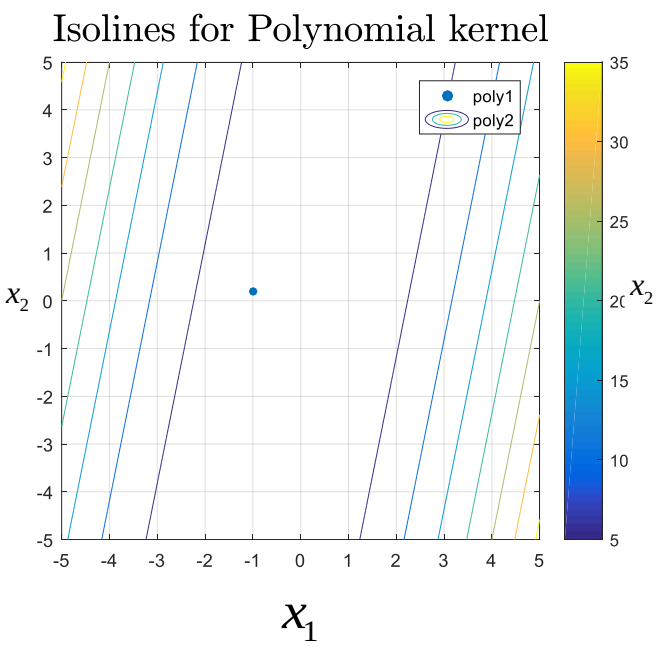
Kernels: Solutions Exercise 2.1

Polynomial Kernel; order p=1, 2, 3; M=1, i.e. 1 data points

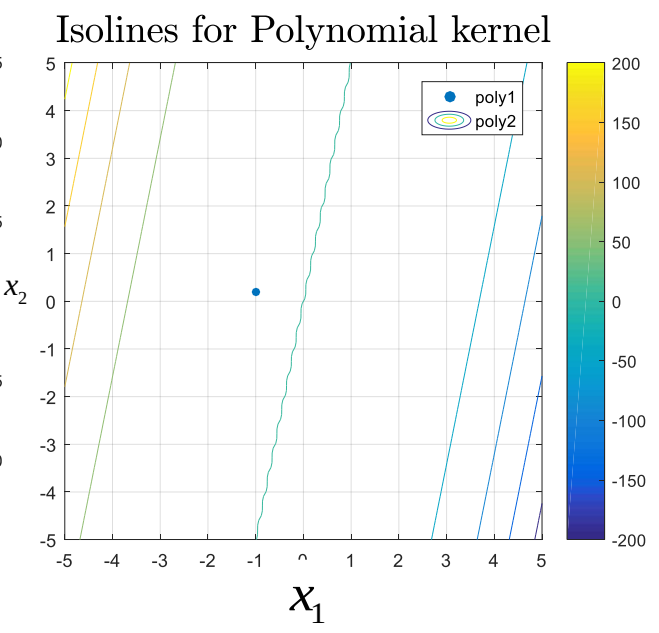
p=1



p=2



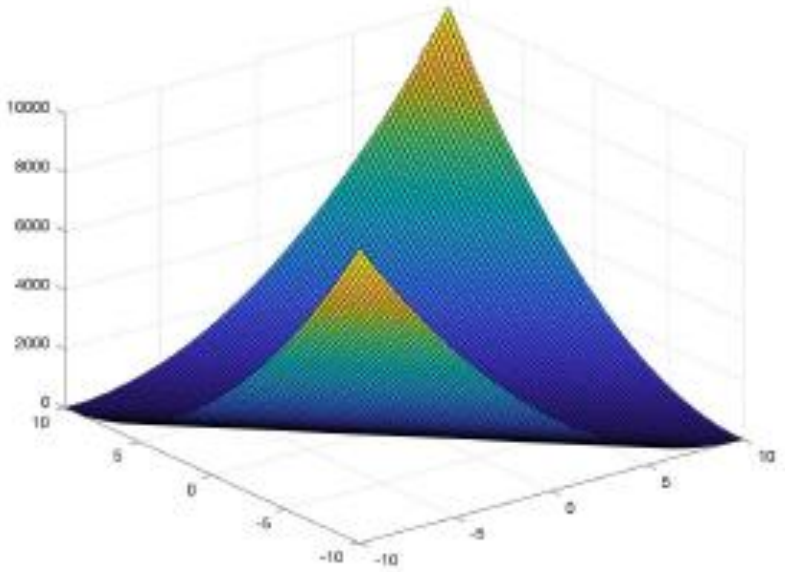
p=3



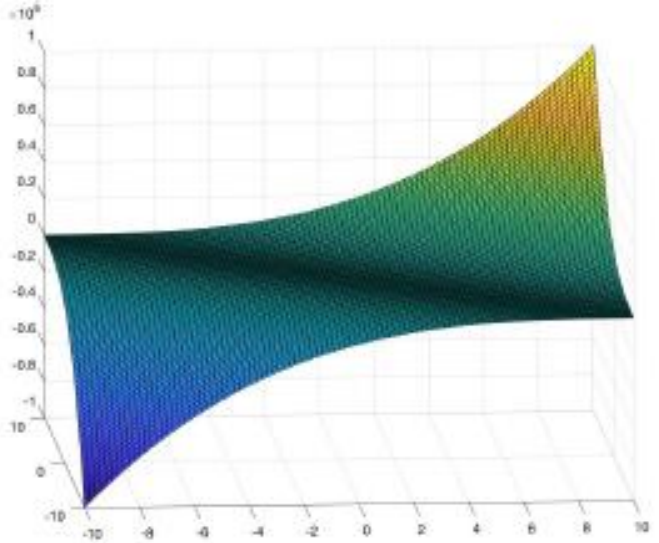
The isolines are lines perpendicular to the vector point from the origin. The order p does not change the geometry. It only changes the values of the isolines.

Kernels: Solutions Exercise 2.1

p=2



p=3



Kernels: Solutions Exercise 2.1

Given two vector datapoints x^1 and x^2, we have:

For $p=1$, $\left\langle x, x^1 \right\rangle^p + \left\langle x, x^2 \right\rangle^p = (x)^T x^1 + (x)^T x^2 = (x)^T \left(x^1 + x^2\right)$

Projections orthogonal to composition of the two vector points

Given two vector datapoints x^1 and x^2, we have:

For $p=2$, $\left\langle x, x^1 \right\rangle^2 + \left\langle x, x^2 \right\rangle^2 = \left((x^1)^T x\right)^2 + \left((x^2)^T x\right)^2$

If the datapoints are 2-dimensional, we have: $x = [x_1, x_2]^T$.

We expand and we get: $ax_1^2 + bx_2^2 + cx_1x_2$.

Equation of an ellipse

$a = (x_1^1)^2 + (x_1^2)^2$, $b = (x_2^1)^2 + (x_2^2)^2$ $\quad c = 4x_1^1x_1^2x_2^1x_2^2$.

Kernels: Solutions Exercise 2.1

Given two vector datapoints x^1 and x^2, we have:

For $p=1$, $\left\langle x, x^1 \right\rangle^p - \left\langle x, x^2 \right\rangle^p = (x)^T x^1 - (x)^T x^2 = (x)^T \left(x^1 - x^2\right)$

Projections orthogonal to composition of the two vector points

Given two vector datapoints x^1 and x^2, we have:

For $p=2$, $\left\langle x, x^1 \right\rangle^2 + \left\langle x, x^2 \right\rangle^2 = \left(\left(x^1\right)^T x\right)^2 - \left(\left(x^2\right)^T x\right)^2$

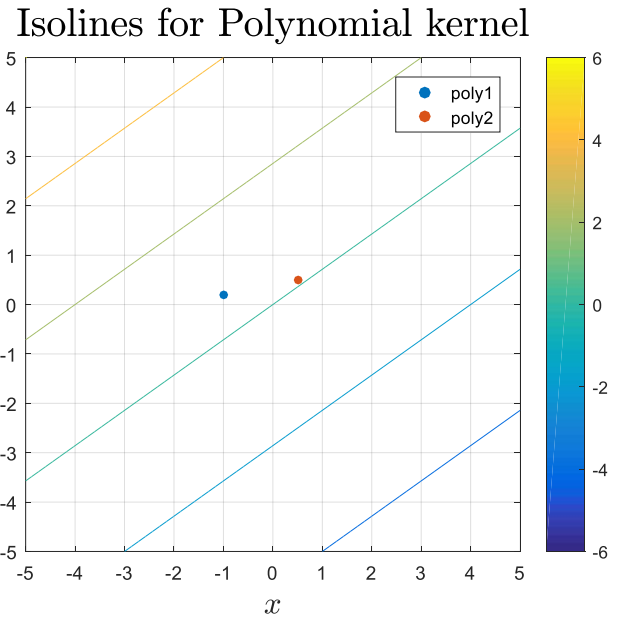
If the datapoints are 2-dimensional, we have: $x = [x_1, x_2]^T$.

We expand and we get: $ax_1^2 - bx_2^2 + cx_1x_2$

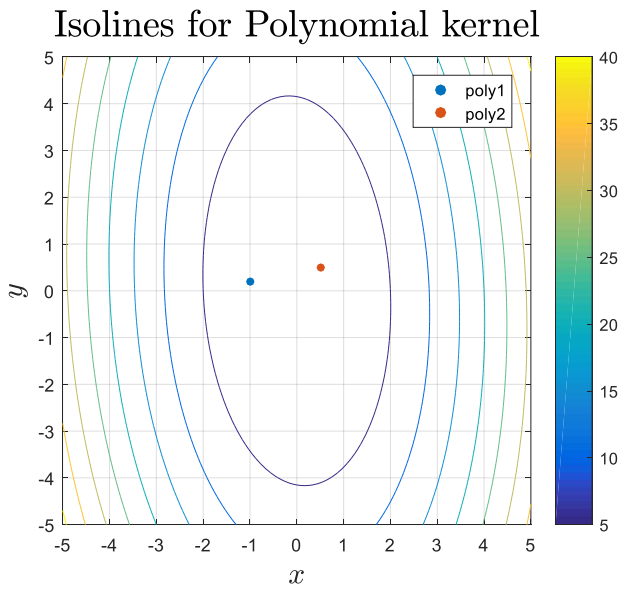
Equation of a hyperbola

Kernels: Solution Exercise II

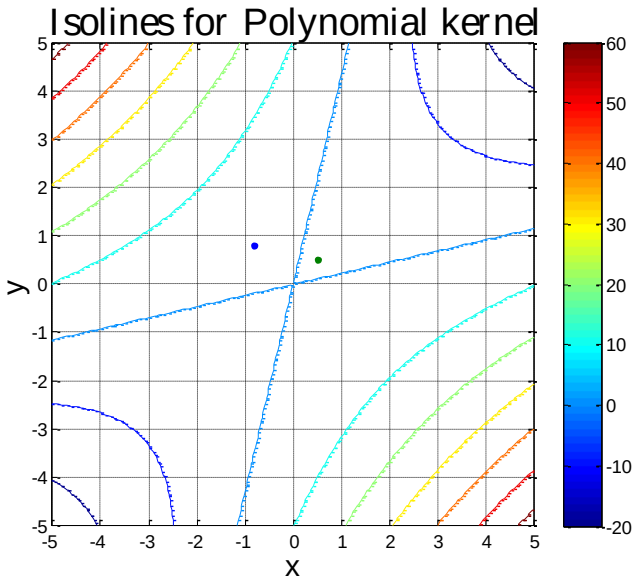
Homogeneous Polynomial Kernel; order p=1, 2, 3; M=2, i.e. 2 data points



p=1



p=2 (sum)

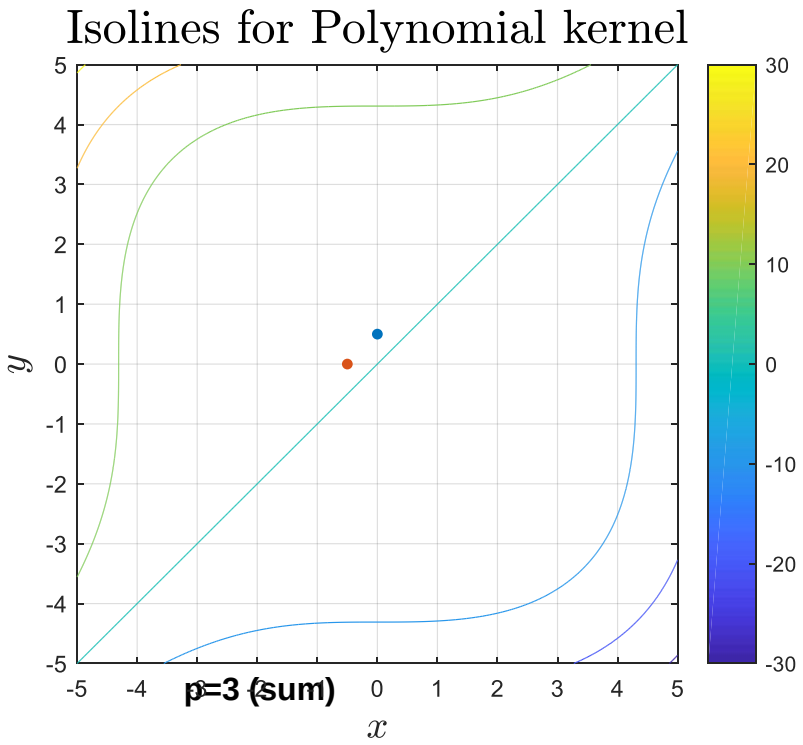


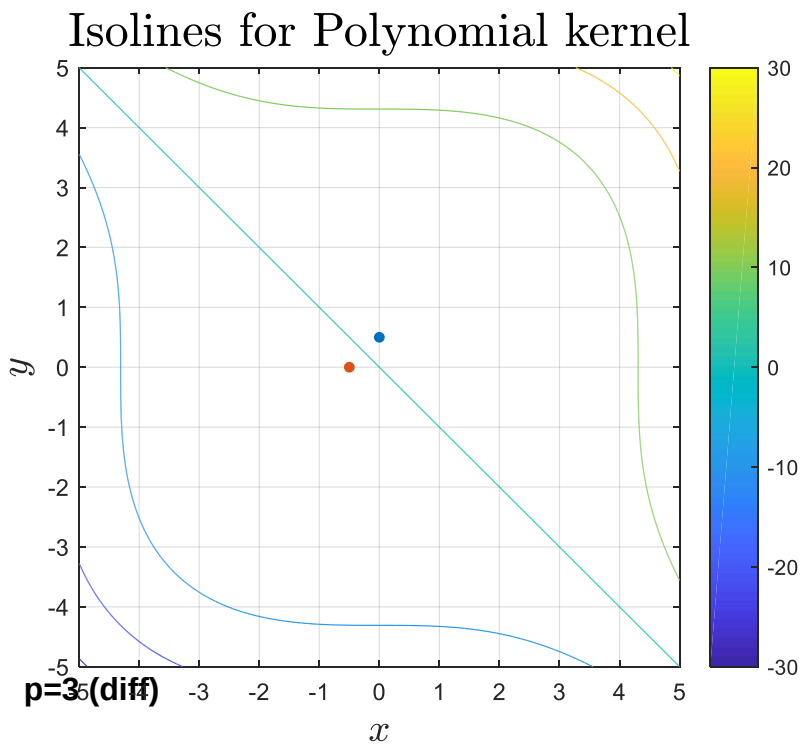
p=2 (difference)

The isolines are lines perpendicular to the combination of the vector points for p=1. With p=2 and the sum of kernels, we have an ellipse. For the difference, we have a hyperbola. The ellipse and hyperbolas are centered at the origin.

Kernels: Solution Exercise II

Homogeneous Polynomial Kernel; order p=1, 2, 3; M=2, i.e. 2 data points

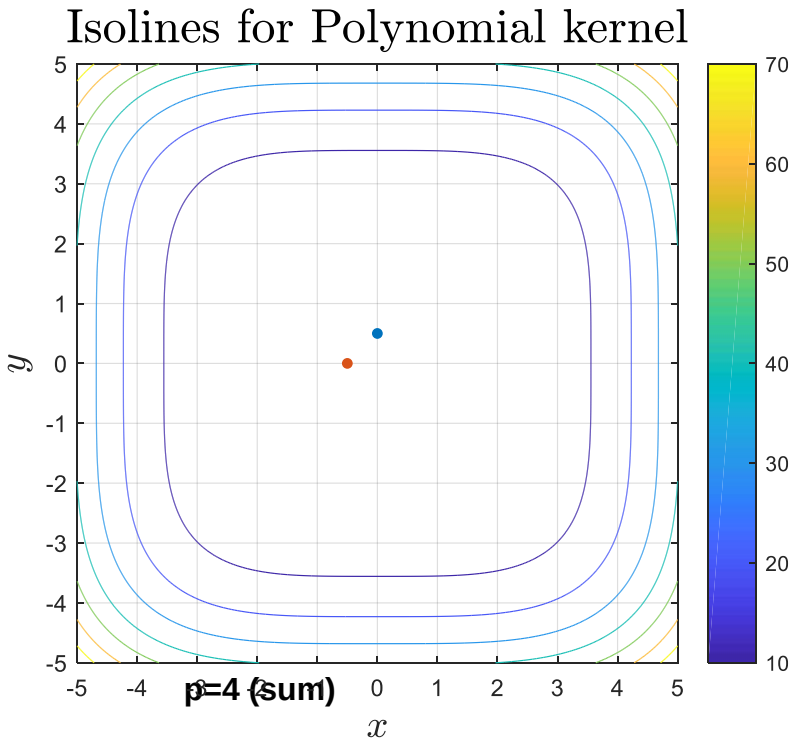


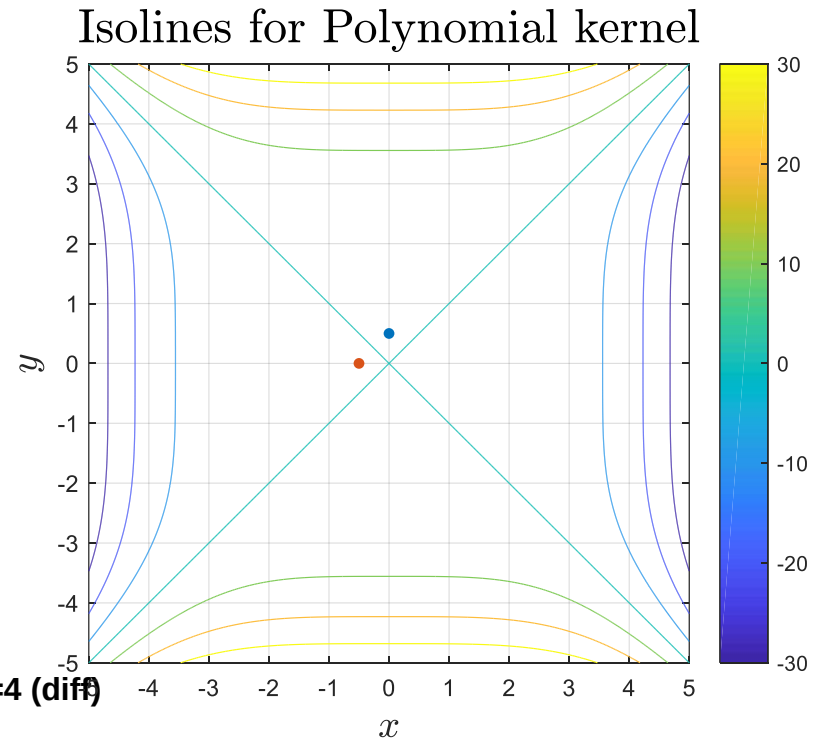


With higher orders for p, the solutions are equivalent to a superposition of polynomial kernels of different orders, i.e. superposition of ellipses and hyperbolas. They remain symmetrical around the origin, but may lead to more curvy shapes as illustrated above.

Kernels: Solution Exercise II

Homogeneous Polynomial Kernel; order p=1, 2, 3; M=2, i.e. 2 data points





With higher orders for p, the solutions are equivalent to a superposition of polynomial kernels of different orders, i.e. superposition of ellipses and hyperbolas. They remain symmetrical around the origin, but may lead to more curvy shapes as illustrated above.

Kernels: Exercise 2.2

Does the effect changes if you use the inhomogeneous polynomial kernel?

$$k(x, x') = \left(\langle x, x' \rangle + c\right)^p, \quad p, c \in \mathbb{N},$$

With 1 point: $\left(\langle x, x' \rangle + c\right)^p = \left(x^T x'\right)^P + c^P + \sum_{k=1}^{P-1} c^k \left(x^T x'\right)^{P-k}$

Same solution as for homogeneous kernel

Offset of value of isoline zero toward positive or negative quadran depending on value of c and p.

Same solution as for homogeneous kernel with a scaling by c

Kernels: Exercise 2.2

Does the effect changes if you use the inhomogeneous polynomial kernel?

$$k(x, x') = \left(\langle x, x' \rangle + c\right)^p, \quad p, c \in \mathbb{N},$$

With 2 points $\left(x^T x^1\right)^P + \left(x^T x^2\right)^P + c^P + \sum_{k=1}^{P-1} c^k \left(\left(x^T x^1\right)^{P-k} + \left(x^T x^2\right)^{P-k}\right)$

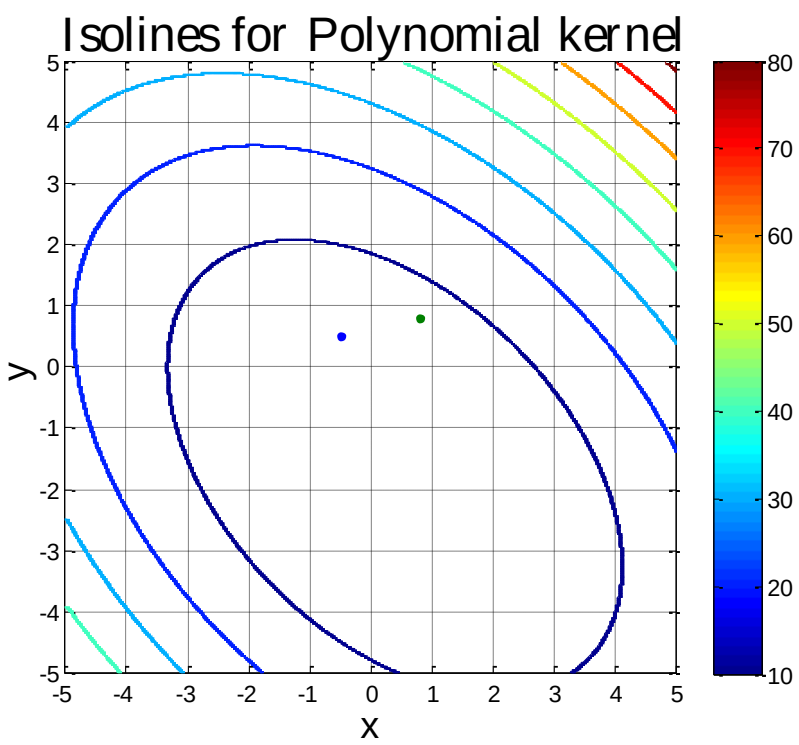
Same solution as for homogeneous kernel

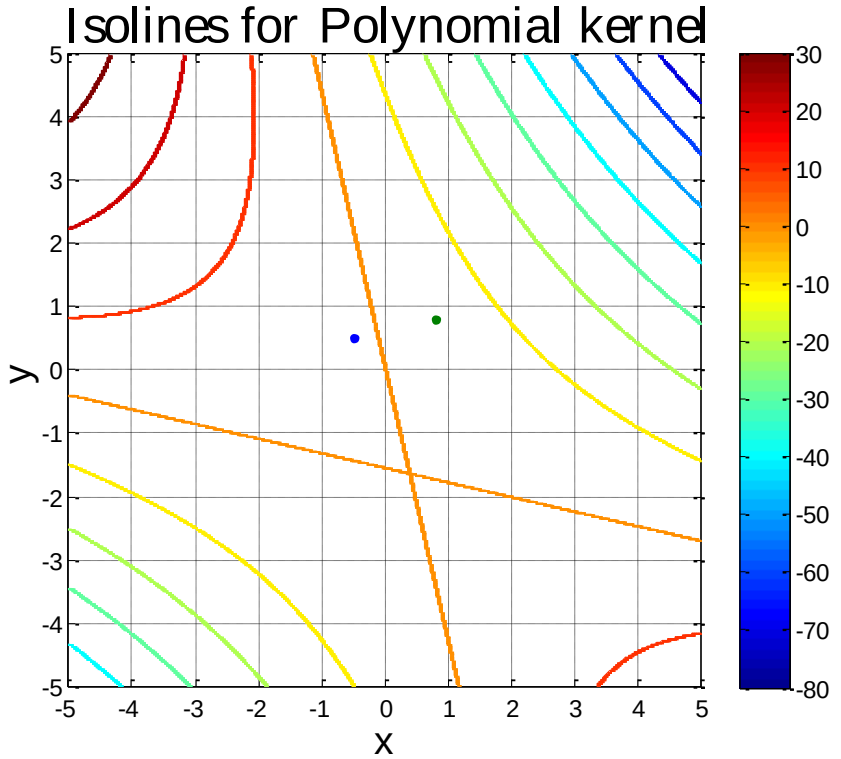
Offset of value of isoline zero toward positive or negative quadran depending on value of c and p.

Same solution as for homogeneous kernel with a scaling by c

Kernels: Solution Exercise II

Polynomial Kernel; order p= 2; c=1, M=3, i.e. 3 data points





The offset in the inhomogeneous polynomial kernels allows to shift the center of the ellipses and hyperbolas for p=2 and above. For p=1, it affects only the value of the isolines.

Kernels: Exercise 3

Kernels can also be created through addition of kernels and through multiplications.

a) Draw the isolines (around x')that result from adding a RBF and polynomial kernel, i.e.:

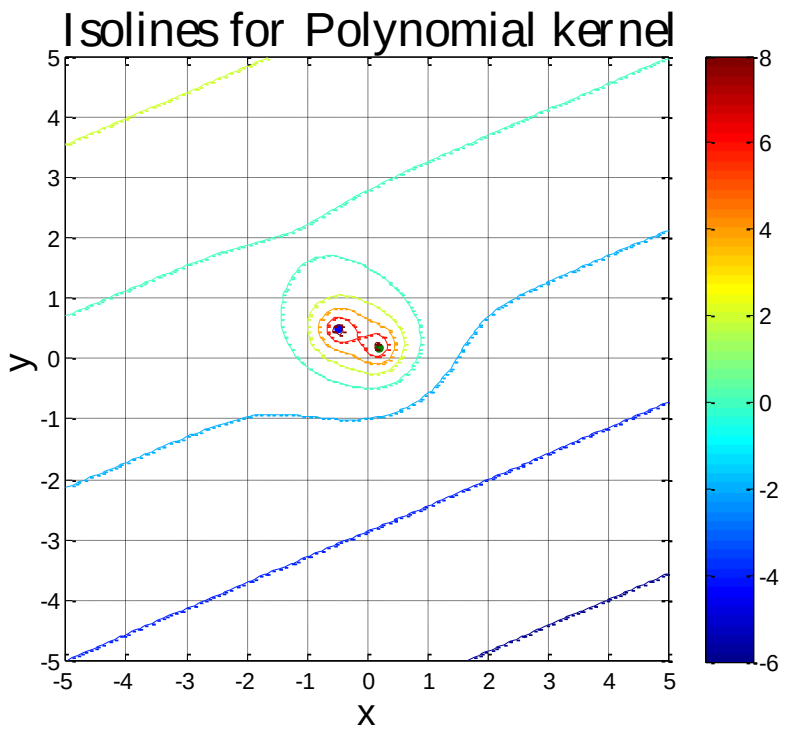
$$k(x, x') = k_{RBF}(x, x') + k_{Poly}(x, x')$$

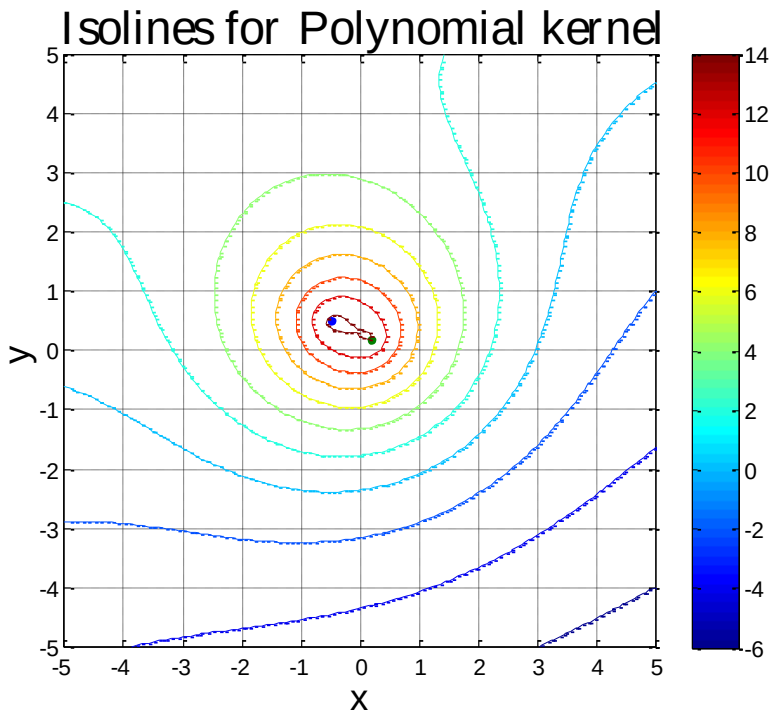
b) Draw the isolines that result from multiplying a RBF and polynomial kernel, i.e.:

$$k(x, x') = k_{RBF}(x, x') \cdot k_{Poly}(x, x')$$

Kernels: Solutions Exercise 3

RBF kernel σ=0.2 (left) / 0.8 (right) and Polynomial Kernel, p=1, c=-1, M=2.

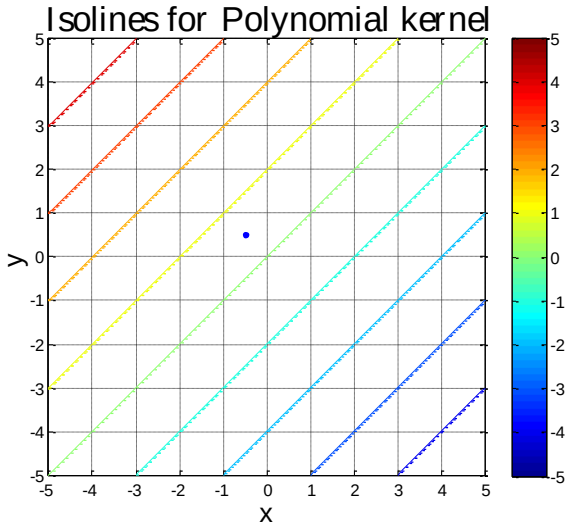


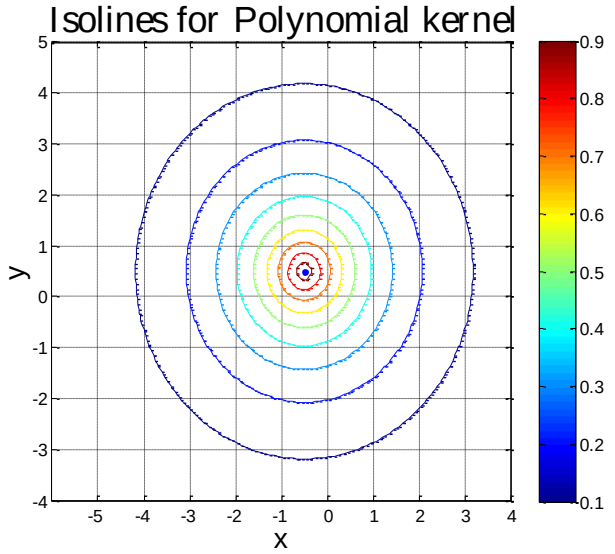


$$k(x,x') = 10k_{RBF}(x,x') + k_{Poly}(x,x')$$

Kernels: Solutions Exercise 3

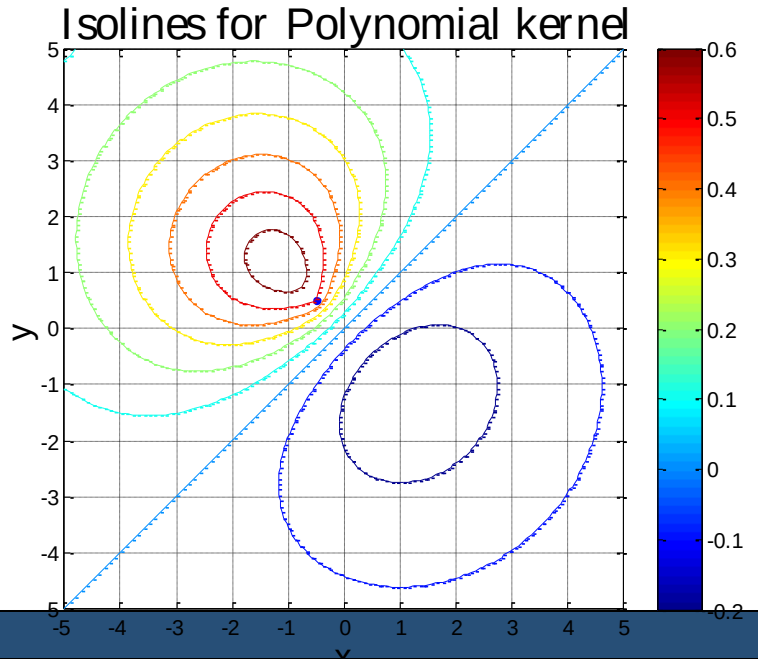
2 Original kernels





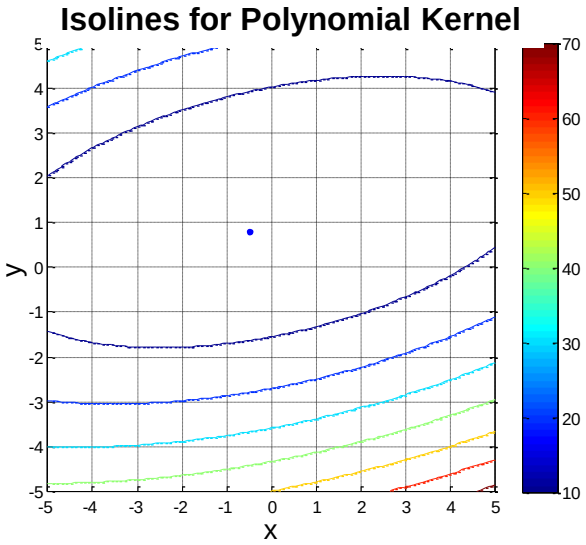
Result of the multiplication:

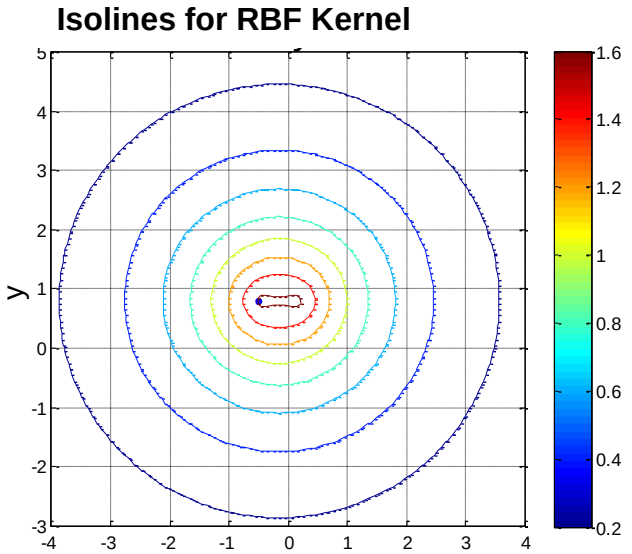
$$k(x, x') = k_{RBF}(x, x') \cdot k_{Poly}(x, x')$$



Kernels: Solutions Exercise 3

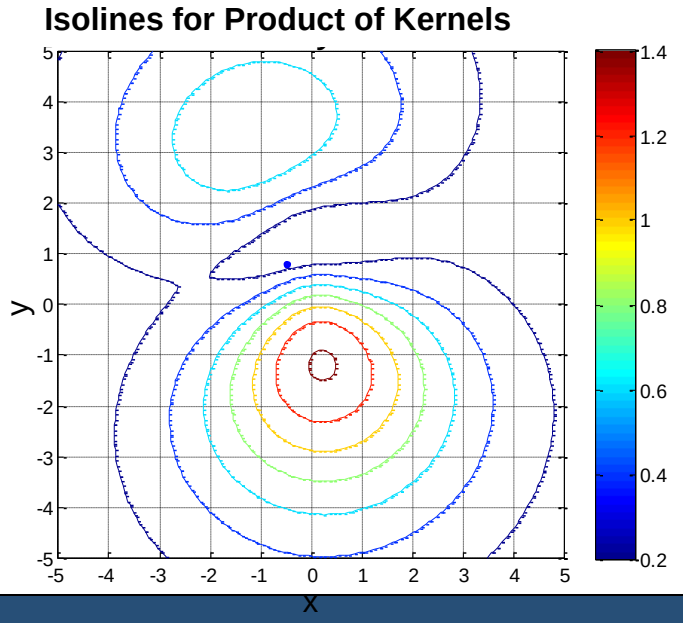
2 Original kernels





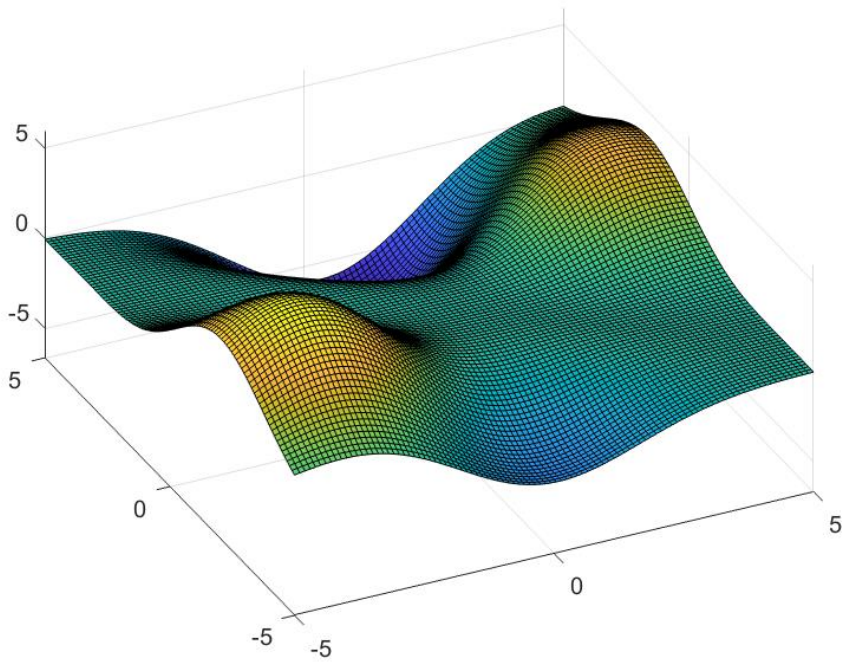
Result of the multiplication:

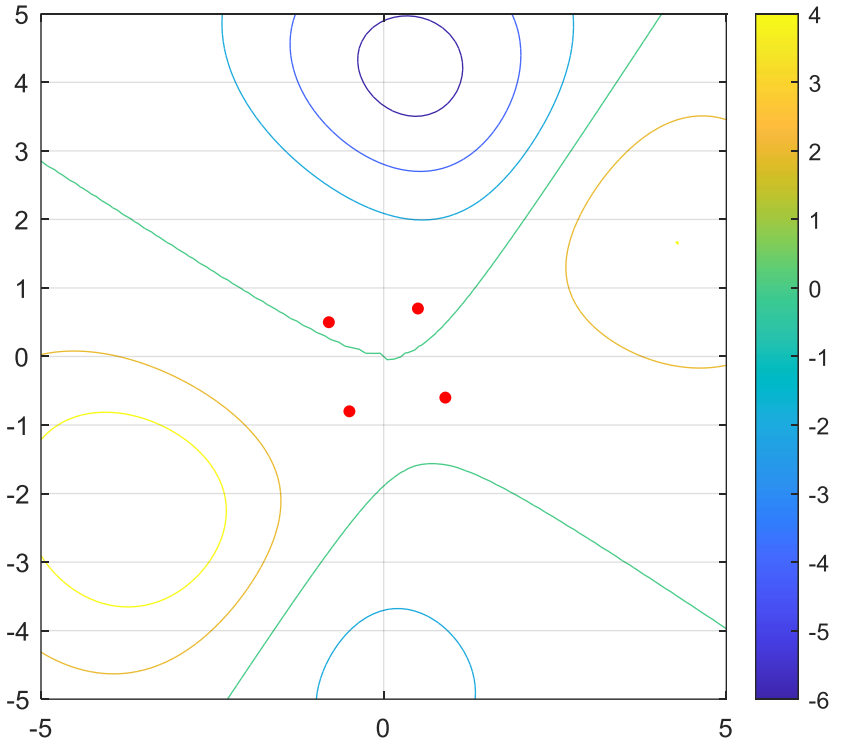
$$k(x, x') = k_{RBF}(x, x') \cdot k_{Poly}(x, x')$$



Kernels: Solutions Exercise 3

4 datapoints



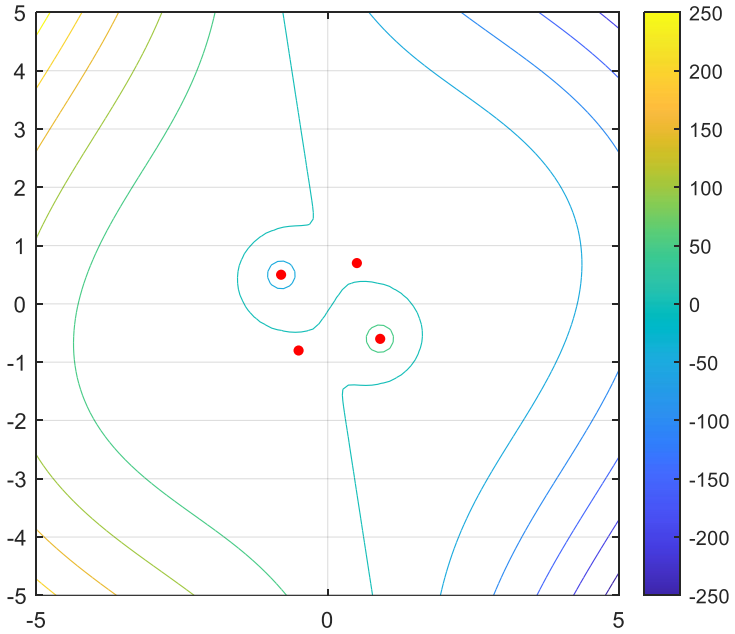


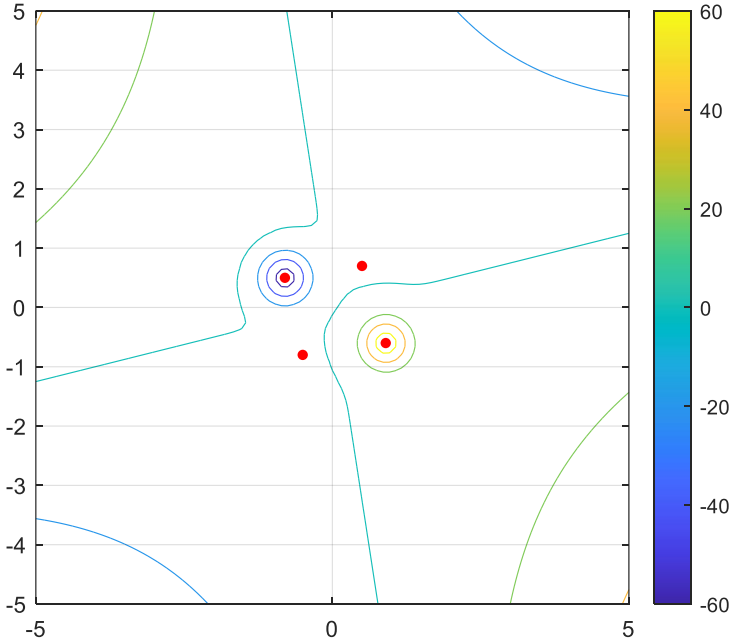
Composition of kernels can allow to create a variety of deformation of the space

Kernel – composition of 4 datapoints $k_{RBF}(x, x^1) \cdot k_{Poly}(x, x^2) + k_{RBF}(x, x^3) \cdot k_{Poly}(x, x^4)$

Kernels: Solutions Exercise 3

4 datapoints





Composition of RBF and ellipses / hyperbolas from Polynomials

Kernels: Exercise 2.3

Two other relatively popular kernels are

the linear kernel: $k(x, x') = x^T x'$.

the cosine kernel: $k(x, x') = \frac{x^T x'}{\|x\|\|x'\|}$.

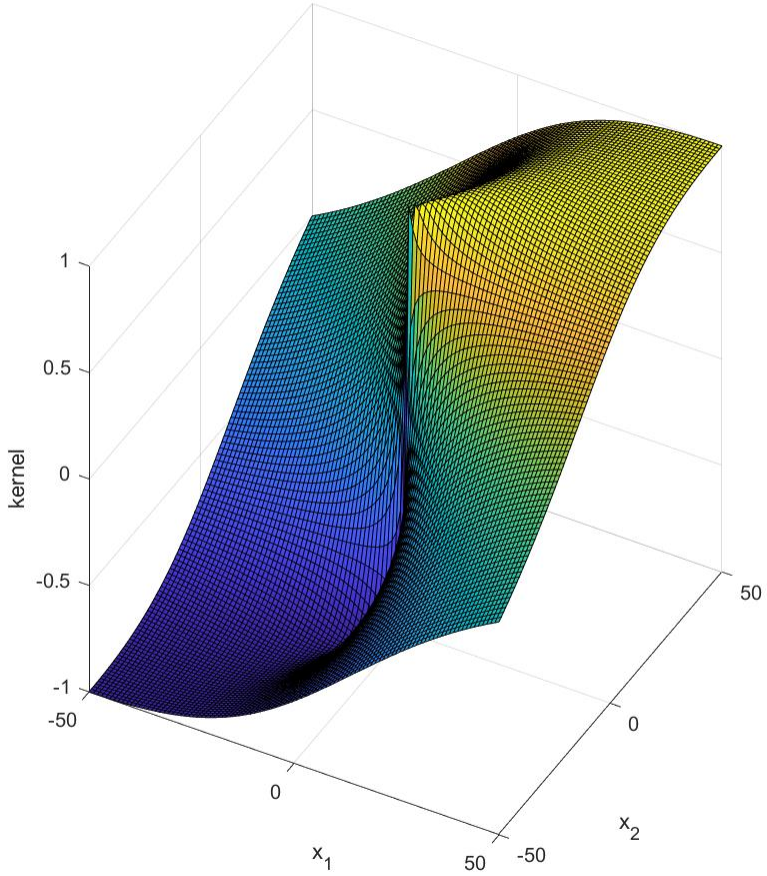
Draw the isolines when using cosine kernel

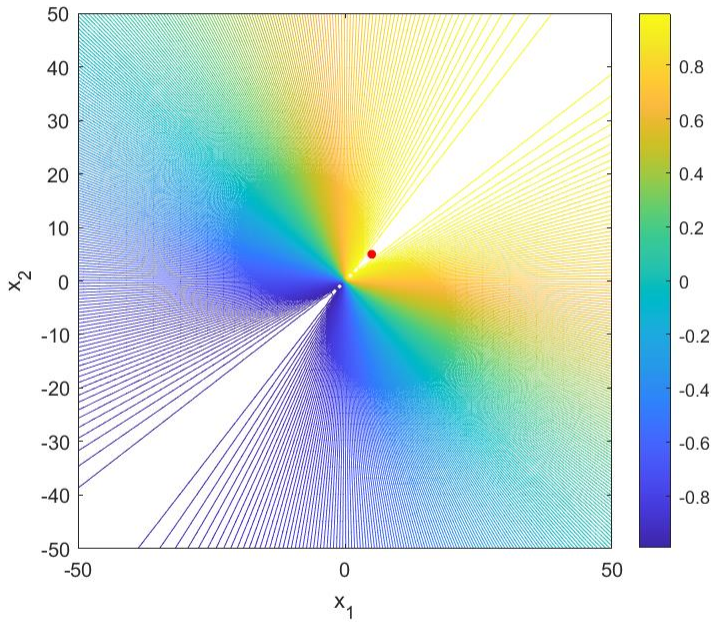
$$k(x, x') = \frac{x^T x'}{\|x\|\|x'\|}.$$

Which group of points cannot be separated by these kernels?

Kernels: Solutions Exercise 2.3

The cosine kernel: $k(x, x') = \frac{x^T x'}{\|x\|\|x'\|} = \cos(\prec(x, x'))$





Kernels: Solutions Exercise 2.3

This linear kernel is the special case of polynomial of order 1.
It cannot separate points that can be reached by a single line.
It can separate all other cases.

The cosine kernel cannot separate points that are colinear (vector pointing from the origin to the point) and in the same quadran.
It can hence not separate groups of points homogeneously distributed in the same quadrans.
This is due to the fact that the cosine kernel is insensitive to the distance to the origin unlike the linear kernel.
This can however be a strength as we will see with kernel K-means.

These concepts will be reused in kernel K-means lecture

Example of application of the linear kernel

Bags of words:

[machine, learning, kernel, rbf, robot, vision, dimension, blue, speed,...]

You want to group webpages with common groups of words.

Set $x \in \mathbb{N}^{1000}$ with each entry in x set to 1 if the word is present else zero.

E.g. $x^1 = [1\ 1\ 1\ 0\ 0\ 0....]^T$ contains the words machine learning and kernel and nothing else.

Features live in low-dimensional space (common group of webpages have a low number of combination of words):

$$k\left(x^i, x^j\right) = \sum_k x_k^{iT} x_k^j = x_1^{iT} x_1^j + x_2^{iT} x_2^j + x_3^{iT} x_3^j + x_4^{iT} x_4^j + \ldots$$

The isoline $k\left(x^1, x^j\right) = 3$ delineate the set of webpages that share the same set of three keywords as x^1.

Example of application of the linear kernel

Sequence of strings (e.g genetic code):

[IPTSLQDVBUV,...]

Want to group strings with common subgroups of strings.

Set $\phi(x), x \in \mathbb{N}^{1000}$ the number of times sub-string x appears in the string word.

One can apply the linear kernel with same encoding as in previous bags of words examples.

Using cosine kernel can allow to delineate webpages that share the same keywords irrespective of their frequency.

Kernels: Summary

- Kernels are real-valued, symmetric functions.
- Kernels represent the inner product across projection of data in feature space.

- The most popular is the RBF kernel. It enables to group datapoints. The tightness of the grouping depends on the value of the hyperparameter σ.
- The polynomial and cosine kernels provide a geometric division of the space that embeds symmetry across the origin or an offset.

- Kernels in ML are useful because they allow to reduce computation of complex non-linear problem to simpler forms applied to linear problems.

- Determining the right kernel is difficult in practice. One usually determines the best choice through crossvalidation to determine the kernel and the best hyperparameters.
- There exist methods to learn the kernel (see course on Gaussian Process and papers to read in class).