

# Núcleos Reprodutores em Matemática e Engenharia

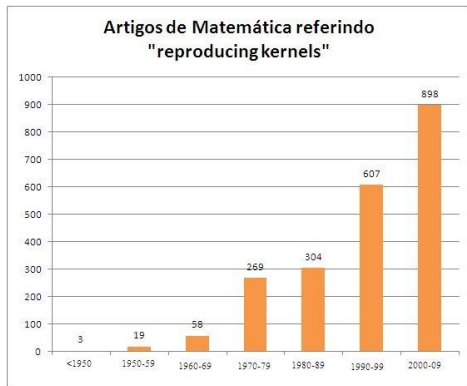
Jorge Buescu  
FCUL  
CMAF

Ciência 2009  
30 de Julho de 2009

# Artigos sobre Reproducing Kernels

Núcleos  
Reprodutores  
em  
Matemática e  
Engenharia

Jorge Buescu  
FCUL  
CMAF



1950: N. Aronszajn, *Theory of reproducing kernels*, TAMS **68**, 3, 337-404.

# Caracterização de RKHS's

## Definição (Matriz de Moore)

Seja  $E \neq \emptyset$ .  $K : E \times E \rightarrow \mathbb{C}$  é uma matriz de Moore se

$$\sum_{i,j} K(x_i, x_j) \xi_i \bar{\xi}_j \geq 0 \quad \forall x_i, x_j \in E, \xi_i, \xi_j \in \mathbb{C}.$$

## Definição (RKHS)

Seja  $E \neq \emptyset$  e  $\mathcal{H}$  um espaço de Hilbert de funções em  $E$ .

$K : E \times E \rightarrow \mathbb{C}$  diz-se um *núcleo reprodutor* em  $\mathcal{H}$  se

$$\forall y \in E \quad K(x, y) \in \mathcal{H} \text{ como função de } x, \quad (0.1)$$

$$\forall y \in E \quad \forall f \in \mathcal{H} \quad f(y) = \langle f(x), K(x, y) \rangle. \quad (0.2)$$

# Exemplos

## Teorema (Moore-Aronszajn)

*$K$  é uma matriz de Moore em  $E$  se e só se existe um único espaço de Hilbert  $\mathcal{H}$  de funções de  $E$  tal que  $K$  é um núcleo reprodutor em  $\mathcal{H}$ .*

# Exemplos

## Teorema (Moore-Aronszajn)

*$K$  é uma matriz de Moore em  $E$  se e só se existe um único espaço de Hilbert  $\mathcal{H}$  de funções de  $E$  tal que  $K$  é um núcleo reprodutor em  $\mathcal{H}$ .*

$E$ : espaço “físico”;  $\mathcal{H}$ : RKHS.

# Exemplos

## Teorema (Moore-Aronszajn)

*$K$  é uma matriz de Moore em  $E$  se e só se existe um único espaço de Hilbert  $\mathcal{H}$  de funções de  $E$  tal que  $K$  é um núcleo reprodutor em  $\mathcal{H}$ .*

$E$ : espaço “físico”;  $\mathcal{H}$ : RKHS.

$E$	$\mathcal{H}$	$K$
$\{1, 2, \dots, n\}$	$\mathbb{C}^n$	$M$ matriz SDP
$I \subset \mathbb{R}$	$L^2(\mathbb{R})$	núcleo integral de Mercer
$\Omega \subset \mathbb{C}^n$	$A^2(\Omega)$	Bergman, Szegő, Hardy...

# (Principais) Áreas de aplicação

- 1 EDPs (Bergman e Schiffer; eqs elípticas em geral)
- 2 Aplicações conformes (Bergman, Nehari)
- 3 Processos gaussianos
- 4 Análise harmónica em semigrupos (FDP)
- 5 Noção de hipergrupo e análise harmónica em hipergrupos
- 6 Teoria de Operadores (de Branges, Rovnyak, Arov)
- 7 Problemas inversos (Arov, Dym)
- 8 Polinómios ortogonais (fórmulas de Christoffel-Darboux)
- 9 Problemas de contagem de zeros
- 10 Sistemas dissipativos lineares

## (Principais) Áreas de aplicação II

- 11 O algoritmo de Schur (Dym, Alpay)
- 12 Interpolação (Nevanlinna-Pick, Carathéodory-Fejér)
- 13 Desigualdades (Ando, Saitoh)
- 14 Transformações integrais (Saitoh)
- 15 Funções de SCV (Arazy, Hua)
- 16 Bases de exponenciais em  $L^2(I)$  e teoria de controlo (Nikolski).
- 17 Aproximação em RKHS e funções de spline
- 18 Reconhecimento de padrões, aprendizagem automática, SVM (support vector machines) (V. Vapnik, G. Wahba)

# Machine Learning: o “kernel trick”

## Proposição (Krein, Saitoh...)

*Um RK  $K(x, y)$  admite uma representação no espaço de Hilbert  $\mathcal{H}$*

$$K(x, y) = \langle \varphi_x, \varphi_y \rangle_{\mathcal{H}}. \quad (0.3)$$

Esta representação aplica-se a qualquer algoritmo que depende apenas do p.i. entre dois vectores. Quando surge um p.i. entre vectores, utiliza-se o valor correspondente do núcleo  $K(x_i, x_j)$ .

# Machine Learning: o “kernel trick”

## Proposição (Krein, Saitoh...)

*Um RK  $K(x, y)$  admite uma representação no espaço de Hilbert  $\mathcal{H}$*

$$K(x, y) = \langle \varphi_x, \varphi_y \rangle_{\mathcal{H}}. \quad (0.3)$$

Esta representação aplica-se a qualquer algoritmo que depende apenas do p.i. entre dois vectores. Quando surge um p.i. entre vectores, utiliza-se o valor correspondente do núcleo  $K(x_i, x_j)$ . Transformamos assim um algoritmo linear num algoritmo não-linear!

# Machine Learning: o “kernel trick”

## Proposição (Krein, Saitoh...)

Um RK  $K(x, y)$  admite uma representação no espaço de Hilbert  $\mathcal{H}$

$$K(x, y) = \langle \varphi_x, \varphi_y \rangle_{\mathcal{H}}. \quad (0.3)$$

Esta representação aplica-se a qualquer algoritmo que depende apenas do p.i. entre dois vectores. Quando surge um p.i. entre vectores, utiliza-se o valor correspondente do núcleo  $K(x_i, x_j)$ . Transformamos assim um algoritmo linear num algoritmo não-linear!

A aplicação  $\varphi$  chama-se *feature map*. A parte notável de (0.3) é que permite proceder sem conhecer *a priori* o feature map ou sequer o *feature space*.

# RKs e *Machine Learning* II

Núcleos  
Reprodutores  
em  
Matemática e  
Engenharia

Jorge Buescu  
FCUL  
CMAF

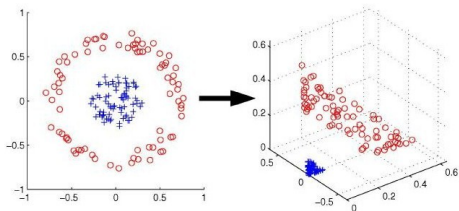


Figure 1: Transforming the data can make it linearly separable

# RKs e *Machine Learning* II

Núcleos  
Reprodutores  
em  
Matemática e  
Engenharia

Jorge Buescu  
FCUL  
CMAF

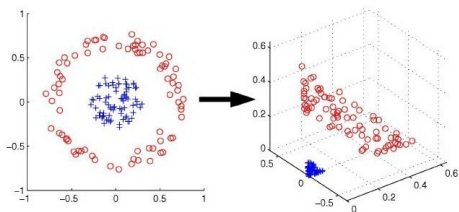


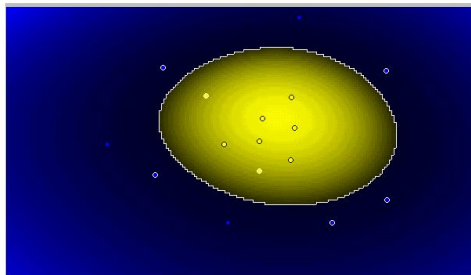
Figure 1: Transforming the data can make it linearly separable

O kernel trick é utilizado em inúmeras áreas da aprendizagem automática (algorítmica):

- 1 SVM (Support Vector Machines)
- 2 Análise de componentes principais
- 3 Análise de correlações
- 4 Clustering...

# Reconhecimento de padrões via SVM

Reconhecimento de padrões (voz, imagem, etc).



e.g. following applet

RH University of London

<http://svm.dcs.rhnc.ac.uk/pagesnew/GPat.shtml>

# RK na indústria: a YDreams

Núcleos  
Reprodutores  
em  
Matemática e  
Engenharia

Jorge Buescu  
FCUL  
CMAF



## Vision of the future.

YVision is a groundbreaking framework to easily build engaging interactive applications. It integrates image processing, 3D rendering, physics, augmented reality, artificial intelligence and others. Due to YVision's innovative core architecture, developers can use control graphs for easy threading and multicore applications. Integrated technology and easy of use will allow you to develop better interactive products with more features, speed and quality.

### Augmented Reality

Create incredible Augmented Reality applications and take your digital characters closer to life in your environment.

### Immersive Environments

Use interactive graphics to create immersive environments and unforgettable experiences.

### Gesture-based Interaction

Our innovative computer vision algorithms can be used to create fully interactive with digital computer graphics.

### INTERACTIVE APPLICATION

SCRIPTING  
DATA DRIVEN ENGINE  
MULTIMEDIA  
GRAPHICS  
PHYSICS  
AI  
3D RENDERING

EMULATION  
COMPUTER VISION  
PRESENTER ABSTRACTS  
PRESENTER INTERACTION

**CORE FRAMEWORK**  
A modular and extensible framework for building interactive applications.

**YDREAMS**

[www.ydreams.com](http://www.ydreams.com)

At YDreams we create interactive experiences based on advanced technology, creativity and design. For the past nine years we have been using computer graphics and complex mathematics to create innovative applications and environments. From virtual education to state-of-the-art museum to corporate buildings, we have worked with corporations and institutions from all over the world.

Headquarters in Portugal and with offices in Spain, Brazil, USA. YDreams continues and develops its open to future technologies, working towards the expansion and evolution of computer interactive technology.

Y DREAMS S.L. - P.º 100000479 (PT)  
Rua de Vila Verde, 140  
4470-125 Vila Verde - Portugal | Tel: +351 251 081 000  
Web: [www.ydreams.com](http://www.ydreams.com)

Y DREAMS S.L.  
Rua de Vila Verde, 140  
4470-125 Vila Verde - Portugal  
Tel: +351 251 081 000 | Fax: +351 251 780 788

## YVision

### YVision mathematics

- 3D Rendering: Linear Algebra - Differential Geometry - Geometry (Projective Transformations)
- Physics Simulation: Linear Algebra - Differential Calculus - Integral Calculus - Numerical Analysis (for approximating continuous mathematics)
- Computer Vision: Linear Algebra - Geometry (Homographies, Projective Transformations) - Cellular Automata - Convolutions - Fourier Analysis
- Synthesis and Transform Audio: Fourier Analysis, Synthesis and Transform
- Machine Learning: Statistical Analysis - Artificial Neural Networks - Principal Component Analysis - Regression - Function Approximation
- Evolutionary Computation Core Framework: Lambda Calculus (root of functional programming) - Turing Machines - Theory of Computation