Support Fuzzy-Set Machines

From Kernels on Fuzzy Sets to Machine Learning Applications
Tutorial WCCI IEEE-FUZZ 2018

Jorge Guevara Díaz[1] Roberto Hirata Jr.[2] Stéphane Canu[3]

[1]IBM Research

[2]Institute of Mathematics and Statistics
University of Sao Paulo-Brazil

[3]Institut National de Sciences Appliquees
University of Normandy-France

July 08, 2018

Outline

Outline

1. Introduction
2. Kernel Machines
3. Kernels on Fuzzy sets
4. Support fuzzy-set machines

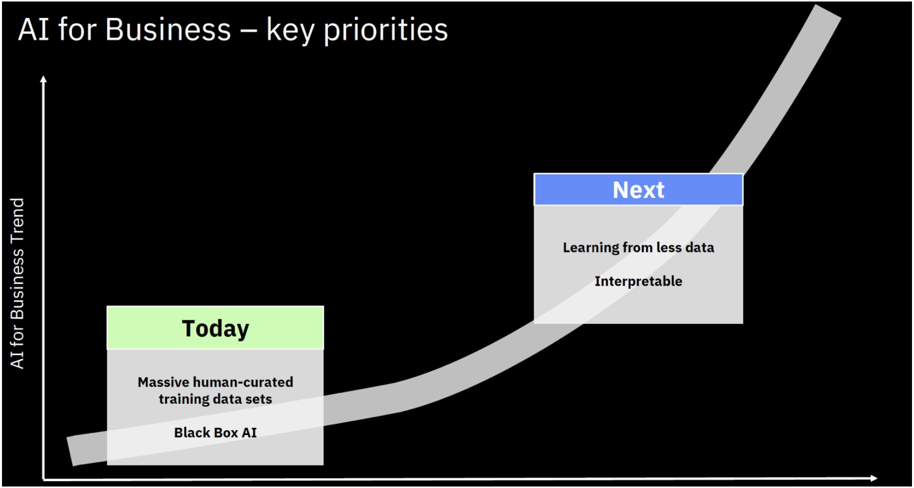
Introduction

Signal Comprehension:
From video and text to rich human perception

Learning and Reasoning:
From scalable machine learning to making a case

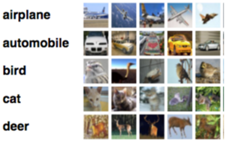
Interaction:
Understanding language, tone, emotion and context

Introduction

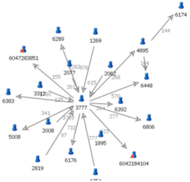


Introduction

Data

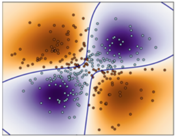


Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa

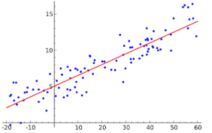


Tasks

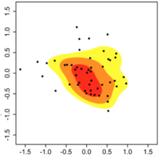
Classification



Regression



Density estimation



Introduction

Data

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa

Learn

y= $f(x)$

Tasks

Classification

Regression

Density estimation

Introduction

- Data representation
  - Images
    - Vectors
    - Conditional random fields
  - Structured data
    - logical predicates
    - Graphs
    - Distributions
  - Uncertain data
    - Probability measures
    - Fuzzy sets
  - Granular data
    - Fuzzy sets
    - Granular representations
- Similarity measures
  - A real valued function that quantifies the similarity between two objects
  - Many of them:
    - Inner products (Kernels)
    - Cosine similarity
    - Fuzzy similarity measures
    - etc
- Property
  - Inverse of distance metrics (in some sense)

What is Machine Learning?

- A field that uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed

  Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development

What is Machine Learning?

- A field that uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed

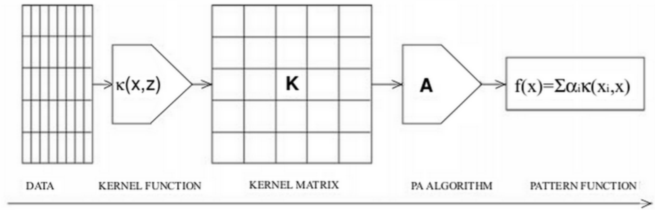
  Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development

- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E

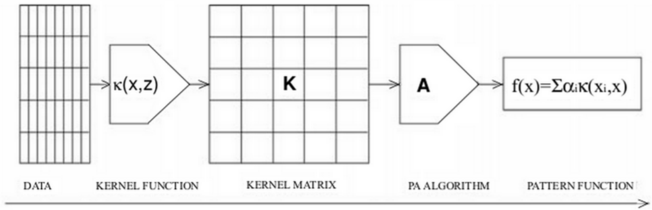
  Mitchell, T. (1997). Machine Learning. McGraw Hill

Outline

Kernel methods



Kernel methods



Support Vector Machines

$$
\begin{aligned}
\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \frac{1}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \\
\text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0. \\
& 0 \le \alpha_i \le \lambda, \; i = 1, \ldots, N.
\end{aligned}
$$

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297
Figure from Shawe-Taylor, John, and Nello Cristianini. Kernel methods for pattern analysis. Cambridge university press, 2004.

Support Vector Machines

$$
\begin{aligned}
\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \\
\text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0. \\
& 0 \leq \alpha_i \leq \lambda, \ \ i = 1, \dots, N.
\end{aligned}
$$

Implementation example CVX (http://cvxr.com/cvx/) and MATLAB

0.4499169	0.0015187	1.0000000
0.8180969	0.3841642	1.0000000
0.6864155	0.9862297	1.0000000
0.8403105	0.0462144	1.0000000
0.2644775	0.0609907	1.0000000
0.1371175	0.3148678	1.0000000
1.1708515	1.3779732	-1.0000000
1.0769097	1.5763168	-1.0000000
0.8492538	1.6056590	-1.0000000
1.3273406	1.5865217	-1.0000000
1.1415545	1.7362638	-1.0000000

Support Vector Machines

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0. \\ & 0 \leq \alpha_i \leq \lambda, \; i = 1, \dots, N. \end{aligned}$$

Implementation example monqp (http://asi.insa-rouen.fr/enseignants/ arakoto/toolbox/index.html) and MATLAB

```
%-----------------------------dual matlab CVX
K=computeKernel(X,y,kernelParameters)
%K=(y*y').*(X*X');
K=K+eps*eye(n);

C=10;
cvx_begin
    variable alpha(n);
        dual variables bDual alDual allDual;
            minimize( 0.5*alpha'*K*alpha-ones(n,1)'*alpha)
            subject to
                bDual  : alpha'*y==0;
                alDual : zeros(n,1)<=alpha<=C*ones(n,1);
cvx_end
%-----------------------------dual solution
supportVectorIndex=find(alpha>0.00001);
```

Support Vector Machines

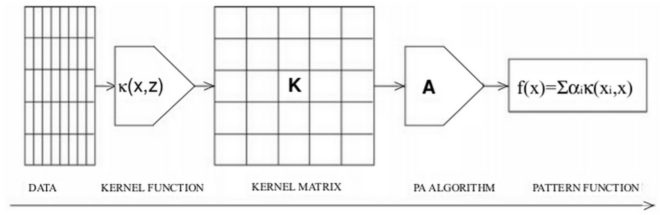
$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0. \\ & 0 \le \alpha_i \le \lambda, \ i = 1, \dots, N. \end{aligned}$$

Implementation example CVX and MATLAB

```
K=computeKernel(X,y,kernelParameters)
%K=(y*y').*(X*X');
n=length(K);
K=K+eps*eye(n);

l = 10^-11;
verbose = 0;
e=ones(n,1);
C=10

[alpha, b, supportVectorIndex] = monqp(K,e,y,0,C,l,verbose);
```

```
%predictions
%------------
G=computeKernel(X(supportVectorIndex,:),XTest,kernelParam)
%G=X(supportVectorIndex,:)*XTest'
ypred = G'*(y(supportVectorIndex).*alpha) + b
```

Kernel methods



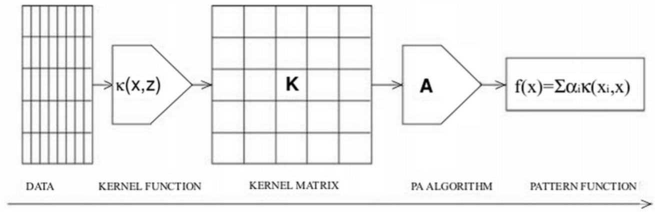
Kernel PCA

$$\begin{array}{ll} \text{Solve} & M\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \\ \text{subject to} & \boldsymbol{\alpha}^\top K\boldsymbol{\alpha} = 1. \end{array}$$

where the eigen functions are given by $V(.) = \sum_i \alpha_i k(x_i, .)$

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Nonlinear component analysis as a kernel eigenvalue problem." Neural computation 10.5 (1998): 1299-1319.

Kernel methods

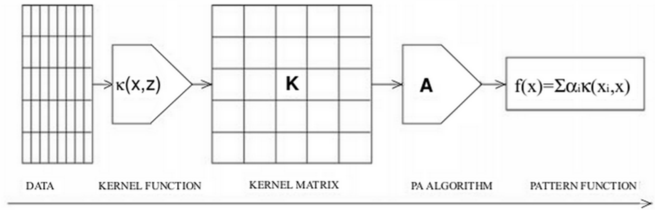


Gaussian process regression

- $f \sim \mathcal{GP}\left(\mathbb{E}[f(X)], k(x,x)\right) = \mathcal{GP}\left(m(x), k(x,x)\right)$
- Predictive distribution $y_i \mid x_*, x, \mathbf{y} \sim \mathcal{N}\left(k(x_*, x)k(x,x)^{-\mathbf{1}}\mathbf{y}, k(x_*, x_*) - k(x_*, x)k(x,x)^{-\mathbf{1}}k(x, x_*)\right)$
- thus, $y = \hat{f}(x_*) \equiv k(x_*, x)k(x,x)^{-\mathbf{1}}\mathbf{y} = \sum_i \alpha_i k(x_i, x_*)$, , where $\boldsymbol{\alpha} = K^{-\mathbf{1}}\mathbf{y}$

Rasmussen, Carl Edward. "Gaussian processes in machine learning." Advanced lectures on machine learning. Springer, Berlin, Heidelberg, 2004. 63-71.

Kernel methods



Support Vector Data Description

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \quad & \boldsymbol{\alpha}^\top K\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top diag(K) \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{1} = 0. \\ & 0 \leq \alpha_i \leq \lambda,\ i = 1, \ldots, N, \end{aligned}$$

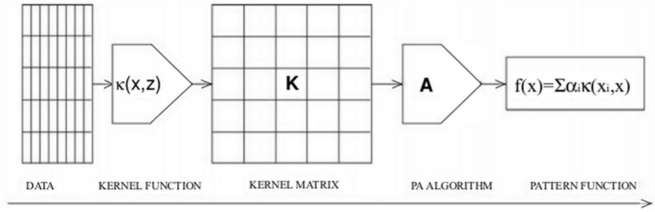
Tax, David MJ, and Robert PW Duin. "Support vector data description." Machine learning 54.1 (2004): 45-66.

Support Vector Data Description

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \quad & \boldsymbol{\alpha}^\top K\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top diag(K) \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{1} = 0. \\ & 0 \leq \alpha_i \leq \lambda,\ \ i = 1, \ldots, N, \end{aligned}$$

```
cvx_begin
    variables  alpha(n);
        dual variables lambda
            minimize (alpha'*K*alpha-alpha'*diag(K));
            subject to
                zeros(n,1)<=alpha<=C*ones(n,1)
                lambda:alpha'*ones(n,1) ==1;
cvx_end
toc

Io=find(alpha>0.00000001);  %supportVectorIndex

cNorm=alpha(Io)'*K(Io,Io)*alpha(Io);

if kernel=='linear'
    c=(alpha(Io))'*K(Io,:);
else
    c=-1; % non linear kernel
end

R=sqrt(-lambda+cNorm);
```

Kernel methods



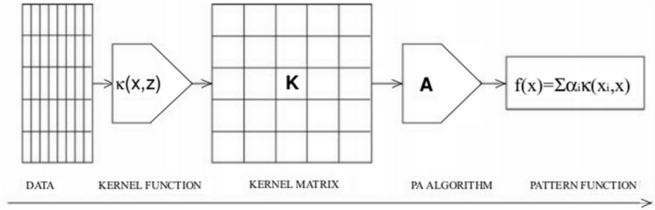
Kernel two-sample test

$$MMD[\mathcal{F}, \mathbb{P}, \mathbb{Q}] = \|\mathbb{E}_{X \sim \mathbb{P}}[k(.,X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[k(.,Y)]\|_{\mathcal{H}}, \quad (1)$$

$$MMD_u^2[\mathcal{F}, s_X, s_Y] = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j)$$

Gretton, Arthur, et al. "A kernel two-sample test." Journal of Machine Learning Research 13.Mar (2012): 723-773.

Kernel methods

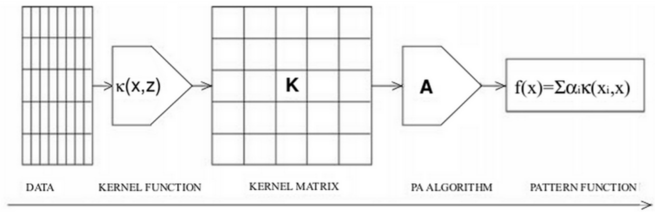


Radial basis function neural network

$$f(x) = \sum_{l=1}^{L} w_l \exp(-\gamma \|x - \mu_k\|^2)$$

Broomhead, David S., and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. No. RSRE-MEMO-4148. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

Kernel methods



- SVM
- kernel PCA
- Gaussian process
- SVDD
- MKL
- SMDD
- Kernel regression
- Kernel two-sample test
- kernel spectral clustering

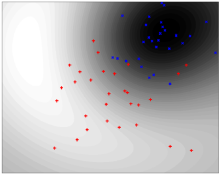
Representer Theorem

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \, Cost\left((x_1, y_1, f(x_1)), \ldots, (x_N, y_N, f(x_N))\right) + \Omega(\|f\|) \quad (2)$$

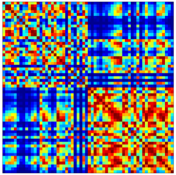
$$f^*(.) = \sum_{i=1}^{N} \alpha_i k(., x_i) \quad (3)$$

Kernel methods

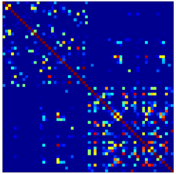
A kernel matrix is a similarity matrix



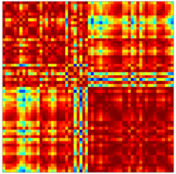
raw data



Gram matrix for $b = 2$



$b = .5$



$b = 10$

RKHS, where the magic happens

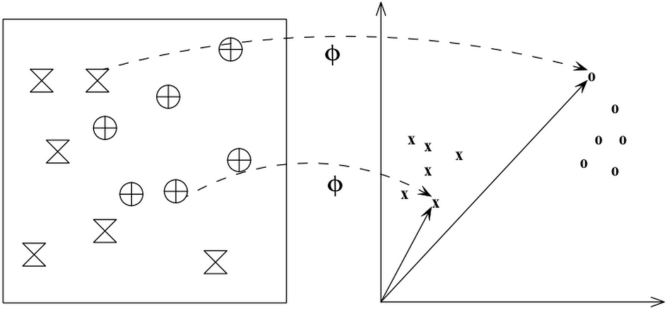


Figure: Kernel mapping[a]

Figure from Shawe-Taylor et al. "Kernel Methods for Pattern Analysis". Cambridge University Press.

Main ingredient

- A real-valued symmetric positive definite kernel k.
  $\sum_{i=1,j=1}^{N} c_i c_j k(x_i, x_j) \geq 0$

$$\boxed{\begin{array}{l}\mathcal{X} \to \mathcal{H} \\ x \mapsto k(.,x)\end{array}}$$

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

$$\begin{array}{ll}\forall x \in \mathcal{X},\ k(.,x) \in \mathcal{H} & \\ \forall x \in \mathcal{X},\ \forall f \in \mathcal{H} & \langle f, k(.,x)\rangle = f(x) \\ \forall x, x' \in \mathcal{X} & k(x,x') = \langle k(.,x), k(.,x')\rangle\end{array}$$

Reproducing Kernel Hilbert Spaces

Definition (Reproducing kernel)

A function

$$\begin{aligned} k : \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \\ (x, y) &\mapsto k(x, t) \end{aligned} \tag{4}$$

is called a reproducing kernel of the Hilbert space $\mathcal{H}$ *if and only if:*

1. $\forall x \in \mathcal{X},\ k(., x) \in \mathcal{H}$
2. $\forall x \in \mathcal{X},\ \forall f \in \mathcal{H}\ \ \langle f, k(., x)\rangle_{\mathcal{H}} = f(x)$

Reproducing Kernel Hilbert Spaces

Definition (Reproducing kernel)

A function

$$\begin{aligned} k : \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \\ (x, y) &\mapsto k(x, t) \end{aligned} \quad (4)$$

is called a reproducing kernel of the Hilbert space $\mathcal{H}$ *if and only if:*

1. $\forall x \in \mathcal{X},\ k(., x) \in \mathcal{H}$
2. $\forall x \in \mathcal{X},\ \forall f \in \mathcal{H}\ \ \langle f, k(., x)\rangle_{\mathcal{H}} = f(x)$

Reproducing property

$$\forall (x, y) \in \mathcal{X} \times \mathcal{X},\ k(x, y) = \langle k(., x), k(., y)\rangle_{\mathcal{H}} \quad (5)$$

Reproducing Kernel Hilbert Spaces

Definition (Real RKHS)

A Hilbert Space of real valued functions on $\mathcal{X}$, denoted by $\mathcal{H}$, with reproducing kernel is called a real Reproducing Kernel Hilbert Space or real RKHS.

Reproducing Kernel Hilbert Spaces

Definition (Real RKHS)

A Hilbert Space of real valued functions on $\mathcal{X}$, denoted by $\mathcal{H}$, with reproducing kernel is called a real Reproducing Kernel Hilbert Space or real RKHS.

Characterization

All the evaluation functionals are continuous on $\mathcal{H}$. :

$$
\begin{aligned}
e_x : \mathcal{H} &\rightarrow \mathbb{R} &(6)\\
f &\mapsto e_x(f) = f(x) &(7)
\end{aligned}
$$

Berlinet, Alain, and Christine Thomas-Agnan. Reproducing kernel Hilbert spaces in probability and statistics. Springer Science Business Media, 2011.

Positive Definite Kernel

Lemma

Any reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric positive definite function, that is, it satisfies:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(x_i, x_j) \geq 0 \tag{8}$$

$\forall N \in \mathbb{N}$, $\forall c_i, c_j \in \mathbb{R}$ *and* $k(x, y) = k(y, x)$, $\forall x, y \in \mathcal{X}$. *The converse is true.*

Positive Definite Kernel

Lemma

Any reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric positive definite function, that is, it satisfies:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(x_i, x_j) \geq 0 \tag{8}$$

$\forall N \in \mathbb{N}$, $\forall c_i, c_j \in \mathbb{R}$ *and* $k(x, y) = k(y, x)$, $\forall x, y \in \mathcal{X}$. *The converse is true.*

Consequently

Kernels k are reproducing kernels of some RKHS. The space spanned by $k(x, .)$ generates a RKHS or a Hilbert space with reproducing kernel k.

Positive Definite Kernel

If k is a reproducing kernel, then

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j k(x_i, x_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \langle k(., x_i), k(., x_j)\rangle_{\mathcal{H}} \\
&= \langle \sum_{i=1}^{N} c_i k(., x_i), \sum_{j=1}^{N} c_j k(., x_j)\rangle_{\mathcal{H}} \\
&= \| \sum_{i=1}^{N} c_i k(., x_i)\|_{\mathcal{H}}^2 \\
&\geq 0
\end{aligned}
$$

Positive Definite Kernel

If k is a reproducing kernel, then

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j k(x_i, x_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \langle k(., x_i), k(., x_j)\rangle_{\mathcal{H}} \\
&= \langle \sum_{i=1}^{N} c_i k(., x_i), \sum_{j=1}^{N} c_j k(., x_j)\rangle_{\mathcal{H}} \\
&= \| \sum_{i=1}^{N} c_i k(., x_i)\|_{\mathcal{H}}^2 \\
&\geq 0
\end{aligned}$$

That is

Elements of the RKHS are real-valued functions on $\mathcal{X}$ of the form $f(.) = \sum_{i=1}^{N} c_i k(., x_i)$.

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x}^\top \mathbf{x}')^2 \\
&= (1 + x_1 x_1' + x_2 x_2')^2 \\
&= (1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_1' x_2 x_2') \\
&= \left\langle (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2\ ,\ \sqrt{2}x_1 x_2), (1, x_1'^2, x_2'^2, \sqrt{2}x_1', \sqrt{2}x_2', \sqrt{2}x_1' x_2') \right\rangle
\end{aligned}
$$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x}^\top \mathbf{x}')^2 \\
&= (1 + x_1 x_1' + x_2 x_2')^2 \\
&= (1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_1' x_2 x_2') \\
&= \left\langle (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2 \ , \ \sqrt{2}x_1 x_2), (1, x_1'^2, x_2'^2, \sqrt{2}x_1', \sqrt{2}x_2', \sqrt{2}x_1' x_2') \right\rangle
\end{aligned}
$$

Polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (a + b\mathbf{x}^\top \mathbf{x}')^\gamma$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x}^\top \mathbf{x}')^2 \\
&= (1 + x_1 x_1' + x_2 x_2')^2 \\
&= (1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_1' x_2 x_2') \\
&= \left\langle (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2\ ,\ \sqrt{2}x_1 x_2), (1, x_1'^2, x_2'^2, \sqrt{2}x_1', \sqrt{2}x_2', \sqrt{2}x_1' x_2') \right\rangle
\end{aligned}
$$

Polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (a + b\mathbf{x}^\top \mathbf{x}')^\gamma$

```
K=((X*Z'+kernelParam(1)).^kernelParam(2));
```

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$
\begin{aligned}
k(x, x') &= \exp\big(-(x - x')^2 \big) \\
&= \exp(-x^2) \exp(-x'^2) \sum_{i=0}^{\infty} \frac{2^i x^i x'^i}{i!}
\end{aligned}
$$

infinite dimensional

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$\begin{aligned} k(x,x') &= \exp\big(-(x-x')^2\big) \\ &= \exp(-x^2)\exp(-x'^2)\sum_{i=0}^{\infty}\frac{2^i x^i x'^i}{i!} \end{aligned}$$

infinite dimensional
RBF kernel $k(\mathbf{x},\mathbf{x}') = \exp(-0.5 * \gamma\|\mathbf{x}-\mathbf{x}'\|^2)$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

$$
\begin{aligned}
k(x, x') &= \exp\left(-(x - x')^2 \right) \\
&= \exp(-x^2)\exp(-x'^2)\sum_{i=0}^{\infty} \frac{2^i x^i x'^i}{i!}
\end{aligned}
$$

infinite dimensional
RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-0.5 * \gamma \|\mathbf{x} - \mathbf{x}'\|^2)$

```
K=exp(-0.5*kernelParam*sqdistAll(X,Z));
```

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

The real-valued kernel on $\mathcal{P} \times \mathcal{P}$, defined by

$$\begin{aligned}\tilde{k}(\mathbb{P}, \mathbb{Q}) =& \langle \mathbb{E}_{\mathbb{P}}[k(X.,)], \mathbb{E}_{\mathbb{Q}}[k(X'.,)] \rangle_{\mathcal{H}} \\ =& \int_{\mathbf{x} \in \mathbb{R}^D} \int_{\mathbf{x}' \in \mathbb{R}^D} k(\mathbf{x}, \mathbf{x}') d\mathbb{P}(\mathbf{x}) d\mathbb{Q}(\mathbf{x}')\end{aligned} \tag{9}$$

is positive definite.

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels
The real-valued kernel on $\mathcal{P} \times \mathcal{P}$, defined by

$$\begin{aligned}\tilde{k}(\mathbb{P},\mathbb{Q}) =& \langle \mathbb{E}_{\mathbb{P}}[k(X.,)], \mathbb{E}_{\mathbb{Q}}[k(X'.,)]\rangle_{\mathcal{H}} \\ =& \int_{\mathbf{x}\in\mathbb{R}^D}\int_{\mathbf{x}'\in\mathbb{R}^D} k(\mathbf{x},\mathbf{x}')d\mathbb{P}(\mathbf{x})d\mathbb{Q}(\mathbf{x}')\end{aligned} \tag{9}$$

is positive definite.

```
for i=1:n
    for j=i:n
        [L,~] = size(S{i});
        [LL,~]= size(S{j});

        k=kernel(S{i},S{j},kernelOp,kernelParam);

        K(i,j)= sum(sum(k))/(L*LL);
        K(j,i)=Krr(i,j);

    end
end
```

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Mean maps with stationary kernels do not have constant norm
  $\|\mu_{\mathbb{P}}\|_{\mathcal{H}} = \|\mathbb{E}_{\mathbb{P}}[k_l(X.,)]\|_{\mathcal{H}} \leq \mathbb{E}_{\mathbb{P}}[\|k_l(X.,)\|_{\mathcal{H}}] = \sqrt{|\epsilon|}$
- normalize mean maps to lie on a surface of some hypersphere

$$\tilde{\tilde{k}}(\mathbb{P}_i, \mathbb{P}_j) = \frac{\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}}{\sqrt{\langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}}}, \tag{10}$$

- the injectivity of $\mu : \mathcal{P} \to \mathcal{H}$ is preserved.

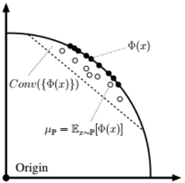


Figure: Figure from [a]

[a] Muandet et al "One-class support measure machines for group anomaly detection."

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Mean maps with stationary kernels do not have constant norm
  $\|\mu_{\mathbb{P}}\|_{\mathcal{H}} = \|\mathbb{E}_{\mathbb{P}}[k_l(X.,)]\|_{\mathcal{H}} \leq \mathbb{E}_{\mathbb{P}}[\|k_l(X.,)\|_{\mathcal{H}}] = \sqrt{|\epsilon|}$
- normalize mean maps to lie on a surface of some hypersphere

$$\tilde{\tilde{k}}(\mathbb{P}_i, \mathbb{P}_j) = \frac{\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}}{\sqrt{\langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}}}, \tag{11}$$

- the injectivity of $\mu : \mathcal{P} \to \mathcal{H}$ is preserved.

```
for i=1:n
    for j=i:n
        [L,~] = size(S{i});
        [LL,~]= size(S{j});

        k=kernel(S{i},S{j},kernelOp,kernelParam);

        K(i,j)= sum(sum(k))/(L*LL);
        K(j,i)=K(i,j);

        KN=K./sqrt(diag(K)*diag(K)');
    end
end
```

Positive Definite Kernel

Do you want implement this kernel?

$$k_{\nu=p+1/2}(r) = \exp\Big(-\frac{\sqrt{2\nu}r}{\ell}\Big)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\Big(\frac{\sqrt{8\nu}r}{\ell}\Big)^{p-i}.$$

Positive Definite Kernel

Do you want implement this kernel?

$$k_{\nu=p+1/2}(r) = \exp\left(-\frac{\sqrt{2\nu}r}{\ell}\right)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}r}{\ell}\right)^{p-i}.$$

That is the Matern kernel, widely used in kriging procedures by geostatisticians

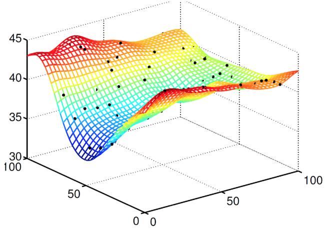


Figure from: https://www.ethz.ch/content/specialinterest/baug/institute-ibk/risk-safety-and-uncertainty/en/research/past-projects/polynomial-chaos-kriging.html

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Linear kernel $k(x, y) = \langle x, y \rangle \; x, y \in \mathbb{R}^D$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Linear kernel $k(x, y) = \langle x, y \rangle \; x, y \in \mathbb{R}^D$
- Polynomial kernel $k(x, y) = (\langle x, y \rangle + 1)^D \; x, y \in \mathbb{R}^D$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Linear kernel $k(x, y) = \langle x, y \rangle \; x, y \in \mathbb{R}^D$
- Polynomial kernel $k(x, y) = (\langle x, y \rangle + 1)^D \; x, y \in \mathbb{R}^D$
- Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/\sigma^2) \; x, y \in \mathbb{R}^D$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Linear kernel $k(x, y) = \langle x, y \rangle \; x, y \in \mathbb{R}^D$
- Polynomial kernel $k(x, y) = (\langle x, y \rangle + 1)^D \; x, y \in \mathbb{R}^D$
- Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/\sigma^2) \; x, y \in \mathbb{R}^D$
- Probability product kernel $\tilde{k}(\mathbb{P}, \mathbb{Q}) = \int_{\mathcal{X}} \mathbb{P}(x)^\rho \mathbb{Q}(x)^\rho dx$

Positive Definite Kernel

Examples of reproducing kernels or positive definite kernels

- Linear kernel $k(x, y) = \langle x, y \rangle \; x, y \in \mathbb{R}^D$
- Polynomial kernel $k(x, y) = (\langle x, y \rangle + 1)^D \; x, y \in \mathbb{R}^D$
- Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/\sigma^2) \; x, y \in \mathbb{R}^D$
- Probability product kernel $\tilde{k}(\mathbb{P}, \mathbb{Q}) = \int_{\mathcal{X}} \mathbb{P}(x)^\rho \mathbb{Q}(x)^\rho dx$
- Kernel on probability measures for $X \sim \mathbb{P}, X' \sim \mathbb{Q}$,
  $\tilde{k}(\mathbb{P}, \mathbb{Q}) = \langle \mathbb{E}_{\mathbb{P}}[k(X.,)], \mathbb{E}_{\mathbb{Q}}[k(X'.,)] \rangle_{\mathcal{H}}$

Positive Definite Kernel

Making new kernels from old If k_1 and k_2 are PD kernels, by closure properties of PD kernels, also are PD kernels:

1. $k_1(x, y) + k_2(x, y)$;
2. $\alpha k_1(x, y), \quad \alpha \in \mathbb{R}^+$;
3. $k_1(x, y)k_2(x, y)$;
4. $k(f(x), f(y))$
5. $\exp(k_1(x, y))$;
6. $p(k_1(x, y)), \quad p$ is a polynomial with positive coefficients.

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,
- similarity measures,

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,
- similarity measures,
- related to metrics by: $k(x, x) - 2k(x, x') + k(x', x')$

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,
- similarity measures,
- related to metrics by: $k(x, x) - 2k(x, x') + k(x', x')$
- the main ingredient to define RKHS

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,
- similarity measures,
- related to metrics by: $k(x, x) - 2k(x, x') + k(x', x')$
- the main ingredient to define RKHS
- very useful in practice in machine learning, i.e., they define gram matrices $K_{i,j} = k(x_i, x_j)$ used in ML algorithms

Kernels

Positive definite kernels are

- coovariance functions, i.e., $k(x, x') = \mathbb{E}\big[f(x)f(x')\big]$,
- similarity measures,
- related to metrics by: $k(x, x) - 2k(x, x') + k(x', x')$
- the main ingredient to define RKHS
- very useful in practice in machine learning, i.e., they define gram matrices $K_{i,j} = k(x_i, x_j)$ used in ML algorithms
- defined on non empty sets: That enables its use on non-vectorial spaces. (sets, graphs, strings, etc)

Outline

Kernels on Fuzzy sets

Material

- Guevara, Jorge, et.al. **"Kernels on Fuzzy Sets: an Overview"**, Learning on Distributions, Functions, Graphs and Groups @ NIPS-2017
- Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.
- Guevara, Jorge. **"Supervised machine learning with kernel embeddings of fuzzy sets and probability measures."** Diss. IME USP, 2016.
- Guevara, Jorge, et.al. **"Fuzzy Set Similarity using a Distance-Based Kernel on Fuzzy Sets"**, 2016, Book Chapter, Handbook of Fuzzy Sets Comparison - Theory, Algorithms and Applications,pages 103-120.
- Guevara, Jorge, et.al. **"Positive Definite Kernel Functions on Fuzzy Sets."** Fuzzy Systems (FUZZ-IEEE), 2014.
- Guevara, Jorge, et.al.**"Kernel Functions in Takagi-Sugeno-Kang Fuzzy System with Nonsingleton Fuzzy Input."** Fuzzy Systems (FUZZ-IEEE), 2013.

Kernels on Fuzzy sets

Fuzzy data

Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	MF
2213	345	23	2	34543	34545	0.9
3234	345	13	4	34556	34534	0.7
4423	355	77	5	45366	34545	0.8
2343	367	27	37	54535	34563	0.002

Kernels on Fuzzy sets

Fuzzy data

Var 1	Var 2	Var 3	Var 4
2213	345	23	2
3234	345	13	4
4423	355	77	5
2343	367	27	37

MF 1	MF 2	MF 3	MF 4
0.3	0.9	1	0.9
0.5	0.9	0.3	0.4
1	0.5	0.002	0.7
0.4	2	0.9	0.01

Kernels on Fuzzy sets

Fuzzy data

Var 1	Var 2	Var 3	Var 4	Var 5
X_1^1	X_2^1	X_3^1	X_4^1	X_5^1
X_1^2	X_2^2	X_3^2	X_4^2	X_5^2
X_1^3	X_2^3	X_3^3	X_4^3	X_5^3
X_1^4	X_2^4	X_3^4	X_4^4	X_5^4

$$\begin{aligned} X : \Omega &\rightarrow [0,1] \\ x &\mapsto X(x). \end{aligned}$$

Kernels on Fuzzy sets

Motivation

- similarity measure between fuzzy sets given by kernels

Kernels on Fuzzy sets

Motivation

- similarity measure between fuzzy sets given by kernels
- geometric interpretation of similarity between fuzzy set in RKHS

Kernels on Fuzzy sets

Motivation

- similarity measure between fuzzy sets given by kernels
- geometric interpretation of similarity between fuzzy set in RKHS
- embedding of fuzzy sets into RKHS

Kernels on Fuzzy sets

Motivation

- similarity measure between fuzzy sets given by kernels
- geometric interpretation of similarity between fuzzy set in RKHS
- embedding of fuzzy sets into RKHS
- using fuzzy data "as is" in kernel methods

Kernels on Fuzzy sets

Motivation

- similarity measure between fuzzy sets given by kernels
- geometric interpretation of similarity between fuzzy set in RKHS
- embedding of fuzzy sets into RKHS
- using fuzzy data "as is" in kernel methods
- covariance matrix for fuzzy samples

Fuzzy Set

- Fuzzy sets on Ω are denoted by X, Y, Z.

Fuzzy Set

- Fuzzy sets on Ω are denoted by X, Y, Z.
- The membership function $\Omega :\to [0, 1]$ of a fuzzy set is denoted by $X(.)$.

Fuzzy Set

- Fuzzy sets on Ω are denoted by X, Y, Z.
- The membership function $\Omega :\rightarrow [0, 1]$ of a fuzzy set is denoted by $X(.)$.
- $\mathcal{F}(\Omega)$ is the set of all the fuzzy sets on Ω.

Fuzzy Set

- Fuzzy sets on Ω are denoted by X, Y, Z.
- The membership function $\Omega :\to [0,1]$ of a fuzzy set is denoted by $X(.)$.
- $\mathcal{F}(\Omega)$ is the set of all the fuzzy sets on Ω.

Definition (Support of a fuzzy set)

The support of a fuzzy set is the set

$$\text{supp}(X) = \{x \in \Omega | X(x) > 0\}.$$

Cross product kernel on fuzzy sets

Definition (Cross product kernel on fuzzy sets)

The cross product kernel on fuzzy sets is a function $k_\times : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \to \mathbb{R}$ *given by:*

$$k_\times(X, Y) = \sum_{\substack{x \in \text{supp}(X),\\ y \in \text{supp}(Y)}} k_1 \otimes k_2\big((x, X(x)), (y, Y(y))\big), \tag{12}$$

where: $k_1 : \Omega \times \Omega \to \mathbb{R}$, $k_2 : [0, 1] \times [0, 1] \to \mathbb{R}$,

$$k_1 \otimes k_2\big(x, X(x), y, Y(y)\big) = k_1(x, y)\, k_2(X(x), Y(y)). \tag{13}$$

Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.

Cross product kernel on fuzzy sets

Lemma

If k_1 and k_2 are real-valued pd kernels, then the cross product kernel on fuzzy sets is pd.

Cross product kernel on fuzzy sets

Lemma

If k_1 and k_2 are real-valued pd kernels, then the cross product kernel on fuzzy sets is pd.

Corollary

Kernel $k_\times$ defines a similarity measure for two fuzzy sets $X, Y \in \mathcal{F}(\Omega)$ as follows:

$$k_\times(X, Y) = \langle \phi_X, \phi_Y \rangle_{\mathcal{H}}, \tag{14}$$

Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.

Cross product kernel on fuzzy sets

$k_1(x,y)$	$k_\times(X,Y)$
linear	$\sum_{\substack{x\in \mathrm{supp}(X),\\ y\in \mathrm{supp}(Y)}} xyX(x)Y(y)$
polynomial	$\sum_{\substack{x\in \mathrm{supp}(X),\\ y\in \mathrm{supp}(Y)}} (\gamma\langle x,y\rangle + b)^\beta X(x)Y(y)$
exponential	$\sum_{\substack{x\in \mathrm{supp}(X),\\ y\in \mathrm{supp}(Y)}} \exp(\gamma\langle x,y\rangle)X(x)Y(y)$
Gaussian	$\sum_{\substack{x\in \mathrm{supp}(X),\\ y\in \mathrm{supp}(Y)}} \exp(-\gamma\Vert x-y\Vert^2)X(x)Y(y)$

Table: Examples of cross product kernels on fuzzy sets.

Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.

Cross product kernel on fuzzy sets

Example

Let $(\Omega, \mathcal{A}, \mu)$ be a finite measure space. Let $k_{\mathbf{1}}, k_{\mathbf{2}}$ be continuous functions with finite integral. The kernel

$$k_{\times}(X, Y) = \iint_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} k_{\mathbf{1}} \otimes k_{\mathbf{2}}((x, X(x)), (y, Y(y)))\, d\mu(x) d\mu(y), \tag{15}$$

is a cross product kernel on fuzzy sets.

Example

Replacing the measure μ of the previous example with a probability measure $\mathbb{P}$ results in the following cross product kernel on fuzzy sets:

$$k_{\times}(X, Y) = \iint_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} k_{\mathbf{1}} \otimes k_{\mathbf{2}}((x, X(x)), (y, Y(y)))\, d\mathbb{P}(x) d\mathbb{P}(y), \tag{16}$$

Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.

Cross product kernel on fuzzy sets

A generalization of $k_\times$ to deal with a D-tuple of fuzzy sets, i.e., $(X_1, \ldots, X_D) \in \mathcal{F}(\Omega_1) \times \cdots \times \mathcal{F}(\Omega_D)$ is implemented by the following kernel:

$$k_\times^\pi((X_1, \ldots, X_D), (Y_1, \ldots, Y_D)) = \prod_{d=1}^{D} k_\times^d(X_d, Y_d). \tag{17}$$

If all the kernels $k_\times^d$ are positive definite then $k_\times^\pi$ is positive definite by closure properties of kernels. Another generalization based on addition of positive definite kernels is also possible:

$$k_\times^\Sigma((X_1, \ldots, X_D), (Y_1, \ldots, Y_D)) = \sum_{d=1}^{D} \alpha_i k_\times^d(X_d, Y_d). \tag{18}$$

Kernel $k_\times^\Sigma$ is positive definite if only if $\alpha_i \in \mathbb{R}^+$ and all the $k_\times^D$ kernels are positive definite.

Cross product kernel on fuzzy sets

Properties

- $k_{\times}$ is a convolution kernel, i.e., it can be derived from $k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^{L} k_l(e_l, e'_l)$,

Cross product kernel on fuzzy sets

Properties

- $k_\times$ is a convolution kernel, i.e., it can be derived from $k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^{L} k_l(e_l, e'_l)$,
- $k_\times$ generalize the cross product kernel on sets, i.e., $k(A, A')$

Cross product kernel on fuzzy sets

Properties

- $k_\times$ is a convolution kernel, i.e., it can be derived from $k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^{L} k_l(e_l, e'_l)$,
- $k_\times$ generalize the cross product kernel on sets, i.e., $k(A, A')$
- $k_\times$ embeds probability distributions into a RKHS.

Cross product kernel on fuzzy sets

Properties

- $k_\times$ is a convolution kernel, i.e., it can be derived from $k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^{L} k_l(e_l, e'_l)$,
- $k_\times$ generalize the cross product kernel on sets, i.e., $k(A, A')$
- $k_\times$ embeds probability distributions into a RKHS.
- fuzzines and radomness modeling (see example when $\mu = \mathbb{P}$)

Cross product kernel on fuzzy sets

Properties

- $k_\times$ is a convolution kernel, i.e., it can be derived from $k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^{L} k_l(e_l, e'_l)$,
- $k_\times$ generalize the cross product kernel on sets, i.e., $k(A, A')$
- $k_\times$ embeds probability distributions into a RKHS.
- fuzzines and radomness modeling (see example when $\mu = \mathbb{P}$)
- noise resistant under supervised classification experiments on attribute noisy datasets (see Paper below)

Guevara, Jorge, et.al. **"Cross product kernels for fuzzy set similarity."** Fuzzy Systems (FUZZ-IEEE), 2017.

Cross product kernel on fuzzy sets

```
for i=1:n
    for j=i:n
        [L,~] = size(S{i});
        [LL,~]= size(S{j});

        %X*X'
        k = (S{i}* S{j}').* (MF{i} * MF{j})')
        K(i,j)= sum(sum(k))/(L*LL);
        K(j,i)=K(i,j);
    end
end
```

Cross product kernel on fuzzy sets

```
for i=1:n
    for j=i:n
        [L,~] = size(S{i});
        [LL,~]= size(S{j});

        %use your favorite kernels
        k=computeKernel(S{i},S{j},param1,kernelOption).* computeKernel(MF{i},MF{j}),param2,kernelOption)
        K(i,j)= sum(sum(k))/(L*LL);
        K(j,i)=K(i,j);
    end
end
```

The intersection kernel on fuzzy sets

A triangular norm or **T-norm** is the function $T : [0, 1]^2 \to [0, 1]$, such that, for all $x, y, z \in [0, 1]$ satisfies:

- T1 commutativity: $T(x, y) = T(y, x)$;
- T2 associativity: $T(x, T(y, z)) = T(T(x, y), z)$;
- T3 monotonicity: $y \leq z \Rightarrow T(x, y) \leq T(x, z)$;
- T4 boundary condition $T(x, 1) = x$.

The intersection kernel on fuzzy sets

A triangular norm or **T-norm** is the function $T : [0,1]^2 \to [0,1]$, such that, for all $x, y, z \in [0,1]$ satisfies:

- T1 commutativity: $T(x,y) = T(y,x)$;
- T2 associativity: $T(x, T(y,z)) = T(T(x,y),z)$;
- T3 monotonicity: $y \leq z \Rightarrow T(x,y) \leq T(x,z)$;
- T4 boundary condition $T(x,1) = x$.

a multiple-valued extension

Using $n \in \mathbb{N}, n \geq 2$ and associativity, a multiple-valued extension $T_n : [0,1]^n \to [0,1]$ of a T-norm T is given by $T_2 = T$ and

$$T_n(x_1, x_2, \ldots, x_n) = T(x_1, T_{n-1}(x_2, x_3, \ldots, x_n)). \tag{19}$$

We will use T to denote T or T_n.

The intersection kernel on fuzzy sets

A **semi-ring of sets**, $\mathcal{S}$ on Ω, is a subset of the power set $\mathcal{P}(\Omega)$, that is, a set of sets satisfying:

1. $\phi \in \mathcal{S}$, ϕ denotes the empty set;
2. $A, B \in \mathcal{S}, \implies A \cap B \in \mathcal{S}$;
3. for all $A, A_1 \in \mathcal{S}$ and $A_1 \subseteq A$, there exists a sequence of pairwise disjoint sets $A_2, A_3, \ldots A_N \subseteq \mathcal{S}$, such

$$A = \bigcup_{i=1}^{N} A_i.$$

The intersection kernel on fuzzy sets

A **semi-ring of sets**, $\mathcal{S}$ on Ω, is a subset of the power set $\mathcal{P}(\Omega)$, that is, a set of sets satisfying:

1. $\phi \in \mathcal{S}$, ϕ denotes the empty set;
2. $A, B \in \mathcal{S}, \implies A \cap B \in \mathcal{S}$;
3. for all $A, A_1 \in \mathcal{S}$ and $A_1 \subseteq A$, there exists a sequence of pairwise disjoint sets $A_2, A_3, \ldots A_N \subseteq \mathcal{S}$, such

$$A = \bigcup_{i=1}^{N} A_i.$$

Finite decomposition

Condition 3 is called *finite decomposition of A*.

The intersection kernel on fuzzy sets

Definition (Measure)

Let $\mathcal{S}$ be a semi-ring and let $\rho : \mathcal{S} \to [0, \infty]$ be a pre-measure, i.e., ρ satisfy:

1. $\rho(\phi) = 0$;
2. *for a finite decomposition of* $A \in \mathcal{S}$, $\rho(A) = \sum_{i=1}^{N} \rho(A_i)$;

by Carathéodory's extension theorem, ρ is a measure on $\sigma(\mathcal{S})$, where $\sigma(S)$ is the smallest σ-algebra containing S.

Gartner et.al., shows that a kernel $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ defined by $k(A, A') = \rho(A \cap A')$ is positive definite, where $\rho : \mathcal{S} \to [0, \infty]$ is a measure.

Gartner, Thomas. Kernels for structured data. Vol. 72. World Scientific, 2008.

The intersection kernel on fuzzy sets

Remark

Notation $\mathcal{F}_{\mathcal{S}}(\Omega)$ stands for the set of all fuzzy sets over Ω whose support belongs to $\mathcal{S}$, i.e.,

$$\mathcal{F}_{\mathcal{S}}(\Omega) = \{X \subset \Omega |\, \text{supp}(X) \in \mathcal{S}\}.$$

where $\mathcal{S}$ is a semi-ring of sets on Ω

The intersection kernel on fuzzy sets

Remark

Notation $\mathcal{F}_{\mathcal{S}}(\Omega)$ stands for the set of all fuzzy sets over Ω whose support belongs to $\mathcal{S}$, i.e.,

$$\mathcal{F}_{\mathcal{S}}(\Omega) = \{X \subset \Omega | \, \text{supp}(X) \in \mathcal{S}\}.$$

where $\mathcal{S}$ is a semi-ring of sets on Ω

Example

If $X \cap Y \in \mathcal{F}_{\mathcal{S}}(\Omega)$ then satisfy (finite decomposition):

$$\text{supp}(X \cap Y) = \bigcup_{i \in I} A_i, \; A_i \in \mathcal{S},$$

where $\{A_1, A_2, \ldots, A_N\}$. are pairwise disjoint sets

Intersection Kernel on Fuzzy Sets

Example cont.

We can measure $\mathrm{supp}(X \cap Y) = \bigcup_{i \in I} A_i,\ A_i \in \mathcal{S}$ using the measure $\rho : \mathcal{S} \rightarrow [0, \infty]$ as follows:

$$\rho\big(\,\mathrm{supp}(X \cap Y)\big) = \rho(\bigcup_{i \in I} A_i) = \sum_{i \in I} \rho(A_i),$$

Intersection Kernel on Fuzzy Sets

Example cont.

We can measure $\operatorname{supp}(X \cap Y) = \bigcup_{i \in I} A_i,\ A_i \in \mathcal{S}$ using the measure $\rho : \mathcal{S} \to [0, \infty]$ as follows:

$$\rho\big(\operatorname{supp}(X \cap Y)\big) = \rho\Big(\bigcup_{i \in I} A_i\Big) = \sum_{i \in I} \rho(A_i),$$

Adding fuzziness

The idea to include fuzziness is to weight each $\rho(A_i)$ by a value given by the contribution of the membership function on all the elements of the set A_i.

Intersection Kernel on Fuzzy Sets

Definition

The intersection kernel on fuzzy sets is a function:
$k_\cap : \mathcal{F}_\mathcal{S}(\Omega) \times \mathcal{F}_\mathcal{S}(\Omega) \rightarrow \mathbb{R}$, *defined by:*

$$k_\cap(X, Y) = \sum_{i \in I} \big(X \cap Y\big)(A_i)\rho(A_i), \tag{20}$$

where $\big(X \cap Y\big)(A) \equiv \sum_{x \in A} \big(X \cap Y\big)\ (x)$

Guevara, Jorge, et.al. **"Positive Definite Kernel Functions on Fuzzy Sets."** Fuzzy Systems (FUZZ-IEEE), 2014.

Intersection Kernel on Fuzzy Sets

Kernel $k_\cap$ can be implemented via T-norm operators:

$$\begin{aligned}
k_\cap(\mathrm{X,Y}) &= \sum_{i \in I} \big(X \cap Y\big)(A_i)\rho(A_i) \\
&= \sum_{i \in I} \sum_{x \in A_i} \big(X \cap Y\big)(x)\rho(A_i) \\
&= \sum_{i \in I} \sum_{x \in A_i} T(X(x), Y(x))\rho(A_i)
\end{aligned}$$

Intersection Kernel on Fuzzy Sets

Some kernel examples for different T-norm operators **Examples**

Kernel $k_\cap$	T-norm
$k_{\cap\_\min}(X,Y) = \sum_{i \in I} \sum_{x \in A_i} \min(X(x), Y(x))\rho(A)$	minimum
$k_{\cap\_\mathrm{pro}}(X,Y) = \sum_{i \in I} \sum_{x \in A_i} X(x)Y(x)\rho(A)$	product
$k_{\cap\_\text{Łuk}}(X,Y) = \sum_{i \in I} \sum_{x \in A_i} \max(X(x) + Y(x) - 1, 0)\rho(A)$	Łukasiewicz
$k_{\cap\_\mathrm{Dra}}(X,Y) = \sum_{i \in I} \sum_{x \in A_i} f(X(x), Y(x))\rho(A)$	Drastic

where f is defined as

$$f(X(x), Y(x)) = \begin{cases} X(x), \text{ if } Y(x) = 1 \\ Y(x), \text{ if } X(x) = 1 \\ 0, \textit{ otherwise} \end{cases}$$

Intersection Kernel on Fuzzy Sets

Lemma

$k_{\min}(X, Y) = \sum_{i\in I} \sum_{x\in A_i} \min(\mu_X(x), \mu_Y(x))\rho(A_i)$
is positive definite

Lemma

$k_P(X, Y) = \sum_{i\in I} \sum_{x\in A_i} \mu_X(x)\mu_Y(x)\rho(A_i)$
is positive definite.

Intersection Kernel on Fuzzy Sets

Lemma

$k_{\min}(X, Y) = \sum_{i \in I} \sum_{x \in A_i} \min(\mu_X(x), \mu_Y(x))\rho(A_i)$
is positive definite

Lemma

$k_P(X, Y) = \sum_{i \in I} \sum_{x \in A_i} \mu_X(x)\mu_Y(x)\rho(A_i)$
is positive definite.

```
%membershipFunction(type, points)
K(X,Y)=sum(min(X(data(i,:)),Y(data(i,:))))
```

The non-singleton kernel on fuzzy sets

Definition

This kernel is a function $k_{nsk} : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \to [0,1]$ *defined by:*

$$\begin{aligned} k_{nsk}(X, Y) &= \sup_{x \in \Omega} \Big(X \cap Y\Big)(x) \\ &= \sup_{x \in \Omega} \Big(T\big(X(x), Y(x)\big) \Big), \end{aligned}$$

where sup *is the supremum.*

Derived from non-singleton fuzzy systems.

Guevara, Jorge, et.al.**"Kernel Functions in Takagi-Sugeno-Kang Fuzzy System with Nonsingleton Fuzzy Input."**

The non-singleton kernel on fuzzy sets

Examples Given $X = (X_1, \ldots, X_d, \ldots, X_D)$ and $Y = (Y_1, \ldots, X_d, \ldots, Y_D)$, such that: $X_d(.) = \exp\left(-\frac{1}{2}\frac{(.-m_d)^2}{\sigma_d^2}\right)$, where, $m_d \in \mathbb{R}$ amd $\sigma_d \in \mathbb{R}^+$, then, the following kernel

$$k_{nsk}(X, Y) = \prod_{d=1}^{D} \exp\left(-\frac{1}{2}\frac{(m_d - m'_d)^2}{\sigma_d^2 + (\sigma'_d)^2}\right), \tag{21}$$

is a positive definite kernel.

Guevara, Jorge, et.al.**"Kernel Functions in Takagi-Sugeno-Kang Fuzzy System with Nonsingleton Fuzzy Input."**

The non-singleton kernel on fuzzy sets

Examples Given $X = (X_1, \ldots, X_d, \ldots, X_D)$ and $Y = (Y_1, \ldots, X_d, \ldots, Y_D)$, such that: $X_d(.) = \exp\left(-\frac{1}{2}\frac{(.-m_d)^2}{\sigma_d^2}\right)$, where, $m_d \in \mathbb{R}$ amd $\sigma_d \in \mathbb{R}^+$, then, the following kernel

$$k_{nsk}^{\gamma}(X, Y) = \prod_{d=1}^{D} \exp\left(-\frac{1}{2}\frac{(m_d - m'_d)^2}{\sigma_d^2 + (\sigma'_d)^2 + \gamma}\right), \tag{22}$$

is a positive definite kernel.

Guevara, Jorge, et.al."**Kernel Functions in Takagi-Sugeno-Kang Fuzzy System with Nonsingleton Fuzzy Input.**"

The non-singleton kernel on fuzzy sets

```
%sigmas
stdX
stdZ
%mus
X
Z

%------------
[m,~]=size(X);
[p,~]=size(Z);

K=zeros(m,p);
for i=1:m
    for j=1:p
        diff=( X(i,:)- Z(j,:)).*( X(i,:)- Z(j,:));
        den=stdX(i,:).*stdX(i,:) +  stdZ(j,:).*stdZ(j,:);
        K(i,j)=exp(- 0.5*sum(diff./den)  );
    end
end
```

Distance-based kernels on fuzzy sets

Based on the concept of *distance substitution kernels*. **Examples** Kernel $K_D(X, X') = \exp(-\lambda D(X, X')^2)$,is PD when we use the following metric on fuzzy sets: $D(X, X') = \dfrac{\sum_{x\in\Omega} |X(x) - X'(x)|}{\sum_{x\in\Omega} |X(x) + X'(x)|}$.

Guevara, Jorge, et.al. **"Fuzzy Set Similarity using a Distance-Based Kernel on Fuzzy Sets"**, 2016, Book Chapter, Handbook of Fuzzy Sets Comparison - Theory, Algorithms and Applications,pages 103-120

Outline

Support fuzzy-set machines

Definition (Support fuzzy-set machines)

Kernels machines with kernel gram matrix constructed by kernels on fuzzy sets.

Support fuzzy-set machines

Definition (Support fuzzy-set machines)

Kernels machines with kernel gram matrix constructed by kernels on fuzzy sets.

Support fuzzy-set machines learn $f = \sum_i \alpha_i k(X,)$

Support fuzzy-set machines

Definition (Support fuzzy-set machines)

Kernels machines with kernel gram matrix constructed by kernels on fuzzy sets.

Support fuzzy-set machines learn $f = \sum_i \alpha_i k(X,)$

Definition (Support fuzzy-sets)

Is the set given by all the fuzzy sets used in a kernel machine such the correspondent $\alpha_i > 0$.

Practical example on supervised classification on noisy data:

Table: Summary of the PIMA and SONAR attribute noise datasets

Dataset	%Noise	Dataset	%Noise
pima-5an-nn	5%		
pima-10an-nn	10%		
pima-15an-nn	15%		
pima-20an-nn	20%		
pima-5an-nc	5%		
pima-10an-nc	10%		
pima-15an-nc	15%		
pima-20an-nc	20%		

Pima, 768 observations, 35/65 class rate Sonar, 208 observations, 47/53 class rate

---

Algorithm 1 First fuzzification approach

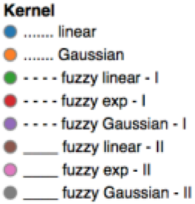
Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Output: $\mathcal{MF} = \{(x_i, X_1(x_i^1), \ldots, X_D(x_i^D), y_i)\}_{i=1}^{N}$
  for each class y_i **do**
    for $d = 1$ to D **do**
      $q_1, q_2, q_3 = quantile(x_{1\leq i \leq N}^d, (0.25, 0.5, 0.75))$
      $\mu_d = q_2$
      $\sigma_d = |q_3 - q_1| / (2 * \sqrt{2 * \log 2})$
      $X_d(.) = \exp(-0.5(. - \mu_d)^2 / \sigma_d^2)$
    end for
  end for

Support fuzzy-set machines

Algorithm 2 Second fuzzification approach

Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Output: $\mathcal{MF} = \left\{\left(x_i, X_1(x_i^1), \ldots, X_D(x_i^D), y_i\right)\right\}_{i=1}^{N}$
  for each class y_i **do**
    for $d = 1$ to D **do**
      $h = histogram(x^d_{1 \leq i \leq N})$
      $h = h / \max(h)$
      $X_d(.) = linearInterpolation(h)$
    end for
  end for

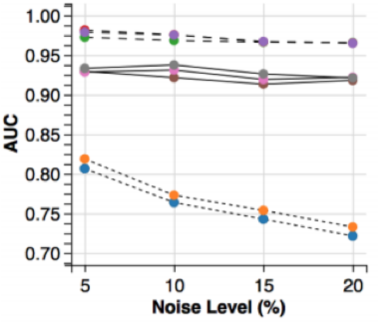
Kernels used on the experiment



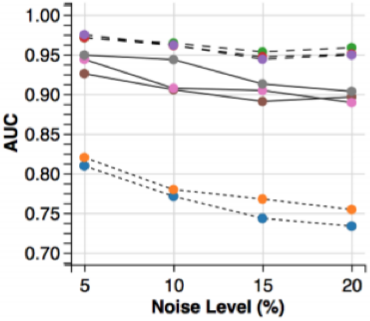
$k_1(x, y)$	$k_\times(X, Y)$
Fuzzy linear	$\sum_{x \in \text{supp}(X), y \in \text{supp}(Y)} xyX(x)Y(y)$
Fuzzy exp	$\sum_{x \in \text{supp}(X), y \in \text{supp}(Y)} \exp(\gamma \langle x, y \rangle)X(x)Y(y)$
Fuzzy Gaussian	$\sum_{x \in \text{supp}(X), y \in \text{supp}(Y)} \exp(-\gamma \|x - y\|^2)X(x)Y(y)$

Table

Support fuzzy-set machines



(a)



(b)

Conclusions and next steps