Kernels & RKHS

Bülent Karasözen

Institute of Applied Mathematics & Department of Mathematics
Middle East Technical University, Ankara

Kernels and feature spaces

Hilbert spaces & RKHS

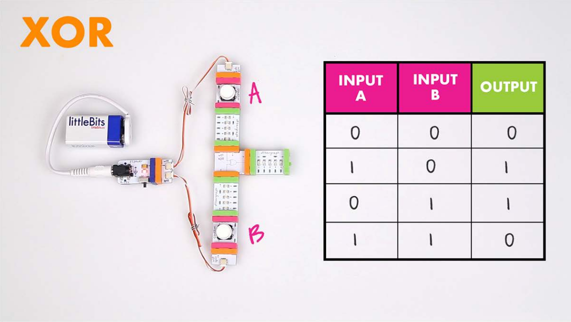
Machine Learning applications

Real world applications

Motivating examples

What is a kernel, how do we construct it?

The XOR gate acts in the same way as the logical "either/or."
The output is "true" if either, but not both, of the inputs are "true."
The output is "false" if both inputs are "false" or if both inputs are "true."



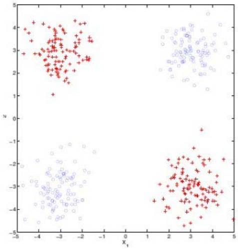
INPUT A	INPUT B	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	0

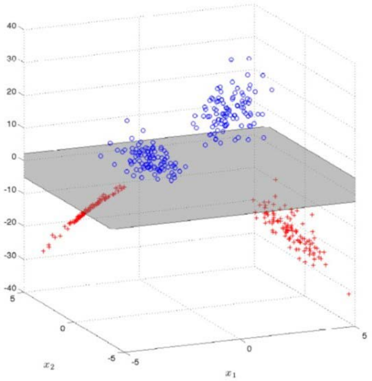
Motivating examples

Using a linear classifier, the red patterns can not be separated from the blue ones.

In the higher dimensional feature space, they are linearly separable

$$\phi(x) = [x_1, x_2, x_1 x_2] \in \mathscr{X}(x) = \mathbb{R}^2$$





Motivating examples

Many classical learning algorithms—such as the perceptron, support vector machine (SVM) and principal component analysis (PCA) employ data instances, e.g., $x, x' \in \mathbb{R}^n$, only through an inner product (x, x'), which basically is a similarity measure between x and x'.

The class of linear functions induced by this inner product may be too restrictive for many real-world problems.

Kernel methods aim to build more flexible and powerful learning algorithms by replacing (x, x') with non-linear, similarity measure.

Feature spaces can be used to compare objects which have much more complex structure; strings, graphs.

Motivating examples

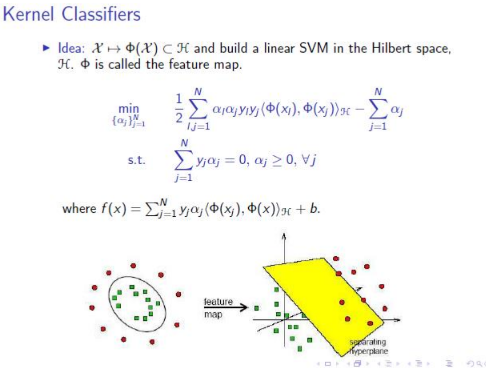
Learning algorithms are defined in terms of dot products between the features, where these dot products can be computed in closed form (kernel trick)

The term “kernel” simply refers to a dot product between (possibly infinitely many) features.

Kernel methods can be viewed as nonlinear versions of linear algorithms.

Motivating examples

The classification of objects with support vector machines (SVM)



Kernel Classifiers

- Idea: $\mathcal{X} \mapsto \Phi(\mathcal{X}) \subset \mathcal{H}$ and build a linear SVM in the Hilbert space, $\mathcal{H}$. Φ is called the feature map.

$$\min_{\{\alpha_j\}_{j=1}^N} \quad \frac{1}{2}\sum_{l,j=1}^{N} \alpha_l \alpha_j y_l y_j \langle \Phi(x_l), \Phi(x_j) \rangle_{\mathcal{H}} - \sum_{j=1}^{N} \alpha_j$$

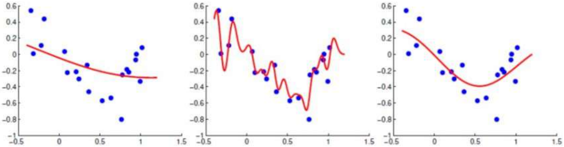
$$\text{s.t.} \quad \sum_{j=1}^{N} y_j \alpha_j = 0,\ \alpha_j \geq 0,\ \forall j$$

where $f(x) = \sum_{j=1}^{N} y_j \alpha_j \langle \Phi(x_j), \Phi(x) \rangle_{\mathcal{H}} + b$.

Motivating examples

Kernel methods can be used to control smoothness of a function used in regression or classification.

Different parameter choices determine whether the regression function overfits, underfits, or fits optimally.



Kernels and feature spaces

Hilbert spaces & RKHS

Machine Learning applications

Real world applications

Hilbert space

Inner product: Let $\mathscr{H}$ be a vector space over $\mathbb{R}$.

A function $(\cdot,\cdot)_{\mathscr{H}}$ is said to be an inner product on $\mathscr{H}$ if

1. $(\alpha_1 f_1 + \alpha_2 f_2, g)_{\mathscr{H}} = \alpha_1 (f, g)_{\mathscr{H}} + \alpha_2 (f_2, g_{\mathscr{H}})$
2. $(f, g)_{\mathscr{H}} = (g, f)_{\mathscr{H}}$
3. $(f, f)_{\mathscr{H}} \geq 0$ and $(f, f)_{\mathscr{H}} = 0$ if and only if $f = 0$.

Norm: $||f|| := \sqrt{(f, f)_{\mathscr{H}}}$.

A Hilbert space is a space on which an inner product is defined, along with the condition that it contains the limits of all Cauchy sequences of functions.

Kernel

Let $\mathscr{X}$ be a non-empty set. A function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is called a kernel if there exists an real Hilbert space and a map $\phi : \mathscr{X} \to \mathbb{H}$ such that $\forall x, x' \in \mathscr{X}$

$$k(x; x') := (\phi(x), \phi(x'))_{\mathscr{H}}.$$

All kernel functions are positive definite. If we have a positive definite function, we know there exists one (or more) feature spaces for which the kernel defines the inner product - it is not necessary to define the feature spaces explicitly.

Positive definite kernel: A symmetric function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is positive definite if $\forall n \geq 1$, $\forall (a_1, \ldots, a_n) \in \mathbb{R}^n, \forall (x_1, \ldots, x_n) \in \mathscr{X}$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, k_j) \geq 0$$

The kernel $k(\cdot, \cdot)$ is strictly positive definite if for mutually distinct x_i, the equality holds only when all the a_i are zero.

Properties of kernels

- Sum of kernels are kernels : Given $\alpha > 0$ and k, k_1, k_2 kernels on $\mathscr{X}$, then αk and $k_1 + k_2$ are kernels on $\mathscr{X}$.
- A difference of kernels may not be a kernel: if $k_1(x,x) - k_2(x,x) < 0$.
- Products of kernels are kernels: Given k_1 on $\mathscr{X}_1$ and k_2 on $\mathscr{X}_2$ than $k_1 \times k_2$ i s a kernel on $\mathscr{X}_1 \times \mathscr{X}_1$.
- If $\mathscr{X}_1 = \mathscr{X}_1 = \mathscr{X}$, then $k =: k_1 \times k_2$ is a kernel on $\mathscr{X}^2$.

Some common kernels

- Polynomial kernel $k(x,x') := ((x,x')+c)^m, \quad c > o,\ m \geq 1$
- Exponential kernel on $\mathbb{R}^d$ $k(x,x') := exp((x,x'))$
- Gaussian kernel on $\mathbb{R}^d$ $k(x,x') := exp(-\gamma^{-2}||x-x'||^2)$

The Gaussian kernel is translation-invariant,

$k_\sigma(x,z) = g_\sigma(x-z)$, where $g_\sigma(x) = \exp - \frac{\|x\|^2}{2\sigma^2}$.

Kernels are unique, but the feature maps are not unique.

$$\mathscr{X} \in \mathbb{R}^2, \text{ and } k(x,y) = (x,y)^2$$

,

$$k(x,y) = x_1^2 x_2^2 + y_1^2 y_2^2 + 2x_1 x_2 y_1 y_2$$

$$\phi_1(x) = (x_1^2\ x_2^2\ \sqrt{2}x_1 x_2), \quad \phi_1(y) = (y_1^2\ y_2^2\ \sqrt{2}y_1 y_2)$$

$$\phi_2(x) = (x_1^2\ x_2^2\ x_1 x_2\ x_1 x_2), \quad \phi_2(y) = (y_1^2\ y_2^2\ y_1 y_2\ y_1 y_2)$$

RKHS

Let $\mathscr{H}$ an Hilbert space of functions $f : \mathscr{X} \to \mathbb{R}$ with inner product $(\cdot,\cdot)_{\mathscr{H}}$. Then $\mathscr{H}$ is called a RKHS on $\mathscr{X}$ if there exists a function $k : \{\mathscr{X} \times \mathscr{X} \to \mathbb{R}$ (the reproducing kernel)$\}$ such that

1. $k(\cdot,x) \in \mathscr{H}$ for all $x \in \mathscr{X}$,
2. $(f, k(\cdot,x))_{\mathscr{H}} = f(x)$ for all $x \in \mathscr{X}$, $f \in \mathscr{H}$ (reproducing property).

The reproducing property is equivalent to state that, for $x \in \mathscr{X}$, the x-translate $k(\cdot,x)$ of the kernel is the Riesz representer of the evaluation functional $\delta_x : \mathscr{H} \to \mathbb{R}$, $\delta_x(f) := f(x)$ for $f \in \mathscr{H}$, that is hence a continuous functional in $\mathscr{H}$. Also the converse holds.

RKHS

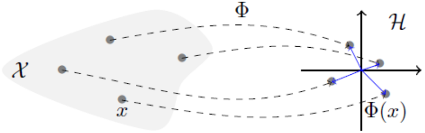
Let $\mathscr{H}$ be a RKHS on Ω with reproducing kernel k. Let $n, n' \in \mathbb{N}$, $\alpha \in \mathbb{R}^n$, $\alpha' \in \mathbb{R}^{n'}$, $X_n, X'_{n'} \subset \mathscr{X}$, and define the functions

$$f(x) := \sum_{i=1}^{n} \alpha_i k(x, x_i), \ \ g(x) := \sum_{j=1}^{n'} \alpha'_j K(x, x'_j), \ \ x \in \mathscr{X}.$$

1. $f, g \in \mathscr{H}$,
2. $(f,)_{\mathscr{H}} = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \alpha'_j K(x_i, x'_j)$.
3. k is the unique reproducing kernel of $\mathscr{H}$ and it is a positive definite kernel.

Theorem (Aronszajn), 1950)

k is a p.d. kernel on $\mathscr{X}$ if and onșly if there exists a Hilbert space $\mathscr{H}$ and a mapping $\phi : \mathscr{X} \to \mathscr{H}$, such that for any $x, x' \in \mathscr{X}$ $k(x, x') = (\phi(x), \phi(x'))_{\mathscr{H}}$



Matrix-valued kernels & kernel trick

In the case of RKHS for vector-valued functions a separable Gaussian kernel can be used

$$K_\sigma(\mathbf{x},\mathbf{z}) = k_\sigma(\mathbf{x},\mathbf{z}) I_n \quad \in \mathbb{R}^n$$

Computational savings of the kernel trick

Polynomial kernel of degree p $\phi(x,y) = (1 + x^T y))^p$.

Computation of the inner product $\alpha = x^T y$ requires $\mathscr{O}(n)$ operations. $\phi(x,y) = (1+\alpha)^d$.

Computing as product of the feature map of vector $\phi(x)$ of length $\mathscr{O}(n^2)$

$$\phi(x) = (1\ x_1\ \ldots x_n\ x_1^2\ \ldots x_n^2\ x_1^p\ \ldots x_n^p\ x_1x_2\ x_1x_3 \ldots x_1x_p\ x_2x_3 \ldots x_px_p)$$

requires $\mathscr{O}\binom{n+p}{n}$ operations for $p = 20,\ n = 100 \approx 3 \cdot 10^{22}$ operations.

Smoothness functional

for any function $f \in \mathscr{H}$ and any two pints $x, x' \in \mathscr{X}$

$$\begin{aligned} |f(x) - f(x')| &= |(f, k_x - k_{x'})_{\mathscr{H}}| \\ &\leq ||f||_{\mathscr{H}}\, ||k_x - k_{x'}||_{\mathscr{H}} \\ &= ||f||_{\mathscr{H}}\, d_k(x, x') \end{aligned}$$

Distance in the feature space

$$\begin{aligned} d_k(x, x')^2 &= ||\phi(x_1) - \phi(x_2)||^2_{\mathscr{H}} \\ d_k(x, x')^2 &= k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2) \end{aligned}$$

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

Kernels and feature spaces

Hilbert spaces & RKHS

Machine Learning applications

Real world applications

Kernel principal decomposition analysis (PCA)

Classical PCA: Find a d-dimensional subspace of a higher dimensional subspace $\mathbb{R}^D$ containing the direction of maximum variance

$$
\begin{aligned}
u_1 &= \arg\max_{|u|| \leq 1} \frac{1}{n} \sum_{i=1}^{n} \left(u^T \left(x_i - \frac{1}{n} \sum_{i=1}^{n} x_i \right) \right)^2 \\
u_1 &= \arg\max_{|u|| \leq 1} u^T C u
\end{aligned}
$$

Covariance matrix

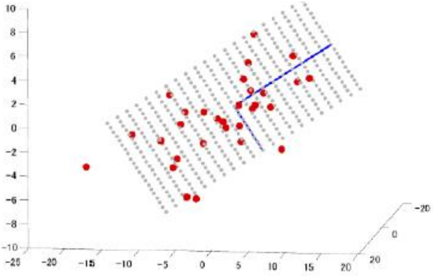
$$
\begin{aligned}
C &= \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \frac{1}{n} \sum_{i=1}^{n} x_i \right) \left(x_i - \frac{1}{n} \sum_{i=1}^{n} x_i \right) \\
u_1 &= \frac{1}{n} XHX
\end{aligned}
$$

$X = [x_1, \ldots, x_n], H = I - n^{-1} \mathbf{1}_{n \times n},$ where $\mathbf{1}_{n \times n} n \times n$ matrix of ones

Principal components u_i are eigenvectors of the covariance matrix C

$$\lambda_i u_i = C u_i$$

Principal component analysis (PCA



Kernel PCA

$$
\begin{aligned}
f_1 &= \arg \max_{|f|||_{\mathscr{H}} \leq 1} \frac{1}{n} \sum_{i=1}^{n} \left(\left((f, \phi(x_i)) - \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \right)_{\mathscr{H}} \right)^2 \\
f_1 &= \arg \max_{|f||_{\mathscr{H}} \leq 1} \operatorname{var}(f)
\end{aligned}
$$

Covariance matrix

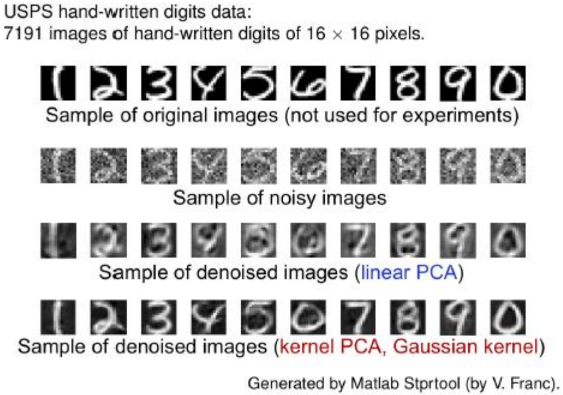
$$
C = \frac{1}{n} \sum_{i=1}^{n} \left(\phi(x_i) - \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \right) \otimes \left(\phi(x_i) - \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \right)
$$

where

$$
(a \otimes b)c := (b, c)_{\mathscr{H}} a
$$

analogous to the outer product of vectors.

Denoising hand-written digits by PCA



USPS hand-written digits data:
7191 images of hand-written digits of 16 × 16 pixels.

Sample of original images (not used for experiments)

Sample of noisy images

Sample of denoised images (linear PCA)

Sample of denoised images (kernel PCA, Gaussian kernel)

Generated by Matlab Stprtool (by V. Franc).

Classical ridge regression

Training points arranged in a matrix $X = [x_, \ldots, x_n] \in \mathbb{R}^D$. To each of these points, there corresponds and output y_i, arranged in a column vector $y = [y_1, \ldots, y_2]^T$.

$$
\begin{aligned}
a^* &= \arg\min_{a \in \mathbb{R}^D} \left(\sum_{i=1}^{n} (y_i - x_i^t a)^2 + \lambda ||a||^2 \right) \\
a^* &= \arg\min_{a \in \mathbb{R}^D} \left(||y - X^T a||2 + \lambda ||a||^2 \right)
\end{aligned}
$$

The regularized least squares solution

$$a^* = (XX^T + \lambda I)^{-1} Xy$$

Solution with singular value decomposition (SVD)

Kernel ridge regression

$$a^* = \arg\min_{a \in \mathscr{H}} \left(\sum_{i=1}^{n} (y_i - (a, \phi(x - i))_{\mathscr{H}})^2 + \lambda ||a||_{\mathscr{H}}^2 \right)$$

$$X = [\phi(x_1) \ldots \phi(x_n)],\ XX^T = \sum_{i=1}^{n} \phi(x_i) \otimes \phi(x_i),\ (XX^T)_{ij} = (\phi(x_i), \phi(x_j))_{\mathscr{H}}) = k(x_i, x_j)$$

Solution

$$a^* = (K + \lambda I_n)^{-1} y$$

Kernel ridge regression

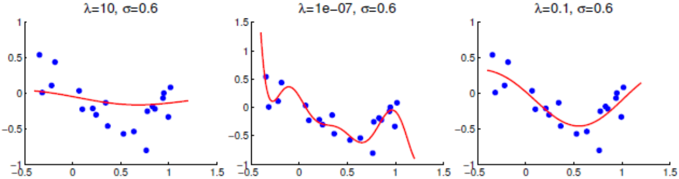
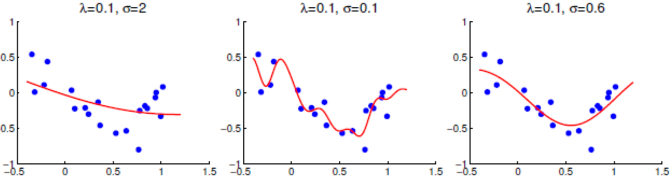


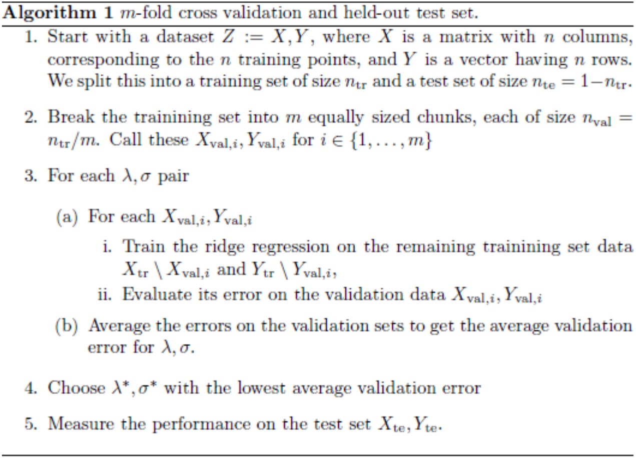
Figure 7.1: Effect of choice of λ on the fit of ridge regression.



Kernel ridge regression

- *Underfitting* Too large a λ resulting very smooth function following the shape of the underlying data with a small prediction error.
- *Overfitting* Too small a λ fitting small fluctuations in the data due to noise, at the expense of smoothness
- An apparently good choice is $\lambda = 0.1$, where the regression curve fits the underlying trend without being overly influenced by noise.
- The kernel width σ affects the fit of ridge regression. Too large a σ results in underfitting: the regression function is too smooth. Too small a σ results in overfitting.
- λ and σ can be chosen by m-fold cross-validation to evaluate the resulting performance of the learning algorithm.

Kernel ridge regression



Algorithm 1 m-fold cross validation and held-out test set.

```
1. Start with a dataset $Z := X, Y$, where $X$ is a matrix with $n$ columns,
   corresponding to the $n$ training points, and $Y$ is a vector having $n$ rows.
   We split this into a training set of size $n_{\text{tr}}$ and a test set of size $n_{\text{te}} = 1 - n_{\text{tr}}$.
2. Break the trainining set into $m$ equally sized chunks, each of size $n_{\text{val}} =$
   $n_{\text{tr}}/m$. Call these $X_{\text{val},i}, Y_{\text{val},i}$ for $i \in \{1, \dots, m\}$
3. For each $\lambda, \sigma$ pair
   (a) For each $X_{\text{val},i}, Y_{\text{val},i}$
       i. Train the ridge regression on the remaining trainining set data
          $X_{\text{tr}} \setminus X_{\text{val},i}$ and $Y_{\text{tr}} \setminus Y_{\text{val},i}$,
       ii. Evaluate its error on the validation data $X_{\text{val},i}, Y_{\text{val},i}$
   (b) Average the errors on the validation sets to get the average validation
       error for $\lambda, \sigma$.
4. Choose $\lambda^*, \sigma^*$ with the lowest average validation error
5. Measure the performance on the test set $X_{\text{te}}, Y_{\text{te}}$.
```

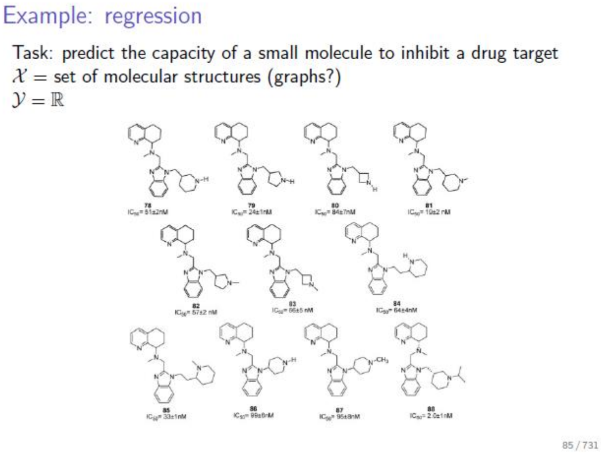
Kernels and feature spaces

Hilbert spaces & RKHS

Machine Learning applications

Real world applications

Regression



Example: regression

Task: predict the capacity of a small molecule to inhibit a drug target
$\mathcal{X}$ = set of molecular structures (graphs?)
$\mathcal{Y} = \mathbb{R}$

78 IC_{50}= 51±2nM

79 IC_{50}= 24±1nM

80 IC_{50}= 84±7nM

81 IC_{50}= 19±2 nM

82 IC_{50}= 57±2 nM

83 IC_{50}= 66±5 nM

84 IC_{50}= 64±4nM

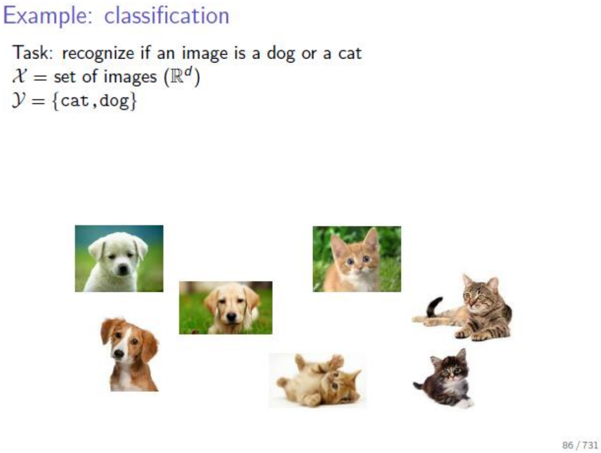
85 IC_{50}= 33±1nM

86 IC_{50}= 99±6nM

87 IC_{50}= 95±8nM

88 IC_{50}= 2.0±1nM

Classification



Example: classification

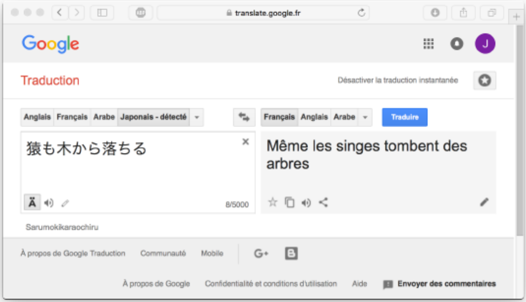
Task: recognize if an image is a dog or a cat
$\mathcal{X}$ = set of images ($\mathbb{R}^d$)
$\mathcal{Y} = \{\texttt{cat}, \texttt{dog}\}$

86 / 731

Structured output

Example: structured output

Task: translate from Japanese to French
$\mathcal{X}$ = finite-length strings of japanese characters
$\mathcal{Y}$ = finite-length strings of french characters



猿も木から落ちる

Sarumokikaraochiru

Même les singes tombent des arbres

Lectures & Software

Arthur Gretton

https://www.gatsby.ucl.ac.uk/~gretton/teaching.html

Julien Mairal and Jean-Philippe Vert

https://members.cbio.mines-paristech.fr/~jvert/svn/kernelcourse/course/2020mva/

Codpy: Curse of dimensionality in Python

https://pypi.org/project/codpy/

https://scikit-learn.org/stable/modules/svm.html

B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi.
Elements of Dimensionality Reduction and Manifold Learning.
Springer International Publishing, 2023.

Jonathan H. Manton and Pierre-Olivier Amblard.
A primer on reproducing kernel Hilbert spaces.
Foundations and Trends® in Signal Processing, 8(1-2):1–126, 2014.

Vern I. Paulsen and Mrinal Raghupathi.
An Introduction to the Theory of Reproducing Kernel Hilbert Spaces.
Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.

Sergei Pereverzyev.
An introduction to artificial intelligence based on reproducing kernel Hilbert spaces.
Compact Textbooks in Mathematics. Birkhäuser/Springer, Cham, 2022.

José Luis Rojo-Álvarez, Manel Martínez-Ramón, Jordi Muñoz-Marí, and Gustau Camps-Valls.
Kernel Functions and Reproducing Kernel Hilbert Spaces, chapter 4, pages 165–207.
John Wiley & Sons, Ltd, 2018.

Bernhard Schölkopf and Alexander J. Smola.

Learning with kernels : support vector machines, regularization, optimization, and beyond.
MIT Press, 2002.

Joe Suzuki.
Kernel methods for machine learning with math and Python—100 exercises for building logic.
Springer, Singapore, [2022] ©2022.

Haitao Zhao, Zhihui Lai, Henry Leung, and Xianyi Zhang.
Feature Learning and Understanding - Algorithms and Applications.
Springer, 2020.