Kernel methods

and why we should love them

(Els mètodes kernel i el perquè els hauríem d'estimar)

Lluís A. Belanche
belanche@cs.upc.edu

Dept. de Ciències de la Computació
Universitat Politècnica de Catalunya

Jornada Pearson @ FME
Barcelona, March 8, 2017

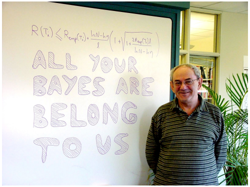
FACULTAT D'INFORMÀTICA DE BARCELONA

Soft Computing Research Group

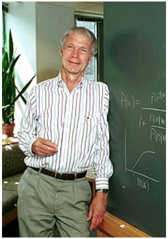
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

A Machine Learning Hall of Fame

Leo Breiman (1928-2005)



Vladimir Vapnik (1936-)



John Hopfield (1933-)

Yann LeCun (1960-)

Reminder of the ridge regression framework

We have a finite i.i.d. **learning data** sample of N observations
$D = \{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}$ $\mathbf{x}_n \in \mathbb{R}^d$, $t_n \in \mathbb{R}$

Statistics: estimation of a continuous random variable T conditioned on a random vector $\mathbf{X}$

Mathematics: estimation of a real function f based on a finite number of noisy examples $\mathbf{x}$

1. The departing **model** is

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \varepsilon_n, \qquad \mathbf{x}_n \in \mathbb{R}^d,\ t_n \in \mathbb{R}$$

ε_n is a continuous r.v. such that

- $\mathbb{E}[\varepsilon_n] = 0$
- $Var[\varepsilon_n] = \sigma_n^2 < \infty$

Reminder of the ridge regression framework

We have a finite i.i.d. **learning data** sample of N observations $D = \{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}$ $\mathbf{x}_n \in \mathbb{R}^d$, $t_n \in \mathbb{R}$

Statistics: estimation of a continuous random variable T conditioned on a random vector $\mathbf{X}$

Mathematics: estimation of a real function f based on a finite number of noisy examples $\mathbf{x}$

1 The departing **model** is

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \varepsilon_n, \qquad \mathbf{x}_n \in \mathbb{R}^d,\ t_n \in \mathbb{R}$$

ε_n is a continuous r.v. such that

- $\mathbb{E}[\varepsilon_n] = 0$
- $Var[\varepsilon_n] = \sigma_n^2 < \infty$

Reminder of the ridge regression framework

We have a finite i.i.d. **learning data** sample of N observations $D = \{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}$ $\mathbf{x}_n \in \mathbb{R}^d, t_n \in \mathbb{R}$

Statistics: estimation of a continuous random variable T conditioned on a random vector $\mathbf{X}$

Mathematics: estimation of a real function f based on a finite number of noisy examples $\mathbf{x}$

1. The departing **model** is

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \varepsilon_n, \qquad \mathbf{x}_n \in \mathbb{R}^d,\ t_n \in \mathbb{R}$$

ε_n is a continuous r.v. such that

- $\mathbb{E}[\varepsilon_n] = 0$
- $Var[\varepsilon_n] = \sigma_n^2 < \infty$

Reminder of the ridge regression framework

2. In linear regression, $y(\mathbf{x};\mathbf{w}) := \mathbf{w}^\top \mathbf{x}$ and $\varepsilon_n \sim \mathcal{N}(0,\sigma^2)$
3. Our **statistical model** is $T_n \sim \mathcal{N}(y(\mathbf{X}_n;\mathbf{w}),\sigma^2)$ or:

$$p(t_n|\mathbf{x}_n;\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}\left(t_n - \mathbf{w}^\top \mathbf{x}_n\right)^2\right),$$

with **parameters** $\{w_0, w_1, \ldots, w_d, \sigma^2\}$
and **input data** $\mathbf{x}_n := (1, \mathbf{x}_n^\top)^\top$

4. Defining $\mathbf{t} := (t_1,\ldots,t_N)^\top, X_{N\times(d+1)}$, a **maximum likelihood** argument leads to the minimization of the regularized (= penalized) **mean empirical error**:

$$E_\lambda(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{t} - X\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2 \,, \lambda > 0$$

($\lambda > 0$ defines a trade-off between the fit to the data and the complexity of the model)

Reminder of the ridge regression framework

2. In linear regression, $y(\mathbf{x};\mathbf{w}) := \mathbf{w}^\top\mathbf{x}$ and $\varepsilon_n \sim \mathcal{N}(0,\sigma^2)$
3. Our **statistical model** is $T_n \sim \mathcal{N}(y(\mathbf{X}_n;\mathbf{w}),\sigma^2)$ or:

$$p(t_n|\mathbf{x}_n;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\left(t_n - \mathbf{w}^\top\mathbf{x}_n\right)^2\right),$$

with **parameters** $\{w_0, w_1, \ldots, w_d, \sigma^2\}$
and **input data** $\mathbf{x}_n := (1, \mathbf{x}_n^\top)^\top$

4. Defining $\mathbf{t} := (t_1,\ldots,t_N)^\top, X_{N\times(d+1)}$, a **maximum likelihood** argument leads to the minimization of the regularized (= penalized) **mean empirical error**:

$$E_\lambda(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{t} - X\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2, \lambda > 0$$

($\lambda > 0$ defines a trade-off between the fit to the data and the complexity of the model)

Reminder of the ridge regression framework

2 In linear regression, $y(\mathbf{x};\mathbf{w}) := \mathbf{w}^\top \mathbf{x}$ and $\varepsilon_n \sim \mathcal{N}(0,\sigma^2)$

3 Our **statistical model** is $T_n \sim \mathcal{N}(y(\mathbf{X}_n;\mathbf{w}),\sigma^2)$ or:

$$p(t_n|\mathbf{x}_n;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\left(t_n - \mathbf{w}^\top \mathbf{x}_n\right)^2\right),$$

with **parameters** $\{w_0, w_1, \ldots, w_d, \sigma^2\}$
and **input data** $\mathbf{x}_n := (1, \mathbf{x}_n^\top)^\top$

4 Defining $\mathbf{t} := (t_1, \ldots, t_N)^\top, X_{N\times(d+1)}$, a **maximum likelihood** argument leads to the minimization of the regularized (= penalized) **mean empirical error**:

$$E_\lambda(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{t} - X\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2 \ , \lambda > 0$$

($\lambda > 0$ defines a trade-off between the fit to the data and the complexity of the model)

Reminder of the ridge regression framework

5 Setting $\nabla E_\lambda(\mathbf{w}) = \mathbf{0}$, we obtain the (regularized) normal equations:

$$X^\top(\mathbf{t} - X\mathbf{w}) = N\lambda\mathbf{w}$$

with solution $\hat{\mathbf{w}} = (X^\top X + \lambda N\mathbf{I})^{-1}X^\top\mathbf{t}$

$$\Longrightarrow y(\mathbf{x}; \hat{\mathbf{w}}) = (\hat{\mathbf{w}})^\top\mathbf{x} = \left[\mathbf{t}^\top X(X^\top X + \lambda N\mathbf{I})^{-1}\right]\mathbf{x}$$

Since X is $N \times (d+1)$, the matrix $X^\top X$ is $(d+1) \times (d+1)$

- The "model size" does not grow with data size (a **parametric** model)
- $X^\top X + \lambda N\mathbf{I}$ always has an inverse, for all $\lambda > 0$

5. Setting $\nabla E_\lambda(\mathbf{w}) = \mathbf{0}$, we obtain the (regularized) normal equations:

$$X^\top(\mathbf{t} - X\mathbf{w}) = N\lambda\mathbf{w}$$

with solution $\hat{\mathbf{w}} = (X^\top X + \lambda N\mathbf{I})^{-1}X^\top\mathbf{t}$

$$\Longrightarrow y(\mathbf{x}; \hat{\mathbf{w}}) = (\hat{\mathbf{w}})^\top\mathbf{x} = \left[\mathbf{t}^\top X(X^\top X + \lambda N\mathbf{I})^{-1}\right]\mathbf{x}$$

Since X is $N \times (d+1)$, the matrix $X^\top X$ is $(d+1) \times (d+1)$

- The "model size" does not grow with data size (a **parametric** model)
- $X^\top X + \lambda N\mathbf{I}$ always has an inverse, for all $\lambda > 0$

Reminder of the ridge regression framework

5. Setting $\nabla E_\lambda(\mathbf{w}) = \mathbf{0}$, we obtain the (regularized) normal equations:

$$X^\top(\mathbf{t} - X\mathbf{w}) = N\lambda\mathbf{w}$$

with solution $\hat{\mathbf{w}} = (X^\top X + \lambda N\mathbf{I})^{-1}X^\top\mathbf{t}$

$$\Longrightarrow y(\mathbf{x}; \hat{\mathbf{w}}) = (\hat{\mathbf{w}})^\top\mathbf{x} = \left[\mathbf{t}^\top X(X^\top X + \lambda N\mathbf{I})^{-1}\right]\mathbf{x}$$

Since X is $N \times (d+1)$, the matrix $X^\top X$ is $(d+1) \times (d+1)$

- The "model size" does not grow with data size (a **parametric** model)
- $X^\top X + \lambda N\mathbf{I}$ always has an inverse, for all $\lambda > 0$

Reminder of the ridge regression framework

5 Setting $\nabla E_\lambda(\mathbf{w}) = \mathbf{0}$, we obtain the (regularized) normal equations:

$$X^\top(\mathbf{t} - X\mathbf{w}) = N\lambda\mathbf{w}$$

with solution $\hat{\mathbf{w}} = (X^\top X + \lambda N\mathbf{I})^{-1}X^\top\mathbf{t}$

$$\Longrightarrow y(\mathbf{x}; \hat{\mathbf{w}}) = (\hat{\mathbf{w}})^\top\mathbf{x} = \left[\mathbf{t}^\top X(X^\top X + \lambda N\mathbf{I})^{-1}\right]\mathbf{x}$$

Since X is $N \times (d+1)$, the matrix $X^\top X$ is $(d+1) \times (d+1)$

- The "model size" does not grow with data size (a **parametric** model)
- $X^\top X + \lambda N\mathbf{I}$ always has an inverse, for all $\lambda > 0$

Introduction to kernel functions

We extend the framework it by means of a **mapping function**:

$$\varphi : \mathcal{X} \to \mathcal{H}_k$$

where $\mathcal{X}$ is the input space, $\mathcal{H}_k$ is the RKHS generated by k and:

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

$$k(\mathbf{x}, \mathbf{x}') \mapsto \left\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \right\rangle_{\mathcal{H}_k}$$

$\langle \cdot, \cdot \rangle$ denotes inner product in $\mathcal{H}_k$ (a.k.a. the **feature space**)

The "*kernel trick*" consists in performing the mapping and the inner product simultaneously by using the associated k

Introduction to kernel functions

We extend the framework it by means of a **mapping function**:

$$\varphi : \mathcal{X} \to \mathcal{H}_k$$

where $\mathcal{X}$ is the input space, $\mathcal{H}_k$ is the RKHS generated by k and:

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

$$k(\mathbf{x}, \mathbf{x}') \mapsto \left\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \right\rangle_{\mathcal{H}_k}$$

$\langle \cdot, \cdot \rangle$ denotes inner product in $\mathcal{H}_k$ (a.k.a. the **feature space**)

The "*kernel trick*" consists in performing the mapping and the inner product simultaneously by using the associated k

First, our feature space must have the structure of a **Hilbert space**: A vector space endowed with an **inner product** whose associated **norm** defines a complete metric:

- Distances, lengths and angles are well-defined for the elements of the space
- Completeness means that all Cauchy sequences defined in the space converge to an element of the space (under the ip-norm)

David Hilbert (1862-1943)

An example of a Hilbert space

The ℓ_2 space of square-summable sequences

$$\ell_2 := \left\{ (a_n)_{n=1}^{\infty}, a_n \in \mathbb{R}, \sum_{n=1}^{\infty} a_n^2 < \infty \right\}$$

- This is a vector space with inner product $\langle a, b \rangle := \sum_{n=1}^{\infty} a_n b_n$
- Completeness comes from the fact that $\mathbb{R}$ is complete

Generating the inner product

1. Given a two-place symmetric function k, consider the space of functions $\varphi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$, as

$$\varphi(\mathbf{x}) := k(\mathbf{x}, \cdot)$$

2. Define the (soon-to-be) vector space:

$$\begin{aligned}\mathcal{H} &:= \mathrm{span}\left\{\varphi(\mathbf{x}) /\ \mathbf{x} \in \mathcal{X}\right\} \\ &= \left\{ f(\cdot) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) /\ N \in \mathbb{N}, \mathbf{x}_n \in \mathcal{X}, \alpha_n \in \mathbb{R} \right\} \\ &= \left\{ f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n} /\ N \in \mathbb{N}, \mathbf{x}_n \in \mathcal{X}, \alpha_n \in \mathbb{R} \right\}\end{aligned}$$

Generating the inner product

1. Given a two-place symmetric function k, consider the space of functions $\varphi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$, as

$$\varphi(\mathbf{x}) := k(\mathbf{x}, \cdot)$$

2. Define the (soon-to-be) vector space:

$$\begin{aligned}
\mathcal{H} \; &:= \operatorname{span}\left\{\varphi(\mathbf{x}) / \; \mathbf{x} \in \mathcal{X}\right\} \\
&= \left\{ f(\cdot) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) / \; N \in \mathbb{N}, \mathbf{x}_n \in \mathcal{X}, \alpha_n \in \mathbb{R} \right\} \\
&= \left\{ f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n} / \; N \in \mathbb{N}, \mathbf{x}_n \in \mathcal{X}, \alpha_n \in \mathbb{R} \right\}
\end{aligned}$$

Generating the inner product

3 Let $f, f' \in \mathcal{H}$, be given as $f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, f' = \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_m}$,
Define an **inner product** in $\mathcal{H}$ as:

$$\langle f, g \rangle_{\mathcal{H}} = \left\langle \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_n} \right\rangle_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_n \alpha'_m k(\mathbf{x}_n, \mathbf{x}'_m)$$

4 Note that $\langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}) = f(\mathbf{x})$ This is called the **reproducing property** of the kernel

5 This definition ensures the identity we need:

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}} = \langle k_{\mathbf{x}}, k_{\mathbf{x}'} \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$$

Generating the inner product

3 Let $f, f' \in \mathcal{H}$, be given as $f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, f' = \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_m}$,
Define an **inner product** in $\mathcal{H}$ as:

$$\langle f, g \rangle_{\mathcal{H}} = \left\langle \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_n} \right\rangle_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_n \alpha'_m k(\mathbf{x}_n, \mathbf{x}'_m)$$

4 Note that $\langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}) = f(\mathbf{x})$ This is called the **reproducing property** of the kernel

5 This definition ensures the identity we need:

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}} = \langle k_{\mathbf{x}}, k_{\mathbf{x}'} \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$$

Generating the inner product

3 Let $f, f' \in \mathcal{H}$, be given as $f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, f' = \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_m}$,
Define an **inner product** in $\mathcal{H}$ as:

$$\langle f, g \rangle_{\mathcal{H}} = \left\langle \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_n} \right\rangle_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_n \alpha'_m k(\mathbf{x}_n, \mathbf{x}'_m)$$

4 Note that $\langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}) = f(\mathbf{x})$ This is called the **reproducing property** of the kernel

5 This definition ensures the identity we need:

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}} = \langle k_{\mathbf{x}}, k_{\mathbf{x}'} \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$$

Generating the inner product

3 Let $f, f' \in \mathcal{H}$, be given as $f = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}$, $f' = \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_m}$,
Define an **inner product** in $\mathcal{H}$ as:

$$\langle f, g \rangle_{\mathcal{H}} = \left\langle \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n}, \sum_{m=1}^{M} \alpha'_m k_{\mathbf{x}'_n} \right\rangle_{\mathcal{H}} := \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_n \alpha'_m k(\mathbf{x}_n, \mathbf{x}'_m)$$

4 Note that $\langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}) = f(\mathbf{x})$ This is called the **reproducing property** of the kernel

5 This definition ensures the identity we need:

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}} = \langle k_{\mathbf{x}}, k_{\mathbf{x}'} \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$$

Definition (Reproducing Kernel Hilbert space)

A RKHS $\mathcal{H}$ is a Hilbert space of functions f defined on a set $\mathcal{X}$ for which all the evaluation functionals:

$$\forall \mathbf{x} \in \mathcal{X},\ \ E_{\mathbf{x}}(f) := f(\mathbf{x})$$

are continuous (evaluation functionals are linear operators)

The existence and uniqueness of a reproducing kernel is derived from the Riesz representation theorem

Example

Consider $\mathcal{X} = [0,1]$ and the kernel $k(s,t) = \min\{s,t\}$. The generated RKHS is the Sobolev space $\mathcal{H}_k$ with inner product $\langle f, g \rangle_{\mathcal{H}_K} = \int_0^1 f'(t)g'(t)\,dt$:

$$\mathcal{H}_k = \left\{ f \in L^2[0,1] \text{ absolutely continuous},\ f(0) = 0, f' \in L^2[0,1] \right\}$$

Definition (Reproducing Kernel Hilbert space)

A RKHS $\mathcal{H}$ is a Hilbert space of functions f defined on a set $\mathcal{X}$ for which all the evaluation functionals:

$$\forall \mathbf{x} \in \mathcal{X},\ \ E_{\mathbf{x}}(f) := f(\mathbf{x})$$

are continuous (evaluation functionals are linear operators)

The existence and uniqueness of a reproducing kernel is derived from the Riesz representation theorem

Example

Consider $\mathcal{X} = [0,1]$ and the kernel $k(s,t) = \min\{s,t\}$. The generated RKHS is the Sobolev space $\mathcal{H}_k$ with inner product $\langle f, g\rangle_{\mathcal{H}_K} = \int_0^1 f'(t)g'(t)\,dt$:

$$\mathcal{H}_k = \left\{ f \in L^2[0,1] \text{ absolutely continuous},\ f(0) = 0, f' \in L^2[0,1] \right\}$$

Definition

A symmetric function k is called a *positive semi-definite* kernel in $\mathcal{X}$ if for every $N \in \mathbb{N}$, and every choice $\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{X}$, the matrix $\mathbf{K} = (k_{ij})$, where $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is *positive semi-definite* (p.s.d.)

Theorem (Characterization)

k is a reproducing kernel and admits the existence of a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\mathcal{H}$ is a RKHS and $k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}}$ if and only if k is a p.s.d. symmetric kernel in $\mathcal{X}$

Definition

A symmetric function k is called a *positive semi-definite* kernel in $\mathcal{X}$ if for every $N \in \mathbb{N}$, and every choice $\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{X}$, the matrix $\mathbf{K} = (k_{ij})$, where $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is *positive semi-definite* (p.s.d.)

Theorem (Characterization)

k is a reproducing kernel and admits the existence of a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\mathcal{H}$ is a RKHS and $k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}}$ if and only if k is a p.s.d. symmetric kernel in $\mathcal{X}$

Why is k p.s.d.?

$$k(\mathbf{x},\mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}}$$

$\forall \mathbf{c} \in \mathbb{R}^N$,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{N} c_i \varphi(\mathbf{x}_i), \sum_{j=1}^{N} c_j \varphi(\mathbf{x}_j) \right\rangle_{\mathcal{H}} \geq 0$$

1. Which holds for all choices of $\varphi(\cdot)$
2. Generalizes inner product (think about the case $\varphi(\mathbf{x}) = \mathbf{x}$)

Learning in RKHSs

The key for learning in RKHSs is the **regularization framework**:

- Consider again a **learning data** sample

$$D = \{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}, \ \mathbf{x}_n \in \mathcal{X}, t_n \in \mathbb{R}$$

- **Goal**: learn a function $y : \mathcal{X} \to \mathbb{R}$ from D and a set of possible solutions (models, hypotheses) $\mathcal{H} = \{y | y : \mathcal{X} \to \mathbb{R}\}$
- Assume a **loss** function $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ that measures the divergence between a model's predictions and the targets:

$$(t, y(\mathbf{x})) \mapsto L(t, y(\mathbf{x}))$$

$$\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} L(t_n, y(\mathbf{x}_n)) + \lambda \|y\|^2_{\mathcal{H}_k} \right\}, \lambda > 0$$

Learning in RKHSs

Theorem (Representer Theorem)

Consider $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ a convex loss function, an observed data sample $D = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$, with $\mathbf{x}_n \in \mathcal{X}, t_n \in \mathbb{R}$, and $\mathcal{H}_k$ a RKHS of functions $y : \mathcal{X} \to \mathbb{R}$ with reproducing kernel k. Then, for all $\lambda > 0$,

1. *There exists a unique solution $\hat{y}_\lambda$ to the problem:*

$$\hat{y}_\lambda := \arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} L(t_n, y(\mathbf{x}_n)) + \lambda \|y\|^2_{\mathcal{H}_k} \right\}$$

2. *There exist $\alpha_1, \dots, \alpha_N \in \mathbb{R}$ such that*

$$\hat{y}_\lambda(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}$$

Theorem (Representer Theorem)

Consider $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ a convex loss function, an observed data sample $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$, with $\mathbf{x}_n \in \mathcal{X}, t_n \in \mathbb{R}$, and $\mathcal{H}_k$ a RKHS of functions $y : \mathcal{X} \to \mathbb{R}$ with reproducing kernel k. Then, for all $\lambda > 0$,

1. *There exists a unique solution $\hat{y}_\lambda$ to the problem:*

$$\hat{y}_\lambda := \arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} L(t_n, y(\mathbf{x}_n)) + \lambda \|y\|^2_{\mathcal{H}_k} \right\}$$

2. *There exist $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ such that*

$$\hat{y}_\lambda(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}$$

Learning in RKHSs

Theorem (Representer Theorem)

Consider $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ *a convex loss function, an observed data sample* $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$*, with* $\mathbf{x}_n \in \mathcal{X}, t_n \in \mathbb{R}$*, and* $\mathcal{H}_k$ *a RKHS of functions* $y : \mathcal{X} \to \mathbb{R}$ *with reproducing kernel* k*. Then, for all* $\lambda > 0$*,*

1. *There exists a unique solution* $\hat{y_\lambda}$ *to the problem:*

$$\hat{y_\lambda} := \arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} L(t_n, y(\mathbf{x}_n)) + \lambda \|y\|^2_{\mathcal{H}_k} \right\}$$

2. *There exist* $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ *such that*

$$\hat{y_\lambda}(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}$$

Kernel ridge regression

We consider the choice $L(t, y(\mathbf{x})) = (t - y(\mathbf{x}))^2$

$$\arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 + \lambda \|y\|^2_{\mathcal{H}_k} \right\}, \lambda > 0$$

1. Given D, λ, the **representer theorem** ensures a solution:

$$\hat{y}_\lambda = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n} \in \mathcal{H}_k, \quad \text{or } \hat{y}_\lambda(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x})$$

2. The **parameters** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\top \in \mathbb{R}^N$ are obtained as:

$$\hat{\alpha} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}, \lambda > 0$$

where (as introduced earlier) $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]$

Kernel ridge regression

We consider the choice $L(t, y(\mathbf{x})) = (t - y(\mathbf{x}))^2$

$$\arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 + \lambda \|y\|^2_{\mathcal{H}_k} \right\}, \lambda > 0$$

1. Given D, λ, the **representer theorem** ensures a solution:

$$\hat{y}_\lambda = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n} \in \mathcal{H}_k, \quad \text{or } \hat{y}_\lambda(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x})$$

2. The **parameters** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\top \in \mathbb{R}^N$ are obtained as:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}, \lambda > 0$$

where (as introduced earlier) $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]$

Kernel ridge regression

We consider the choice $L(t, y(\mathbf{x})) = (t - y(\mathbf{x}))^2$

$$\arg\min_{y \in \mathcal{H}_k} \left\{ \frac{1}{N} \sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 + \lambda \|y\|^2_{\mathcal{H}_k} \right\}, \lambda > 0$$

1. Given D, λ, the **representer theorem** ensures a solution:

$$\hat{y_\lambda} = \sum_{n=1}^{N} \alpha_n k_{\mathbf{x}_n} \in \mathcal{H}_k, \quad \text{or } \hat{y_\lambda}(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x})$$

2. The **parameters** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\top \in \mathbb{R}^N$ are obtained as:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}, \lambda > 0$$

where (as introduced earlier) $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]$

$$\|y\|^2_{\mathcal{H}} = \langle y, y \rangle_{\mathcal{H}_k} = \left\langle \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot), \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) \right\rangle_{\mathcal{H}_k}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha} \geq 0$$

$$\sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 = \sum_{n=1}^{N} \left(t_n - \sum_{m=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}_m) \right)^2$$

$$= \sum_{n=1}^{N} (t_n - (\mathbf{K}\boldsymbol{\alpha})_n)^2 = \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2$$

- The solution parameter vector is $\hat{\alpha} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \mathbf{t}$
- We do ridge regression based only on $\mathbf{K}$ (and throw away X)
- "Model size" grows with data size (a **non-parametric** model)

Learning in RKHSs

$$\|y\|^2_{\mathcal{H}} = \langle y, y \rangle_{\mathcal{H}_k} = \left\langle \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot), \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) \right\rangle_{\mathcal{H}_k}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha} \geq 0$$

$$\sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 = \sum_{n=1}^{N} \left(t_n - \sum_{m=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}_m) \right)^2$$

$$= \sum_{n=1}^{N} (t_n - (\mathbf{K}\boldsymbol{\alpha})_n)^2 = \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2$$

- The solution parameter vector is $\hat{\alpha} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \mathbf{t}$
- We do ridge regression based only on $\mathbf{K}$ (and throw away X)
- "Model size" grows with data size (a **non-parametric** model)

$$\|y\|^2_{\mathcal{H}} = \langle y, y\rangle_{\mathcal{H}_k} = \left\langle \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot), \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) \right\rangle_{\mathcal{H}_k}$$

$$= \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha} \geq 0$$

$$\sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 = \sum_{n=1}^{N} \left(t_n - \sum_{m=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}_m) \right)^2$$

$$= \sum_{n=1}^{N} (t_n - (\mathbf{K}\boldsymbol{\alpha})_n)^2 = \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2$$

- The solution parameter vector is $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \mathbf{t}$
- We do ridge regression based only on $\mathbf{K}$ (and throw away X)
- "Model size" grows with data size (a **non-parametric** model)

$$\|y\|^2_{\mathcal{H}} = \langle y, y \rangle_{\mathcal{H}_k} = \left\langle \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot), \sum_{n=1}^{N} \alpha_n k(\mathbf{x}_n, \cdot) \right\rangle_{\mathcal{H}_k}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha} \geq 0$$

$$\sum_{n=1}^{N} (t_n - y(\mathbf{x}_n))^2 = \sum_{n=1}^{N} \left(t_n - \sum_{m=1}^{N} \alpha_n k(\mathbf{x}_n, \mathbf{x}_m) \right)^2$$

$$= \sum_{n=1}^{N} (t_n - (\mathbf{K}\boldsymbol{\alpha})_n)^2 = \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2$$

- The solution parameter vector is $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \mathbf{t}$
- We do ridge regression based only on $\mathbf{K}$ (and throw away X)
- "Model size" grows with data size (a **non-parametric** model)

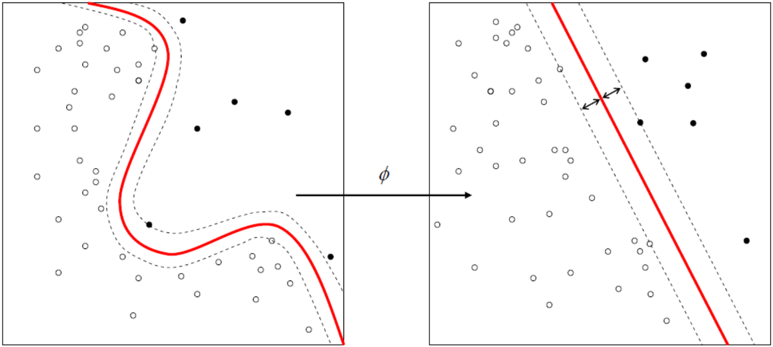
Learning in RKHSs

$$
\begin{aligned}
\|y\|^2_{\mathcal{H}} = \langle y, y\rangle_{\mathcal{H}_k} &= \left\langle \sum_{n=1}^N \alpha_n k(\mathbf{x}_n, \cdot), \sum_{n=1}^N \alpha_n k(\mathbf{x}_n, \cdot) \right\rangle_{\mathcal{H}_k} \\
&= \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha} \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\sum_{n=1}^N (t_n - y(\mathbf{x}_n))^2 &= \sum_{n=1}^N \left(t_n - \sum_{m=1}^N \alpha_n k(\mathbf{x}_n, \mathbf{x}_m) \right)^2 \\
&= \sum_{n=1}^N (t_n - (\mathbf{K}\boldsymbol{\alpha})_n)^2 = \|\mathbf{t} - \mathbf{K}\boldsymbol{\alpha}\|^2
\end{aligned}
$$

- The solution parameter vector is $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1}\mathbf{t}$
- We do ridge regression based only on $\mathbf{K}$ (and throw away X)
- "Model size" grows with data size (a **non-parametric** model)

Kernel methods



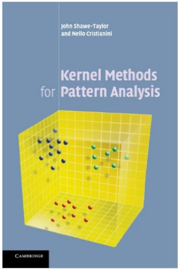
Machine learning: an SVM in action (from the Wikipedia)

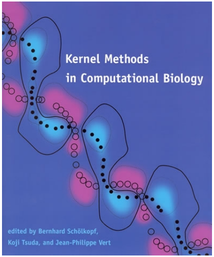
These techniques yield models that are:

- **non-linear** (in the input space $\mathcal{X}$)
- **linear** (in the feature space $\mathcal{H}_k$)

Many (classical and new) learning algorithms can be **kernelized**

1. They require solving a problem where the data appear in the form of pairwise **inner products** (or pairwise **Euclidean distances**)
2. The solution is expressed as a linear combination of the kernel function centered at *some* of the data: **sparsity**
3. Examples include SVMs, ridge regression, perceptrons, FDA, PLS [supervised], and PCA, k-means, Parzen Windows [unsupervised]





Introduction to kernel functions

Kernels inherit important **properties** from inner products:

1. Symmetry

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$$

2. Cauchy-Schwarz inequality

$$|k(\mathbf{x}, \mathbf{x}')| \leq \sqrt{k(\mathbf{x}, \mathbf{x})} \cdot \sqrt{k(\mathbf{x}', \mathbf{x}')}$$

3. Definiteness

$$k(\mathbf{x}, \mathbf{x}) = \|\varphi(\mathbf{x})\|^2 \geq 0$$

4. Closure properties:
   - Sums and products, direct sums and tensor products
   - Multiplication by positive coefficients
   - Limits of point-wise convergent sequences
   - Composition with certain analytic functions

Example

1. If k is a kernel and p is a polynomial of degree q with non-negative coefficients, then the function

$$k_p(\mathbf{x}, \mathbf{x}') := p(k(\mathbf{x}, \mathbf{x}'))$$

is also a kernel.

2. The special case where k is linear and

$$p(z) = (z + c)^q, \; c \geq 0, q \in \mathbb{N}$$

leads to the so-called **polynomial kernel**:

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^q, \; \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

What is the underlying mapping φ here? φ leads into the space spanned by all products of at most q dimensions of $\mathbb{R}^d$

Introduction to kernel functions

Example

1. If k is a kernel and p is a polynomial of degree q with non-negative coefficients, then the function

$$k_p(\mathbf{x}, \mathbf{x}') := p(k(\mathbf{x}, \mathbf{x}'))$$

   is also a kernel.

2. The special case where k is linear and

$$p(z) = (z + c)^q, \ \ c \geq 0, q \in \mathbb{N}$$

   leads to the so-called **polynomial kernel**:

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^q, \ \ \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

   What is the underlying mapping φ here? φ leads into the space spanned by all products of at most q dimensions of $\mathbb{R}^d$

Example

1. If k is a kernel and p is a polynomial of degree q with non-negative coefficients, then the function

$$k_p(\mathbf{x}, \mathbf{x}') := p(k(\mathbf{x}, \mathbf{x}'))$$

   is also a kernel.

2. The special case where k is linear and

$$p(z) = (z + c)^q, \;\; c \geq 0, q \in \mathbb{N}$$

   leads to the so-called **polynomial kernel**:

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^q, \;\; \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

   What is the underlying mapping φ here? φ leads into the space spanned by all products of at most q dimensions of $\mathbb{R}^d$

Example

1. If k is a kernel and p is a polynomial of degree q with non-negative coefficients, then the function

$$k_p(\mathbf{x}, \mathbf{x}') := p(k(\mathbf{x}, \mathbf{x}'))$$

is also a kernel.

2. The special case where k is linear and

$$p(z) = (z + c)^q, \ \ c \geq 0, q \in \mathbb{N}$$

leads to the so-called **polynomial kernel**:

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^q, \ \ \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

What is the underlying mapping φ here? φ leads into the space spanned by all products of at most q dimensions of $\mathbb{R}^d$

Definition (Radial kernels)

We say that a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is **radial** if it has the form

$$k(\mathbf{x}, \mathbf{x}') = t(\|\mathbf{x} - \mathbf{x}'\|),$$

where $t : [0, \infty) \to [0, \infty)$ is a differentiable function
Radial kernels fulfill $k(\mathbf{x}, \mathbf{x}) = t(0)$

The Gaussian kernel

Consider the function $t(z) = \exp(-\gamma z^2), \gamma > 0$. The resulting radial kernel is known as the **Gaussian RBF kernel**:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$$

The kernel matrix for this kernel has always full rank for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_N$, so the feature map φ has infinite dimension

Introduction to kernel functions

Definition (Radial kernels)

We say that a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is **radial** if it has the form

$$k(\mathbf{x}, \mathbf{x}') = t(\|\mathbf{x} - \mathbf{x}'\|),$$

where $t : [0, \infty) \to [0, \infty)$ is a differentiable function
Radial kernels fulfill $k(\mathbf{x}, \mathbf{x}) = t(0)$

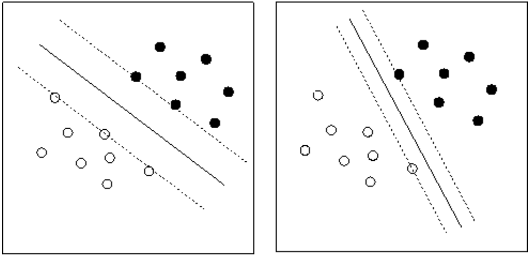
The Gaussian kernel

Consider the function $t(z) = \exp(-\gamma z^2), \gamma > 0$. The resulting radial kernel is known as the **Gaussian RBF kernel**:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

The kernel matrix for this kernel has always full rank for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_N$, so the feature map φ has infinite dimension

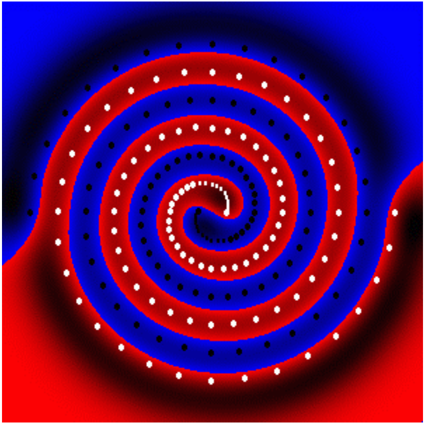
Support Vector Machines


(a) Larger margin (b) Smaller margin

Which solution is more likely to lead to better **generalization**?

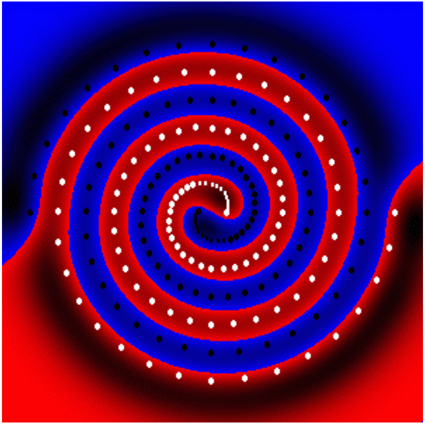
Support Vector Machines

A **Support Vector Machine** (SVM) is a kernelized two-class classifier (a hyperplane) that aims at leaving the maximum possible margin of separation between the classes, with allowance for margin violations via a complexity parameter



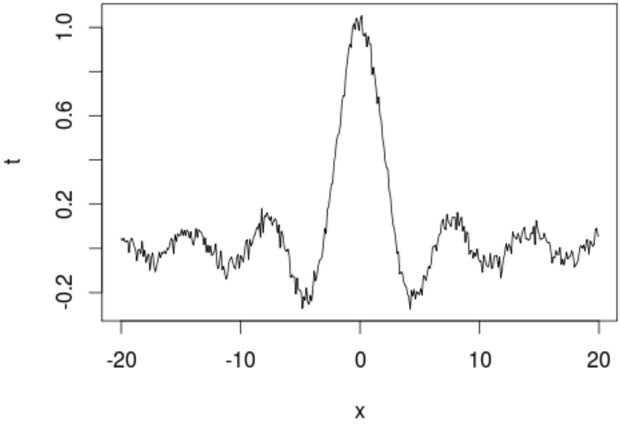
This is a hyperplane! (in some RKHS) –from `www.kernel-methods.net`

Introduction to kernel functions



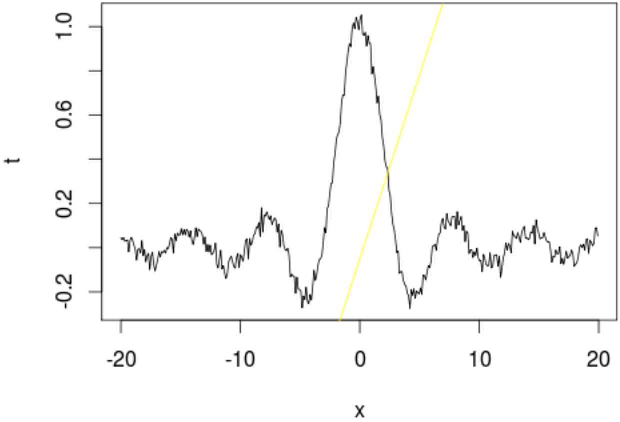
This is a hyperplane! (in some RKHS) –from `www.kernel-methods.net`

Kernel ridge regression in action



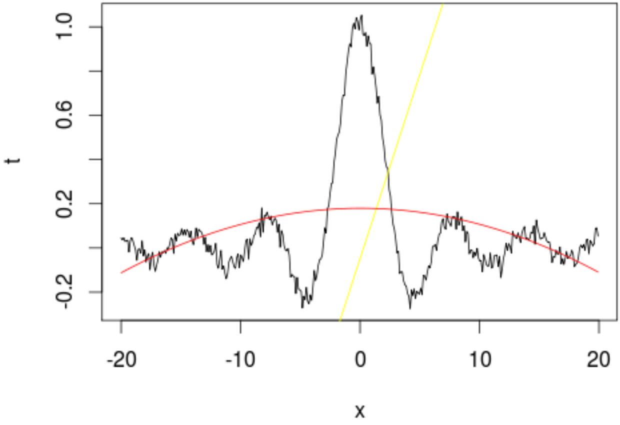
$$\frac{\sin(x)}{x} + \mathcal{N}(x, 0.03^2), \;\; x \in [-20, 20]$$

Kernel ridge regression in action



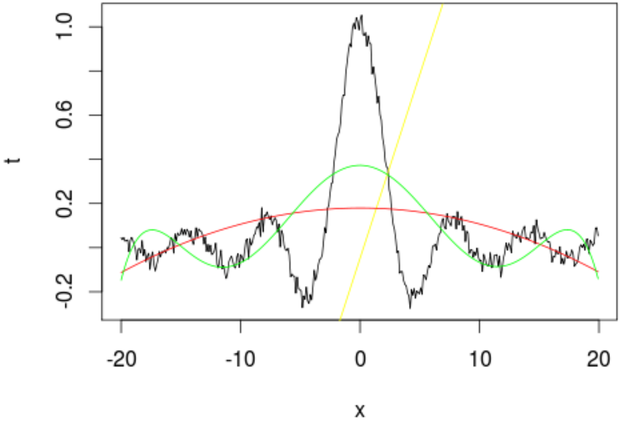
fitting a polynomial of degree 1 ...

Kernel ridge regression in action



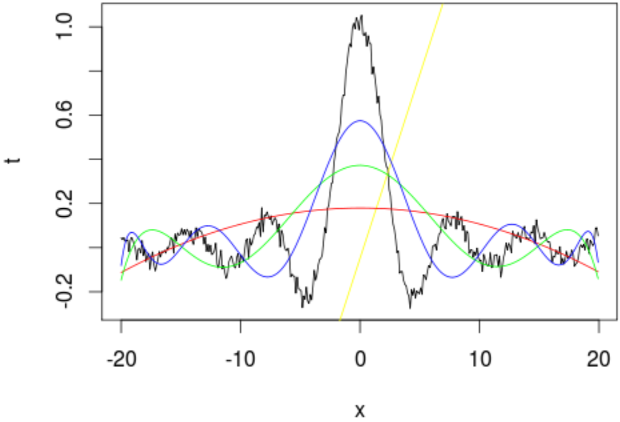
fitting a polynomial of degree 2 ...

Kernel ridge regression in action



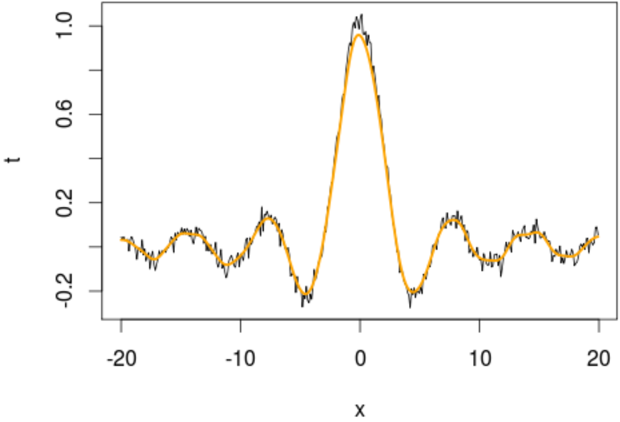
fitting a polynomial of degree 6 ...

Kernel ridge regression in action



fitting a polynomial of degree 11 ...

Kernel ridge regression in action



Kernel ridge regression with RBF kernel ($\gamma = 1, \lambda = 1$)

PCA vs. Kernel PCA

$$\sum_{n=1}^{N} \mathbf{x}_n = \mathbf{0}$$

$$\sum_{n=1}^{N} \varphi(\mathbf{x}_n) = \mathbf{0}$$

$$C = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top$$

$$\mathbf{K} = \frac{1}{N}\sum_{n=1}^{N} \varphi(\mathbf{x}_n)\varphi(\mathbf{x}_n)^\top$$

$$C\mathbf{v} = \lambda \mathbf{v}$$

$$\mathbf{K}\mathbf{v} = \lambda \mathbf{v}$$

$$\mathbf{v} = \sum_{n=1}^{N} \alpha_n \varphi(\mathbf{x}_n)$$

$$N\lambda \mathbf{K}\boldsymbol{\alpha} = \mathbf{K}\mathbf{K}\boldsymbol{\alpha}$$

$$\mathbf{K}\boldsymbol{\alpha} = N\lambda\boldsymbol{\alpha}$$

Kernel methods in action

Definition (The Spectrum kernel)

- Let Σ be a finite alphabet; for $p \geq 1$ and a sequence $\mathbf{x}$, a p-gram is any block of p adjacent characters from Σ in $\mathbf{x}$
- The p-spectrum of a sequence $\mathbf{x}$ is the vector of counters of all p-grams that $\mathbf{x}$ contains

Define $k(\mathbf{x}, \mathbf{x}') = \sum_{s \in \Sigma^p} |s \in \mathbf{x}| \cdot |s \in \mathbf{x}'|$

Example

Proteins **GGTGTCA** with alphabet $\Sigma = \{\mathbf{C}, \mathbf{A}, \mathbf{G}, \mathbf{T}\}$ (the four nucleobases that make up the DNA) and $p = 2$:

GA	GC	GT	GG	CA	CC	...
0	0	2	1	1	0	...

Although the number of distinct p-grams in a text $\mathbf{x}$ is at most $|\Sigma|^p$, this kernel can be computed in $O(|\mathbf{x}| + |\mathbf{x}'|)$ time and memory, for all p

Kernel methods in action

Definition (The Spectrum kernel)

- Let Σ be a finite alphabet; for $p \geq 1$ and a sequence $\mathbf{x}$, a p-gram is any block of p adjacent characters from Σ in $\mathbf{x}$
- The p-spectrum of a sequence $\mathbf{x}$ is the vector of counters of all p-grams that $\mathbf{x}$ contains

Define $k(\mathbf{x}, \mathbf{x}') = \sum_{s \in \Sigma^p} |s \in \mathbf{x}| \cdot |s \in \mathbf{x}'|$

Example

Proteins **GGTGTCA** with alphabet $\Sigma = \{\mathbf{C}, \mathbf{A}, \mathbf{G}, \mathbf{T}\}$ (the four nucleobases that make up the DNA) and $p = 2$:

GA	GC	GT	GG	CA	CC	...
0	0	2	1	1	0	...

Although the number of distinct p-grams in a text $\mathbf{x}$ is at most $|\Sigma|^p$, this kernel can be computed in $O(|\mathbf{x}| + |\mathbf{x}'|)$ time and memory, for all p

Kernel methods in action

Definition (The Spectrum kernel)

- Let Σ be a finite alphabet; for $p \geq 1$ and a sequence $\mathbf{x}$, a p-gram is any block of p adjacent characters from Σ in $\mathbf{x}$
- The p-spectrum of a sequence $\mathbf{x}$ is the vector of counters of all p-grams that $\mathbf{x}$ contains

Define $k(\mathbf{x}, \mathbf{x}') = \sum_{s \in \Sigma^p} |s \in \mathbf{x}| \cdot |s \in \mathbf{x}'|$

Example

Proteins **GGTGTCA** with alphabet $\Sigma = \{\mathbf{C}, \mathbf{A}, \mathbf{G}, \mathbf{T}\}$ (the four nucleobases that make up the DNA) and $p = 2$:

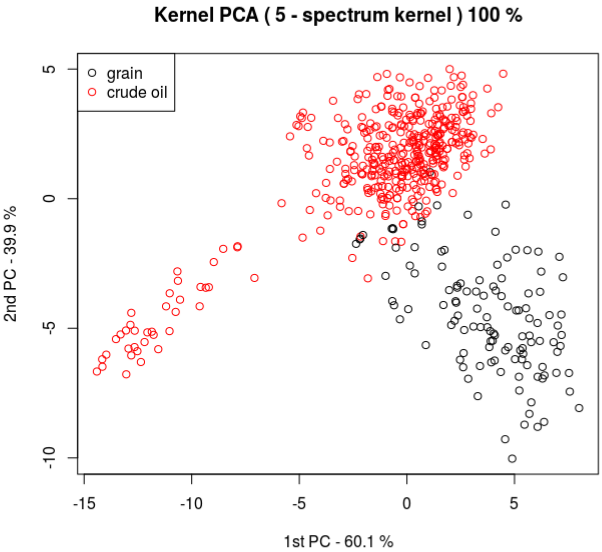
GA	GC	GT	GG	CA	CC	...
0	0	2	1	1	0	...

Although the number of distinct p-grams in a text $\mathbf{x}$ is at most $|\Sigma|^p$, this kernel can be computed in $O(|\mathbf{x}| + |\mathbf{x}'|)$ time and memory, for all p

Text visualization: the Reuters news articles dataset

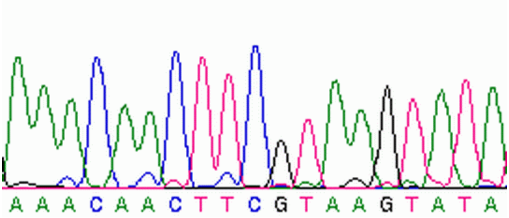
- **coffee**: "mexico has temporarily suspended overseas coffee sales due to falling prices triggered by the failure of the international coffee organisation (ico) meeting to agree a quota system at its latest meeting, the official notimex news agency said. "we're just waiting a while for prices to improve," an unidentified mexican trader told the agency. mexico has already sold 80 pct of ..."
- **crude oil**: "u.s. department of energy secretary john herrington said he was "optimistic" about the chances of providing a more generous depletion allowance for oil and gas producers, but added that the plan faces strong opposition from some members of the reagan administration. herrington, speaking to houston oil executives at a breakfast meeting, said administration debate over his plan for a 27.5 pct annual depletion allowance was "heavy and strong" largely because of some fears that ... "

Kernel methods in action



Classification of DNA sequences with SVMs

- A *promoter* is a region of DNA that initiates or facilitates transcription of a particular gene
- The dataset consists of 106 DNA sequences described by 57 categorical variables, represented as the nucleotide at each position: [A]denine, [C]ytosine, [G]uanine, [T]hymine
- The response is a binary class: "+" for a promoter gene and "−" for a non-promoter gene



Kernel methods in action

Classification of DNA sequences with SVMs

The similarity between two multivariate categorical vectors is the fraction of the number of matching values.

Definition (Overlap/Dirac kernel)

$$k_0(\mathbf{x}, \mathbf{x}') = \frac{1}{d} \sum_{i=1}^{d} \mathbb{I}_{\{x_i = x_i'\}}$$

Another kernel that can be used is the RBF kernel:

Definition (Gaussian Radial Basis Function kernel)

$$k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma ||\mathbf{x} - \mathbf{x}'||^2\right) \ , \gamma > 0$$

In order to use this kernel, categorical variables with m modalities are coded using a binary expansion representation.

Classification of DNA sequences with SVMs

The similarity between two multivariate categorical vectors is the fraction of the number of matching values.

Definition (Overlap/Dirac kernel)

$$k_0(\mathbf{x}, \mathbf{x}') = \frac{1}{d} \sum_{i=1}^{d} \mathbb{I}_{\{x_i = x_i'\}}$$

Another kernel that can be used is the RBF kernel:

Definition (Gaussian Radial Basis Function kernel)

$$k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma ||\mathbf{x} - \mathbf{x}'||^2\right) \ , \gamma > 0$$

In order to use this kernel, categorical variables with m modalities are coded using a binary expansion representation.

Classification of DNA sequences with SVMs

The similarity between two multivariate categorical vectors is the fraction of the number of matching values.

Definition (Overlap/Dirac kernel)

$$k_0(\mathbf{x}, \mathbf{x}') = \frac{1}{d} \sum_{i=1}^{d} \mathbb{I}_{\{x_i = x'_i\}}$$

Another kernel that can be used is the RBF kernel:

Definition (Gaussian Radial Basis Function kernel)

$$k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma ||\mathbf{x} - \mathbf{x}'||^2\right) \ , \gamma > 0$$

In order to use this kernel, categorical variables with m modalities are coded using a binary expansion representation.

Definition (Univariate kernel $k_1^{(U)}$)

$$k_1^{(U)}(x_i, x_i') = \begin{cases} h_\alpha(P_Z(x_i)) & \text{if } x_i = x_i' \\ 0 & \text{if } x_i \neq x_i' \end{cases}$$

where

$$h_\alpha(z) = (1 - z^\alpha)^{1/\alpha}, \ \alpha > 0$$

Definition (Multivariate kernel k_1)

$$k_1(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{\gamma}{d}\sum_{i=1}^{d} k_1^{(U)}(x_i, x_i')\right), \ \gamma > 0$$

Theorem

The kernel matrices generated by k_1 are p.s.d.

Kernel methods in action

Definition (Univariate kernel $k_1^{(U)}$)

$$k_1^{(U)}(x_i, x_i') = \begin{cases} h_\alpha(P_Z(x_i)) & \text{if } x_i = x_i' \\ 0 & \text{if } x_i \neq x_i' \end{cases}$$

where

$$h_\alpha(z) = (1 - z^\alpha)^{1/\alpha}, \;\; \alpha > 0$$

Definition (Multivariate kernel k_1)

$$k_1(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{\gamma}{d} \sum_{i=1}^{d} k_1^{(U)}(x_i, x_i') \right), \;\; \gamma > 0$$

Theorem

The kernel matrices generated by k_1 are p.s.d.

Kernel methods in action

Definition (Univariate kernel $k_1^{(U)}$)

$$k_1^{(U)}(x_i, x_i') = \begin{cases} h_\alpha(P_Z(x_i)) & \text{if } x_i = x_i' \\ 0 & \text{if } x_i \neq x_i' \end{cases}$$

where

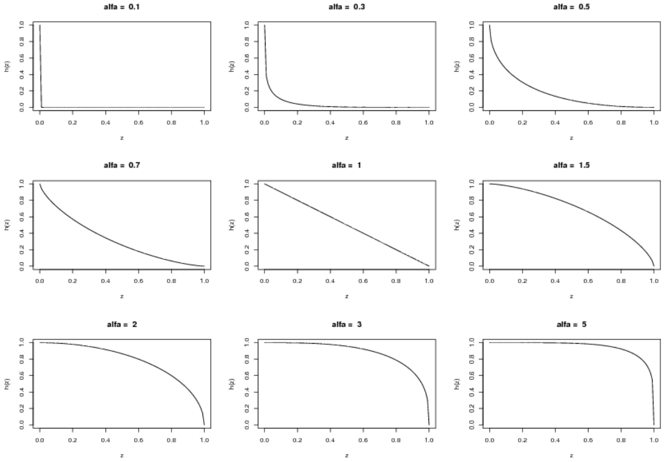
$$h_\alpha(z) = (1 - z^\alpha)^{1/\alpha}, \ \alpha > 0$$

Definition (Multivariate kernel k_1)

$$k_1(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{\gamma}{d} \sum_{i=1}^{d} k_1^{(U)}(x_i, x_i')\right), \ \gamma > 0$$

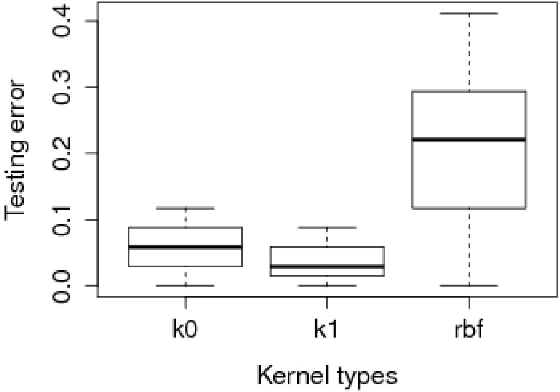
Theorem

The kernel matrices generated by k_1 are p.s.d.



The family of inverting functions $h_\alpha(z)$ for different values of α

Kernel methods in action



Test error distributions on the PromoterGene problem

(*joint work with M. Villegas*)

A personal view

The Hype

‘‘They used to call it Statistics, now it is called Machine Learning!’’ (*anonymous*)

The Truth

‘‘The ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics’’ (*Michael I. Jordan*)

The Hype

``They used to call it Statistics, now it is called Machine Learning!'' (*anonymous*)

The Truth

``The ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics'' (*Michael I. Jordan*)

The pseudo-Truth

''What is the difference between statistics, machine learning, AI and data mining?''

1. If there are up to 3 variables, it is statistics
2. If overfitting is an issue, it is machine learning
3. If you produce a promo video of it, it is AI
4. If you don't know what overfitting is, it is data mining

The pseudo-Truth

''What is the difference between statistics, machine learning, AI and data mining?''

1. If there are up to 3 variables, it is statistics
2. If overfitting is an issue, it is machine learning
3. If you produce a promo video of it, it is AI
4. If you don't know what overfitting is, it is data mining

The pseudo-Truth

''What is the difference between statistics, machine learning, AI and data mining?''

1. If there are up to 3 variables, it is statistics
2. If overfitting is an issue, it is machine learning
3. If you produce a promo video of it, it is AI
4. If you don't know what overfitting is, it is data mining

The pseudo-Truth

''What is the difference between statistics, machine learning, AI and data mining?''

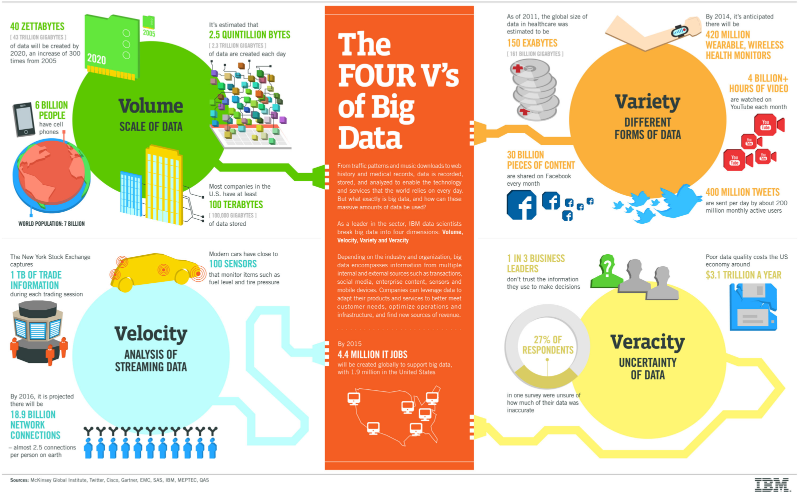
1. If there are up to 3 variables, it is statistics
2. If overfitting is an issue, it is machine learning
3. If you produce a promo video of it, it is AI
4. If you don't know what overfitting is, it is data mining

The pseudo-Truth

``What is the difference between statistics, machine learning, AI and data mining?''

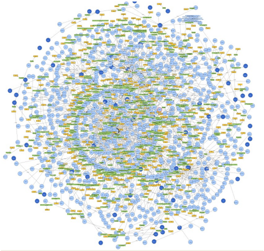
1. If there are up to 3 variables, it is statistics
2. If overfitting is an issue, it is machine learning
3. If you produce a promo video of it, it is AI
4. If you don't know what overfitting is, it is data mining

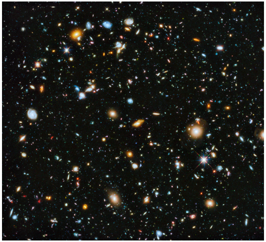
Machine learning, statistics and all that jazz



www.ibmbigdatahub.com/infographic/four-vs-big-data

Machine learning, statistics and all that jazz





Modern sources of data

Machine learning, statistics and all that jazz

$$\Rightarrow \begin{cases} \text{“\textit{Panther}”} & (0.97) \\ \text{“\textit{Drawing}”} & (0.86) \end{cases}$$

THE END

Thank
You
Gracias
Merci
Arigato
bolzïn
Mehrbani
Shukuria
Tashakkur
Grazie
Biyan
Juspaxar
Dankscheen
Efcharisto
Yaqhanyelay
suksama
Shukria
Komapsumnida
gozaimashita
atu
Tingki
Paldies
Maake
Ekhmet
Maketai
hui
Sanco
Wabeeja
Medawagse
Mersi
unalchéesh
Hatur
aniha
Denkauja
Aguyje
Spassibo
Fakaaue
Spasibo
Nenachalhya
Baiika
Yuspagaràtam
Minmonchar
Atto
Gaejtho
Maiteka
ekoju
Tavtapuch
Sikomo
Gui
lah
Merastawhy
Dhanyabaad
Chaltu
nuhun
Snachalhuya
Print...