2. Kernels and Regularisation

COMP0078: Supervised Learning

Carlo Ciliberto

(Slides thanks to Mark Herbster)

October 12, 2023

University College London
Department of Computer Science

Today's Plan

Overview

- Inner product space review
- Convexity review
- Ridge Regression
- Basis Functions (Explicit Feature Maps)
- Kernel Functions (Implicit Feature Maps)

Overview

- We show how a linear method such as least squares may be **lifted** to a (potentially) higher dimensional space to provide a nonlinear regression.
- We consider both **explicit** and **implicit** feature maps
- A feature map is simply a function that maps the “inputs” into a new space.
- Thus the original method is now nonlinear in original “inputs” but linear in the “mapped inputs”
- Explicit feature maps are often known as the *Method of Basis Functions*
- Implicit feature maps are often known as the *(reproducing) “Kernel Trick”*

Review: Inner Product Space

Vector Space

Vector space over the reals

The triple $(X, +, *)$ defines a *vector space* where X is a set and $+ : X \times X \to X$ is vector addition and $* : \mathbb{R} \times X \to X$ is scalar multiplication with the abbreviation $a\mathbf{x} := a * \mathbf{x}$. For which the following properties hold.

1. $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$
2. $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$
3. There exists $\mathbf{0} \in X$ such that for all $\mathbf{x} \in X$ then $\mathbf{x} + \mathbf{0} = \mathbf{x}$
4. $\gamma(\mathbf{x} + \mathbf{y}) = \gamma\mathbf{x} + \gamma\mathbf{y}$
5. $(\gamma + \mu)\mathbf{x} = \gamma\mathbf{x} + \mu\mathbf{x}$
6. $\gamma(\mu\mathbf{x}) = (\gamma\mu)\mathbf{x}$
7. $0\mathbf{x} = \mathbf{0}$ and $1\mathbf{x} = \mathbf{x}$

Notes

1. $\gamma + \mu$ and $\gamma\mu$ are the usual scalar addition and multiplication over $\mathbb{R}$
2. A vector space can be defined over other *fields* than the reals for example arithmetic mod-2. I.e $X = \{\mathbf{0}, \mathbf{1}\}$ and the scalar set is just $\{0, 1\}$.

Normed Space

A function $\|\cdot\| : X \to \mathbb{R}$ is a *norm* on a vector space if

1. $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$
2. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$
3. $\|\gamma\mathbf{x}\| = |\gamma|\|\mathbf{x}\|$

A norm defines a *metric* (distance) between vectors in the space via $d(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|$. Check that the triangle inequality holds $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Examples

1. $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i^p|)^{\frac{1}{p}}$ where $\mathbf{x} \in \mathbb{R}^n$ and $p \in [1, \infty)$.
2. $\|\mathbf{x}\|_M := \sqrt{\mathbf{x}^\top M\mathbf{x}}$ where $\mathbf{x} \in \mathbb{R}^n$ and M is an $n \times n$ symmetric positive definite matrix.

Definition

A real-valued symmetric $n \times n$ square matrix M is *positive definite* iff $\mathbf{x}^\top M\mathbf{x} > 0 \ \ (\forall\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\})$.

Normed Space – Intuitions

A norm generalises the notion of euclidean magnitude $\|\cdot\|_2$. The p-norm plays a particularly important role in machine learning where kernel methods may be seen as a generalisation of the case $p = 2$. The case $p = 1$ is particularly important when learning sparse predictors.

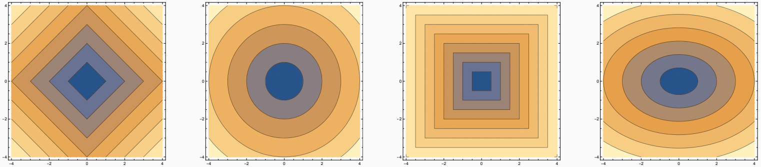


Figure 1: Level sets of $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$, $\|\cdot\|_{\begin{pmatrix}1 & 0\\ 0 & 3\end{pmatrix}}$

Where $\|\mathbf{x}\|_\infty := \lim_{p\to\infty} \|\mathbf{x}\|_p = \max_{i\in[n]} x_i$.

Level set of f at α is $\{x : f(x) \leq \alpha\}$.

Question : What metric does $\|\cdot\|_1$ correspond to in $\mathbb{R}^2$?

Real Inner product space

The quadruple $(X, +, *, \langle\rangle)$ defines an *inner product space* where $(X, +, *)$ is a vector space and $\langle\rangle : X \times X \rightarrow \mathbb{R}$ is an inner product s.t.

1. $\langle \mathbf{x}, \mathbf{x} \rangle = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$ and $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$
2. $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$
3. $\langle \gamma \mathbf{x}, \mathbf{y} \rangle = \gamma \langle \mathbf{x}, \mathbf{y} \rangle$ and $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$

The inner product induces a norm via $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. The geometric interpretation of the inner product so that if $\theta := \cos^{-1}(\frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|})$ then θ is the angle between the vectors.

Examples

1. The Euclidean inner product $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{n} x_i y_i$
2. More generally $\langle \mathbf{x}, \mathbf{y} \rangle_M := \mathbf{x}^\top M \mathbf{y}$ where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and M is an $n \times n$ symmetric positive definite matrix. (Exercise: Check)

Note: Many different notations are used for inner product including $\langle \mathbf{x}, \mathbf{y} \rangle = (\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$

A more complicated example – 1

Here the space is set of all functions from $[0,\infty) \to \mathbb{R}$ with the following three properties.

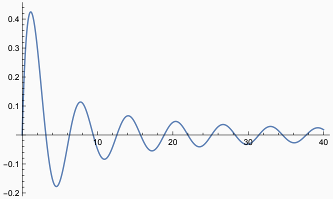
1. $f(0) = 0$
2. f is absolutely continuous (hence $f(b) - f(a) = \int_a^b f'(x)dx$)
3. $\int_0^\infty [f'(x)]^2\, dx < \infty$

Addition '$+$' is the usual addition of functions similarly multiplication by a scalar. The inner product is defined as

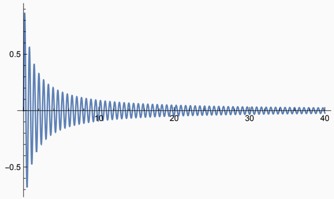
$$\langle f, g\rangle := \int_0^\infty f'(x)g'(x)dx$$

Exercise : Argue that this is an inner product space.

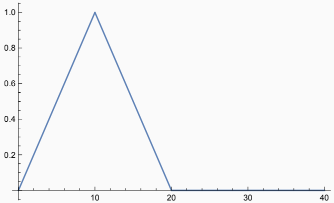
A more complicated example – 2



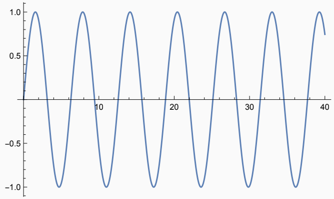
(a) $f(x) = \frac{sin(x)}{x+1}$; $\|f\| \approx 0.68$



(b) $f(x) = \frac{sin(10x)}{x+1}$; $\|f\| \approx 7.06$



(c) $f(x) = .1x[x \leq 10] + (2 - .1x)[10 < x \leq 20]$; $\|f\| = \frac{1}{\sqrt{5}}$



(d) $f(x) = sin(x)$; Not in space (why?)

Figure 2: Example functions and their norms

Review: Convexity

Convexity

Definition

A function $f : \mathcal{X} \to \mathbb{R}$ is **convex** iff $\forall p, q \in \mathcal{X}$ and $\alpha \in (0, 1)$ we have

$$f(\alpha p + (1 - \alpha)q) \leq \alpha f(p) + (1 - \alpha)f(q)$$

A function f is **concave** if $-f$ is convex. A function is additionally **strictly convex** if we can replace"$\leq$" with "$<$" when $p \neq q$.

Definition

A set $\mathcal{X}$ is convex if $p, q \in \mathcal{X} \implies (\alpha p + (1 - \alpha)q) \in \mathcal{X}$ for $\forall \alpha \in (0, 1)$.

Some results

1. If f and g are convex then $f + g$ is convex.
2. If f is convex and g is affine (linear + a constant) then $f(g(\cdot))$ is convex.
3. Suppose M is a symmetric matrix then M is a PSD matrix iff $f(x) = x^\top Mx$ is convex.
4. A level set of a convex function is convex.

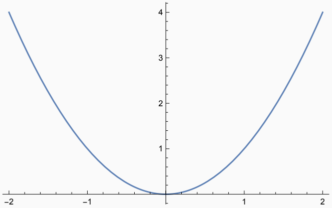
Exercise: prove the results above.

Checking for convexity

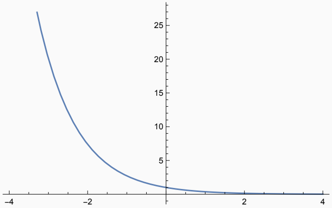
Some Results

1. For $f : (a, b) \to \mathbb{R}$ if f' is increasing then f is convex.
2. For $f : (a, b) \to \mathbb{R}$ if $f'' \geq 0$ then f is convex.
3. For $f : \mathcal{X} \to \mathbb{R}$ if $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{X}$ is a convex set and if the Hessian of f evaluated at x denoted $H(x)$ is positive semidefinite for all $x \in \mathcal{X}$ then f is convex.

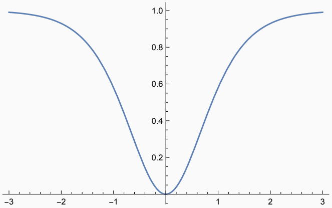
Examples



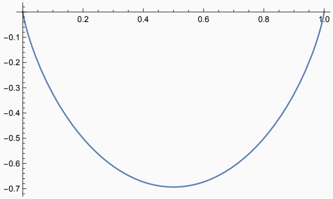
(a) $f(x) = x^2$ “Quadratic”



(b) $f(x) = e^{-x}$ “Exponential decay”



(c) $f(x) = tanh(x)^2$ “Neural” (not convex)



(d) $f(x) = x \log(x) + (1 - x) \log(1 - x)$; “negative entropy”

Why Convexity?

- At the heart of many ML algorithms is an optimisation problem.
- For example, minimize error, minimise regularised error, minimise “energy”, maximise likelihood.
- If the (unconstrained) optimisation prob. is convex and there is a minima[1] then methods gradient-based methods can be applied to smooth problems.
- On the other hand. The existence of “many” minima each associated with a distinct function suggests that for many optimisation approaches there will be a high “variance”.
- Take-away : everything else equal convex objectives in ML are simpler to work with.

[1] Or more technically a minimal surface.

Ridge Regression

Linear interpolation

Problem

We wish to find a function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ which best interpolates a data set $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \subseteq \mathbb{R}^n \times \mathbb{R}$

- If the data have been generated in the form $(\mathbf{x}, f(\mathbf{x}))$, the vectors $\mathbf{x}_i$ are linearly independent and $m = n$ then there is a unique interpolant whose parameter $\mathbf{w}$ solves

$$X\mathbf{w} = \mathbf{y}$$

  where, recall, $\mathbf{y} = (y_1, \dots, y_m)^\top$ and $X = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\top$
- Otherwise, this problem is *ill-posed*

Ill-posed problems

A problem is well-posed – in the sense of Hadamard (1902) – if

(1) a solution exists

(2) the solution is unique

(3) the solution depends continuously on the data

A problem is ill-posed if it is not well-posed

Learning problems are in general ill-posed (usually because of (2))

Regularization theory provides a general framework to solve ill-posed problems

Ridge Regression

Motivation:

1. Give a set of *k* hypothesis classes $\{\mathcal{H}_r\}_{r\in\mathbb{N}_k}$ we can choose an appropriate hypothesis class with *cross-validation*
2. An alternative compatible with linear regression is to choose a single “complex” hypothesis class and then modify the error function by adding a “complexity” term which penaltizes complex functions
3. This is known as **regularization**
4. Cross-validation may still be needed to set the regularization parameter (see below) and other parameters defining the complexity term

Ridge Regression

We minimize the regularized (penalized) empirical error

$$\mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) := \sum_{i=1}^{m}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \sum_{\ell=1}^{n} w_\ell^2 \equiv (\mathbf{y} - X\mathbf{w})^\top(\mathbf{y} - X\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

The positive parameter λ defines a trade-off between the error on the data and the norm of the vector **w** (degree of regularization)

Setting $\nabla \mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) = 0$, we obtain the modified normal equations

$$-2X^\top(\mathbf{y} - X\mathbf{w}) + 2\lambda\mathbf{w} = 0 \tag{1}$$

whose solution (called *regularized solution*) is

$$\mathbf{w} = (X^\top X + \lambda I_n)^{-1} X^\top \mathbf{y} \tag{2}$$

Dual representation

It can be shown that the regularized solution can be written as

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \quad \Rightarrow \quad f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^\top \mathbf{x} \qquad (*)$$

where the vector of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$ is given by

$$\boldsymbol{\alpha} = (XX^\top + \lambda I_m)^{-1}\mathbf{y} \tag{3}$$

• **Function representations:** we call the functional form (or representation) $f(\mathbf{x}) = \mathbf{w}^\top x$ the *primal form* and (*) the *dual form* (or representation)

The dual form is computationally convenient when $n > m$

Dual representation (continued – 1)

We rewrite eq.(1) as

$$\mathbf{w} = \frac{X^\top(\mathbf{y} - X\mathbf{w})}{\lambda}$$

Thus we have

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \tag{4}$$

with

$$\alpha_i = \frac{y_i - \mathbf{w}^\top \mathbf{x}_i}{\lambda} \tag{5}$$

Consequently, we have that

$$\mathbf{w}^\top \mathbf{x} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^\top \mathbf{x}$$

proving eq.(*).

Dual representation (continued – 2)

Plugging eq.(4) in eq.(5) we obtain

$$\alpha_i = \frac{y_i - (\sum_{j=1}^m \alpha_j \mathbf{x}_j)^\top \mathbf{x}_i}{\lambda}$$

Thus (with defining $\delta_{ij} = 1$ if $i = j$ and as 0 otherwise)

$$y_i = (\sum_{j=1}^m \alpha_j \mathbf{x}_j)^\top \mathbf{x}_i + \lambda \alpha_i$$

$$y_i = \sum_{j=1}^m (\alpha_j \mathbf{x}_j{}^\top \mathbf{x}_i + \alpha_j \lambda \delta_{ij})$$

$$y_i = \sum_{j=1}^m (\mathbf{x}_j{}^\top \mathbf{x}_i + \lambda \delta_{ij}) \alpha_j$$

Hence $(XX^\top + \lambda \mathbf{I}_m)\boldsymbol{\alpha} = \mathbf{y}$ from which eq.(3) follows.

Computational Considerations

Training time:

- Solving for **w** in the primal form requires $O(mn^2 + n^3)$ operations while solving for α in the dual form requires $O(nm^2 + m^3)$ (see $(*)$) operations

If $m \ll n$ it is more efficient to use the dual representation Running

(testing) time:

- Computing $f(\mathbf{x})$ on a test vector $\mathbf{x}$ in the primal form requires $O(n)$ operations while the dual form (see $(*)$) requires $O(mn)$ operations

Sparse representation

We can benefit even further in the dual representation if the inputs are sparse!

Example

Suppose each input $\mathbf{x} \in \mathbb{R}^n$ has most of its components equal to zero (e.g., consider images where most pixels are 'black' or text documents represented as 'bag of words')

- If k denotes the number of nonzero components of the input then computing $\mathbf{x}^\top \mathbf{t}$ requires at most $O(k)$ operations.
  $\boxed{\text{How do we do this?}}$
- If $km \ll n$ (which implies $m, k \ll n$) the dual representation requires $O(km^2 + m^3)$ computations for training and $O(mk)$ for testing

Basis Functions

Basis Functions – Explicit Feature Map

The above ideas can naturally be generalized to nonlinear function regression

By a *feature map* we mean a function $\phi : \mathbb{R}^n \to \mathbb{R}^N$

$$\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))^\top, \quad \mathbf{x} \in \mathbb{R}^n$$

- The $\phi_1, \ldots, \phi_N$ are called called *basis* functions
- Vector $\phi(\mathbf{x})$ is called the *feature vector* and the space

  $$\{\boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$$

  the *feature space*

The non-linear regression function has the primal representation

$$f(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi_j(\mathbf{x})$$

Feature Maps (Example 1 : [BIAS])

We've already seen one example with $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^{n+1}$

$$\phi(\mathbf{x}) = (\mathbf{x}, 1)^\top$$

In the context of linear regression before application of the feature map we had

$$\operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^n} \sum_{i=1}^{m} (\mathbf{w}^\top \mathbf{x}_i - y)^2$$

After the feature map we have

$$\operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}} \sum_{i=1}^{m} (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

thus allowing us to learn a linear fit with a constant offset.

Feature Maps (Example 2 : [XOR])

Consider the XOR function defined as

x_1	x_2	x_1 XOR x_2
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Does there exist a linear classifier that fits XOR perfectly? What if we add a bias term? Why?

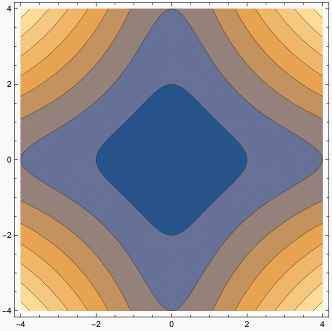
What if instead we first apply the feature map $\phi(\mathbf{x}) := (\mathbf{x}, x_1 x_2)^\top$?

Feature Maps (Example 2 : [XOR] – continued)

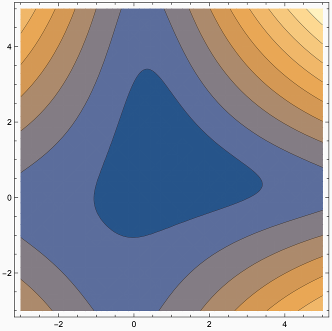
Let's visualize the unit balls associated with the "correlation" feature map

The distance between two points $\mathbf{x}, \mathbf{t}$ after mapping in feature space is

$$\|\phi(\mathbf{x}) - \phi(\mathbf{t})\|_2 = \sqrt{(x_1 - t_1)^2 + (x_2 - t_2)^2 + (x_1 x_2 - t_1 t_2)^2}$$



(a) Ball centred at $(0, 0)$.



(b) Ball centred at $(1, 1)$

Figure 3: Unit ball of 2d correlation feature map in "original space"

- Observe that the unit balls of in R^2 w.r.t. to the feature-mapped

Feature (Example 3 : Correlations)

More generally the trick behind XOR was to add to the original vector the “correlate” $x_1 x_2$.

More generally for second order correlations if $\mathbf{x} \in \mathbb{R}^n$ we have

$$\phi(\mathbf{x}) := (\mathbf{x}, x_1 x_1, x_1 x_2, \ldots, x_1 x_n, x_2 x_2, x_2 x_3, \ldots, x_2 x_n, \ldots, x_n x_n)^\top$$

I.e., $\phi : \mathbb{R}^n \to \mathbb{R}^{\frac{n^2+3n}{2}}$.

What is the motivation for this feature map?

More generally we might also include higher order correlations.

What is a potential problem with this technique?

Kernels

Computational Considerations Revisited

Again, if $m \ll N$ it is more efficient to work with the dual representation

Key observation: in the dual representation we don't need to know ϕ explicitly; we just need to know the inner product between any pair of feature vectors!

Example: Consider the following feature map with second order correlations ($N = n^2$)

$$\phi(\mathbf{x}) = (x_1 x_1, x_1 x_2, \ldots, x_n x_n)^\top$$

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle &= (x_1 x_1, x_1 x_2, \ldots, x_n x_n)(t_1 t_1, t_1 t_2, \ldots, t_n t_n)^\top \\ &= x_1 x_1 t_1 t_1 + x_1 x_2 t_1 t_2 + \ldots + x_n x_n t_n t_n \\ &= (x_1 t_1 + \ldots + x_n t_n)(x_1 t_1 + \ldots + x_n t_n) \\ &= (\mathbf{x}^\top \mathbf{t})^2\end{aligned}$$

Observe that $(\mathbf{x}^\top \mathbf{t})^2$ requires only $O(n)$ computations whereas the more direct $(x_1 x_1, x_1 x_2, \ldots, x_n x_n)(t_1 t_1, t_1 t_2, \ldots, t_n t_n)^\top$ requires $O(n^2)$

Kernel Functions – Implicit Feature Map

Given a feature map ϕ we define its associated kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as

$$K(\mathbf{x}, \mathbf{t}) := \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

- Key Point: for some feature map ϕ computing $K(\mathbf{x}, \mathbf{t})$ is independent of N (only dependent on n). Where *necessarily* $\phi(\mathbf{x})$ depends on N.

Example (cont.) If $\phi(\mathbf{x}) = (x_{i_1} x_{i_2} \cdots x_{i_r} : i_1, \ldots, i_r \in \{1, \ldots, n\})$ then we have that

$$K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$$

In this case $K(\mathbf{x}, \mathbf{t})$ is computed with $O(n)$ operations, which is essentially independent of r or $N = n^r$. On the other hand, computing $\phi(\mathbf{x})$ requires $O(N)$ operations – Exponential in r!.

Question: So far the feature map has all r-order correlates how can we change it so that (r-1)-order, (r-2)-order, etc., correlates are

Redundancy of the feature map

Warning

The feature map is not unique! If ϕ generates K so does $\hat{\phi} = \mathbf{U}\phi$ where $\mathbf{U}$ in an (any!) $N \times N$ orthogonal matrix. Even the dimension of ϕ is not unique!

Proof.

$$(\mathbf{U}\phi)^\top(\mathbf{U}\phi) = \phi^\top \mathbf{U}^\top \mathbf{U}\phi = \phi^\top \phi$$

Example

If $n = 2$, $K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^2$ is generated by both $\phi(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2, x_2 x_1)$ and $\hat{\phi}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$.

Regularization-based learning algorithms

Let us open a short parenthesis and show that the dual form of ridge regression holds true for other loss functions as well

$$\mathcal{E}_{\mathrm{emp}_\lambda}(\mathbf{w}) = \sum_{i=1}^{m} V(y_i, \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle) + \lambda\langle \mathbf{w}, \mathbf{w}\rangle, \quad \lambda > 0 \tag{6}$$

where $V : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function

Theorem

If V is differentiable wrt. its second argument and $\mathbf{w}$ is a minimizer of E_λ then it has the form

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i) \quad \Rightarrow \quad f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

This result is usually called the *Representer Theorem*

Representer theorem

Setting the derivative of E_λ wrt. $\mathbf{w}$ to zero we have

$$\sum_{i=1}^{m} V'(y_i, \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle)\phi(\mathbf{x}_i) + 2\lambda\mathbf{w} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i) \qquad (7)$$

where V' is the partial derivative of V wrt. its second argument and we defined

$$\alpha_i = \frac{1}{2\lambda} V'(y_i, \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle) \qquad (8)$$

Thus we conclude that

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle = \sum_{i=1}^{m} \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i),$$

Some remarks

- Plugging eq.(7) in the rhs. of eq.(8) we obtain a set of equations for the coefficients α_i:

$$\alpha_i = \frac{1}{2\lambda} V'\left(y_i, \sum_{j=1}^{m} K(\mathbf{x}_i, \mathbf{x}_j)\alpha_j\right) \;, \quad i = 1, \ldots, m$$

  When V is the square loss and $\phi(\mathbf{x}) = \mathbf{x}$ we retrieve the linear eq.(5)

- Substituting eq.(7) in eq.(6) we obtain an objective function for the α's:

$$\sum_{i=1}^{m} V(y_i, (\mathbf{K}\boldsymbol{\alpha})_i) + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}, \quad \text{where} : \mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{m}$$

Remark: the Representer Theorem holds true under more general conditions on V (for example V can be any continuous function)

What functions are “kernels”?

Question

Given a function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, which properties of K guarantee that there exists a *Hilbert space* $\mathcal{H}$ and a feature map $\phi : \mathbb{R}^n \to \mathcal{H}$ such that $K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle$?

Note 1

We’ve generalized the definition of *finite-dimensional* feature maps

$$\phi : \mathbb{R}^n \to \mathbb{R}^N$$

to now allow potentially *infinite-dimensional* feature maps

$$\phi : \mathbb{R}^n \to \mathcal{H}$$

Note (technical) 2

A Hilbert space is an inner product space which also contains the limit points of all its Cauchy sequences.

Positive Semidefinite Kernel

Definition

A function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is **positive semidefinite** if it is symmetric and the matrix $(K(\mathbf{x}_i, \mathbf{x}_j) : i, j = 1, \ldots, k)$ is positive semidefinite for every $k \in \mathbb{N}$ and every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^n$

Theorem

K is positive semidefinite if and only if

$$K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle, \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

for some feature map $\phi : \mathbb{R}^n \to \mathcal{W}$ and Hilbert space $\mathcal{W}$

Note. We may replace domain $\mathbb{R}^n$ by any abstract set $\mathcal{X}$ in the above definitions.

Positive semidefinite kernel (cont.)

Proof of "$\Leftarrow$"

If $K(\mathbf{x}, \mathbf{t}) = \langle \phi(\mathbf{x}), \phi(\mathbf{t}) \rangle$ then we have that

$$\sum_{i,j=1}^{m} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \sum_{i=1}^{m} c_i \phi(\mathbf{x}_i), \sum_{j=1}^{m} c_j \phi(\mathbf{x}_j) \right\rangle = \left\| \sum_{i=1}^{m} c_i \phi(\mathbf{x}_i) \right\|^2 \geq 0$$

for every choice of $m \in \mathbb{N}$, $x_i \in \mathbb{R}^d$ and $c_i \in R$, $i = 1, \ldots, m$

Note

the proof of '$\Rightarrow$' requires the notion of reproducing kernel Hilbert spaces. Informally, one can show that the linear span of the set of functions $\{K(\mathbf{x}, \cdot) : \mathbf{x} \in \mathbb{R}^n\}$ can be made into a Hilbert space H_K with inner product induced by the definition $\langle K(\mathbf{x}, \cdot), K(\mathbf{t}, \cdot) \rangle_K := K(\mathbf{x}, \mathbf{t})$. In particular, the map $\phi : \mathbb{R}^n \to H_K$ defined as $\phi(\mathbf{x}) = K(\mathbf{x}, \cdot)$ is a feature map associated with K. Observe (check!) then with $f(\cdot) := \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \cdot)$ that $\|f\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$.

Two Example Kernels

Polynomial Kernel(s)

If $p : \mathbb{R} \to \mathbb{R}$ is a polynomial with nonnegative coefficients then $K(\mathbf{x}, \mathbf{t}) = p(\mathbf{x}^\top \mathbf{t}), \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$ is a positive semidefinite kernel. For example if $a \geq 0$

- $K(\mathbf{x}, \mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$
- $K(\mathbf{x}, \mathbf{t}) = (a + \mathbf{x}^\top \mathbf{t})^r$
- $K(\mathbf{x}, \mathbf{t}) = \sum_{i=0}^{d} \frac{a^i}{i!} (\mathbf{x}^\top \mathbf{t})^i$

are each positive semidefinite kernels.

Gaussian Kernel

An important example of a “radial” kernel is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{t}) = \exp(-\beta \|\mathbf{x} - \mathbf{t}\|^2), \quad \beta > 0, \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

note: any corresponding feature map $\phi(\cdot)$ is ∞-dimensional.

Polynomial and Anova Kernel

Anova Kernel

$$K_a(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{n} (1 + x_i t_i)$$

Compare to the polynomial kernel $K_p(\mathbf{x}, \mathbf{t}) = (1 + \mathbf{x}^\top \mathbf{t})^d$

$$\mathbf{x} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix} \Rightarrow \boldsymbol{\phi}_p(\mathbf{x}) = \begin{matrix} 1 \\ \sqrt{d}x_1 \\ \sqrt{d}x_2 \\ \vdots \\ \sqrt{d}x_n \\ \sqrt{d(d-1)}x_1\,x_2 \\ \vdots \\ \sqrt{\binom{d}{i_0, i_1, \ldots, i_n}}\, x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \\ \vdots \end{matrix} \qquad \mathbf{x} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{matrix} \Rightarrow \boldsymbol{\phi}_a(\mathbf{x}) = \begin{matrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1\,x_2 \\ \vdots \\ \vdots \\ x_1\,x_2 \ldots x_n \end{matrix}$$

where $\sum_{j=0}^{n} i_j = d$

Problem: Argue that $\langle \boldsymbol{\phi}_a(\mathbf{x}), \boldsymbol{\phi}_a(\mathbf{t}) \rangle = K_a(\mathbf{x}, \mathbf{t})$.

Kernel construction

Which operations/combinations (eg, products, sums, composition, etc.) of a given set of kernels is still a kernel?

If we address this question we can build more interesting kernels starting from simple ones

Example

We have already seen that $K(\mathbf{x},\mathbf{t}) = (\mathbf{x}^\top \mathbf{t})^r$ is a kernel. For which class of functions $p : \mathbb{R} \to \mathbb{R}$ is $p(\mathbf{x}^\top \mathbf{t})$ a kernel? More generally, if K is a kernel when is $p(K(\mathbf{x},\mathbf{t}))$ a kernel?

General linear kernel

If $\mathbf{A}$ is an $n \times n$ psd matrix the function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top \mathbf{A} \mathbf{t}$$

is a kernel

Proof

Since $\mathbf{A}$ is psd we can write it in the form $\mathbf{A} = \mathbf{R}\mathbf{R}^\top$ for some $n \times n$ matrix $\mathbf{R}$. Thus K is represented by the feature map $\phi(\mathbf{x}) = \mathbf{R}^\top \mathbf{x}$

Alternatively, note that:

$$\sum_{i,j} c_i c_j \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j = \sum_{i,j} c_i c_j (\mathbf{R}^\top \mathbf{x}_i)^\top (\mathbf{R}^\top \mathbf{x}_j) = \sum_i c_i [(\mathbf{R}^\top \mathbf{x}_i)]^\top [\sum_j c_j (\mathbf{R}^\top \mathbf{x}_j)] = \| \sum_i c_i \mathbf{R}^\top \mathbf{x}_i \|^2 \geq 0$$

Kernel composition

More generally, if $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a kernel and $\phi : \mathbb{R}^n \to \mathbb{R}^N$, then

$$\tilde{K}(\mathbf{x}, \mathbf{t}) = K(\phi(\mathbf{x}), \phi(\mathbf{t}))$$

is a kernel from $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

Proof

By hypothesis, K is a kernel and so, for every $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$ the matrix $(K(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) : i, j = 1, \ldots, m)$ is psd

In particular, the previous example corresponds to $K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top \mathbf{t}$ and $\phi(\mathbf{x}) = \mathbf{R}^\top \mathbf{x}$

Kernel construction (cont.)

Question

If $K_1, \dots, K_q$ are kernels on $\mathbb{R}^n$ and $F : \mathbb{R}^q \to \mathbb{R}$, when is the function

$$F(K_1(\mathbf{x}, \mathbf{t}), \dots, K_q(\mathbf{x}, \mathbf{t})), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$$

a kernel?

Equivalently: when for every choice of $m \in \mathbb{N}$ and $\mathbf{A}_1, \dots, \mathbf{A}_q$ $m \times m$ psd matrices, is the following matrix psd?

$$(F(A_{1,ij}, \dots, A_{q,ij}) : i, j = 1, \dots m)$$

We discuss some examples of functions F for which the answer to these question is YES

Nonnegative combination of kernels

If $\lambda_j \geq 0$, $j = 1, \ldots, q$ then $\sum_{j=1}^{q} \lambda_j K_j$ is a kernel

This fact is immediate (a non-negative combination of psd matrices is still psd)

Example: Let $q = n$ and $K_j(\mathbf{x}, \mathbf{t}) = x_j t_j$.

In particular, this implies that

- aK_1 is a kernel if $a \geq 0$
- $K_1 + K_2$ is a kernel

Product of kernels

The pointwise product of two kernels K_1 and K_2

$$K(\mathbf{x},\mathbf{t}) := K_1(\mathbf{x},\mathbf{t})K_2(\mathbf{x},\mathbf{t}), \quad \mathbf{x},\mathbf{t} \in \mathbb{R}^d$$

is a kernel

Proof

Idea: The fact that the element-wise product of PSD matrices is again PSD implies that product of kernels is again a kernel.

Thus we need to show that if **A** and **B** are psd matrices, so is $\mathbf{C} = (A_{ij}B_{ij} : i,j = 1,\dots,n)$ (**C** is also called the Schur product of **A** and **B**).

Since **A** and **B** are psd we can write them in the form $\mathbf{A} = \mathbf{U}\mathbf{U}^\top$ and $\mathbf{B} = \mathbf{V}\mathbf{V}^\top$ for some $n \times n$ matrices **U** and **V**.

$$\begin{aligned}
\sum_{i,j=1}^{m} z_i z_j C_{ij} &= \sum_{ij} z_i z_j \sum_r U_{ir} U_{jr} \sum_s V_{is} V_{js} = \sum_{ij}\sum_{rs} z_i z_j U_{ir} U_{jr} V_{is} V_{js} \\
&= \sum_{rs}\sum_{ij} z_i z_j U_{ir} U_{jr} V_{is} V_{js} = \sum_{rs}\sum_{i} z_i U_{ir} V_{is} \sum_j z_j U_{jr} V_{js} \\
&= \sum_{rs}(\sum_i z_i U_{ir} V_{is})^2 \geq 0
\end{aligned}$$

Summary of constructions

Theorem

If K_1, K_2 are kernels, $a \geq 0$, A is a symmetric positive semi-definite matrix, K a kernel on $\mathbb{R}^N$ and $\phi : \mathbb{R}^n \to \mathbb{R}^N$ then the following functions are positive semidefinite kernels on $\mathbb{R}^n$

1. $\mathbf{x}^\top A \mathbf{t}$
2. $K_1(\mathbf{x}, \mathbf{t}) + K_2(\mathbf{x}, \mathbf{t})$
3. $aK_1(\mathbf{x}, \mathbf{t})$
4. $K_1(\mathbf{x}, \mathbf{t})K_2(\mathbf{x}, \mathbf{t})$
5. $K(\phi(\mathbf{x}), \phi(\mathbf{t}))$

Polynomial of kernels

Let $F = p$ where $p : \mathbb{R}^q \to \mathbb{R}$ is a polynomial in q variables with nonnegative coefficients. By properties 2,3 and 4 above we conclude that p is a valid function

In particular if $q = 1$,

$$\sum_{i=1}^{d} a_i (K(\mathbf{x}, \mathbf{t}))^i$$

is a kernel if $a_1, \ldots, a_d \geq 0$

Polynomial kernels

The above observation implies that if $p : \mathbb{R} \to \mathbb{R}$ is a polynomial with nonnegative coefficients then $p(\mathbf{x}^\top \mathbf{t}), \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$ is a kernel on $\mathbb{R}^n$. In particular if $a \geq 0$ the following are valid polynomial kernels

- $(\mathbf{x}^\top \mathbf{t})^r$
- $(a + \mathbf{x}^\top \mathbf{t})^r$
- $\sum_{i=0}^{d} \frac{a^i}{i!} (\mathbf{x}^\top \mathbf{t})^i$

‘Infinite polynomial’ kernel

If in the last equation we set $r = \infty$ the series

$$\sum_{i=0}^{r} \frac{a^i}{i!} (\mathbf{x}^\top \mathbf{t})^i$$

converges everywhere uniformly to $\exp(a\mathbf{x}^\top \mathbf{t})$ showing that this function is also a kernel.

Assume for simplicity that $n = 1$. A feature map corresponding to the kernel $\exp(axt)$ is

$$\phi(x) = \left(1,\ \sqrt{a}x,\ \sqrt{\frac{a}{2}}x^2,\ \sqrt{\frac{a^3}{6}}x^3, \ldots\right) = \left(\sqrt{\frac{a^i}{i!}}x^i : i \in \mathbb{N}\right)$$

- The feature space has an infinite dimensionality!

Translation invariant and radial kernels

We say that a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is

- *Translation invariant* if it has the form

$$K(\mathbf{x}, \mathbf{t}) = H(\mathbf{x} - \mathbf{t}), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

  where $H : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function

- *Radial* if it has the form

$$K(\mathbf{x}, \mathbf{t}) = h(\|\mathbf{x} - \mathbf{t}\|), \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

  where $h : [0, \infty) \to [0, \infty)$ is a differentiable function

The Gaussian kernel

An important example of a radial kernel is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{t}) = \exp(-\beta \|\mathbf{x} - \mathbf{t}\|^2), \quad \beta > 0, \mathbf{x}, \mathbf{t} \in \mathbb{R}^d$$

It is a kernel because it is the product of two kernels

$$K(\mathbf{x}, \mathbf{t}) = (\exp(-\beta(\mathbf{x}^\top \mathbf{x} + \mathbf{t}^\top \mathbf{t}))) \exp(2\beta \mathbf{x}^\top \mathbf{t})$$

(We saw before that $\exp(2\beta \mathbf{x}^\top \mathbf{t})$ is a kernel. Clearly $\exp(-\beta(\mathbf{x}^\top \mathbf{x} + \mathbf{t}^\top \mathbf{t}))$ is a kernel with one-dimensional feature map $\phi(\mathbf{x}) = \exp(-\beta \mathbf{x}^\top \mathbf{x})$)

Exercise:

Can you find a feature map representation for the Gaussian kernel?

Other kernels

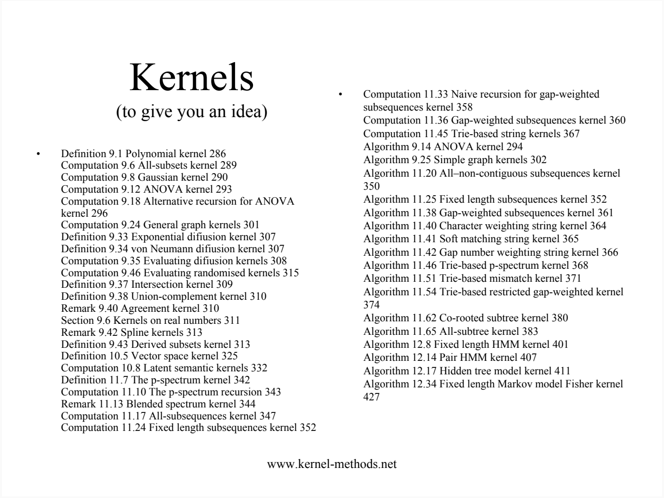
In are examples we have mainly focused on kernels defined on $\mathbb{R}^n$, more generally and usefully they can be defined on other input spaces X for example.

1. Kernels between sets
2. Kernels on text and strings
3. Kernels between graphs
4. Kernel between vertices on a graph

• Defining useful kernels between on new domains X allows for a host of ML algorithms to be transferred to that domain. For example ridge regression, k-NN, SVMs k-means, PCA, etc.

Further examples (Kernels)

From *Kernel Methods for Pattern Analysis*, Shawe-Taylor.J, and Cristianini N., Cambridge University Press (2004)



Kernels

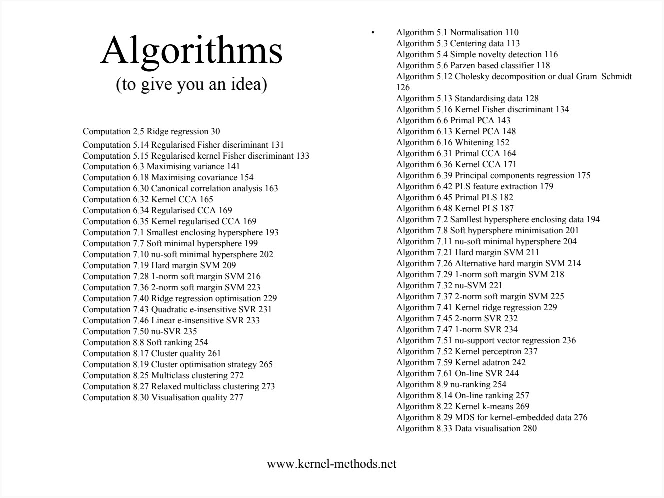
(to give you an idea)

- Definition 9.1 Polynomial kernel 286
  Computation 9.6 All-subsets kernel 289
  Computation 9.8 Gaussian kernel 290
  Computation 9.12 ANOVA kernel 293
  Computation 9.18 Alternative recursion for ANOVA kernel 296
  Computation 9.24 General graph kernels 301
  Definition 9.33 Exponential difiusion kernel 307
  Definition 9.34 von Neumann difiusion kernel 307
  Computation 9.35 Evaluating difiusion kernels 308
  Computation 9.46 Evaluating randomised kernels 315
  Definition 9.37 Intersection kernel 309
  Definition 9.38 Union-complement kernel 310
  Remark 9.40 Agreement kernel 310
  Section 9.6 Kernels on real numbers 311
  Remark 9.42 Spline kernels 313
  Definition 9.43 Derived subsets kernel 313
  Definition 10.5 Vector space kernel 325
  Computation 10.8 Latent semantic kernels 332
  Definition 11.7 The p-spectrum kernel 342
  Computation 11.10 The p-spectrum recursion 343
  Remark 11.13 Blended spectrum kernel 344
  Computation 11.17 All-subsequences kernel 347
  Computation 11.24 Fixed length subsequences kernel 352
- Computation 11.33 Naive recursion for gap-weighted subsequences kernel 358
  Computation 11.36 Gap-weighted subsequences kernel 360
  Computation 11.45 Trie-based string kernels 367
  Algorithm 9.14 ANOVA kernel 294
  Algorithm 9.25 Simple graph kernels 302
  Algorithm 11.20 All–non-contiguous subsequences kernel 350
  Algorithm 11.25 Fixed length subsequences kernel 352
  Algorithm 11.38 Gap-weighted subsequences kernel 361
  Algorithm 11.40 Character weighting string kernel 364
  Algorithm 11.41 Soft matching string kernel 365
  Algorithm 11.42 Gap number weighting string kernel 366
  Algorithm 11.46 Trie-based p-spectrum kernel 368
  Algorithm 11.51 Trie-based mismatch kernel 371
  Algorithm 11.54 Trie-based restricted gap-weighted kernel 374
  Algorithm 11.62 Co-rooted subtree kernel 380
  Algorithm 11.65 All-subtree kernel 383
  Algorithm 12.8 Fixed length HMM kernel 401
  Algorithm 12.14 Pair HMM kernel 407
  Algorithm 12.17 Hidden tree model kernel 411
  Algorithm 12.34 Fixed length Markov model Fisher kernel 427

www.kernel-methods.net

Further examples (Algorithms)

From *Kernel Methods for Pattern Analysis*, Shawe-Taylor.J, and Cristianini N., Cambridge University Press (2004)



Algorithms

(to give you an idea)

www.kernel-methods.net

Computational Summary for ridge regression

Summary : Computation with Basis Functions

Data: $X, \quad (m \times n)$; $\mathbf{y}, \quad (m \times 1)$
Basis Functions: $\phi_1, \ldots, \phi_N$ where $\phi_i : \mathbb{R}^n \to \mathbb{R}$
Feature Map: $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^N$

$$\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x})), \quad \mathbf{x} \in \mathbb{R}^n$$

Mapped Data Matrix:

$$\Phi := \begin{pmatrix} \boldsymbol{\phi}(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_m) \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \ldots & \phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_m) & \ldots & \phi_N(\mathbf{x}_m) \end{pmatrix}, \quad (m \times N)$$

Regression Coefficients: $\mathbf{w} = (\Phi^\top \Phi + \lambda I_N)^{-1} \Phi^\top \mathbf{y}$
Regression Function: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x})$

Summary : Computation with Kernels

Data: $X, \quad (m \times n)$; $\mathbf{y}, \quad (m \times 1)$
Kernel Function: $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$
Kernel Matrix:

$$\mathbf{K} := \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \dots & K(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) & \dots & K(\mathbf{x}_m, \mathbf{x}_m) \end{pmatrix}, \quad (m \times m)$$

Regression Coefficients: $\alpha = (\mathbf{K} + \lambda I_m)^{-1}\mathbf{y}$
Regression Function: $\hat{y}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x})$

Suggested Readings

Chapters 2,3 (Additionally read chapter 9 for more depth). *Kernel Methods for Pattern Analysis*, Shawe-Taylor.J, and Cristianini N., Cambridge University Press (2004)

Going Deeper ...

- *Regularization Matters: Generalization and Optimization of Neural Nets v.s. their Induced Kernel.* One of series of paper on the Neural Tangent Kernel (connecting overparameterised neural networks to a particular RKHS). One of aim of this research is to get a better understanding for why NNs generalise.
- *Convolution Kernels on Discrete Structures.* Classic paper with a variety of nice ideas on Kernels for discrete structures.

Problems – 1

1. Prove results on page 9.
2. Consider the solution to linear regression optimisation problem. When is it advantageous to compute it via the primal solution? When is it advantageous to compute it via the dual solution? Explain why
3. Given a kernel $K : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbf{R}$ where $K(\mathbf{x}, \mathbf{t}) := (1 + \langle \mathbf{x}, \mathbf{t} \rangle)^2$. Find a feature map $\phi : \mathbb{R}^2 \to \mathbb{R}^6$ which corresponds to the kernel.
4. For each of the following functions $K : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ argue whether they are a valid kernel (i.e. the kernel can be written as an inner product in some feature space) and when the answer is positive derive an associated feature map representation.
   - 4.1 $K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top D\mathbf{t}$, where D is the matrix
   $$\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$
   - 4.2 $K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top D\mathbf{t}$, where D is the matrix
   $$\begin{bmatrix} -1 & 2 \\ 2 & 4 \end{bmatrix}$$
   - 4.3 $K(\mathbf{x}, \mathbf{t}) = \exp(x_1 t_1)$, where x_1 is the first component of the vector $\mathbf{x}$ and, likewise, t_1 is the first component of the vector $\mathbf{t}$.
   - 4.4 $K(\mathbf{x}, \mathbf{t}) = \mathbf{x}^\top \mathbf{t} - (\mathbf{x}^\top \mathbf{t})^2$.
   - 4.5 Now prove that if $K : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is a given valid kernel then the following transformed kernels are also valid:
     - 4.5.1 $K(A\mathbf{x}, A\mathbf{t})$, where A is a given 2×2 matrix.
     - 4.5.2 $f(\mathbf{x})K(\mathbf{x}, \mathbf{t})f(\mathbf{t})$, where f is a given real-valued function.
5. Consider a Gaussian kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by $K(\mathbf{x}, \mathbf{z}) := e^{-\|\mathbf{x}-\mathbf{z}\|^2}$, does there exist a finite-dimensional feature map representation? I.e., does there exist a $\phi : \mathbb{R}^n \to \mathbb{R}^d$ such that $K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$? Indicate an answer "yes" or "no" and provide an argument supporting your answer. [hard]

Problems – 2

1. Argue that the vector space definition implies that every element $\mathbf{x} \in X$ has an additive inverse $-\mathbf{x} \in X$ such that $\mathbf{x} + (-\mathbf{x}) = 0$
2. *Kernels between sets.* Let X be a finite set define $K : 2^X \times 2^X \to \mathbb{R}$ as

$$K(A, B) := 2^{|A \cap B|}$$

   where $A, B \subseteq X$. Prove that K is a kernel.
3. *Min Kernel.*
   3.1 Argue that $\min(x, t)$ (where $x, t \in [0, \infty)$) is a kernel for "a more complicated example" on page 9. See discussion on 41, on going from a kernel to a Hilbert space. [technical]
   3.2 *Determining an explicit feature map.* Find a set of basis functions $\phi_i : [0, \infty) \to \mathbb{R}$ ($i = 1, 2, \ldots, \infty$) such that

$$\min(x, t) = \sum_{i=1}^{\infty} \phi_i(x)\phi_i(t)$$

   [technical, **very difficult**]