Kernels and distances for structured data

work mostly done by Thomas Gärtner (Fraunhofer Inst., Germany) and John Lloyd (Australian National University)

ILP'02, Machine Learning 57(3):205–232, 2004

See also Thomas' SIGKDD Explorations survey (2003) and ICML'04 tutorial

Support vector machines

- Wide margin classifier
  - support vectors are the datapoints closest to the separating hyper-surface
- Kernel: (implicit) transformation to feature space
  - to deal with problems that are not linearly separable in input space
  - feature space is often high-dimensional

Primal and dual form

- Linear classifiers construct a hyperplane separating the input points
  - decision rule $$h(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w} \cdot \mathbf{x} \rangle + b)$$
  - hypothesis $$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$
  - equivalently $$h(\mathbf{x}) = \mathrm{sgn}\left(\sum_i \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b \right)$$

  where α_i represent hypothesis in dual co-ordinates

Kernels

- Learning in feature space:

$$h(\mathbf{x}) = \mathrm{sgn}\left(\sum_i \alpha_i y_i \left\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \right\rangle + b\right)$$

- A kernel calculates the inner product directly in input space: $k(\mathbf{x},\mathbf{z}) = \left\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \right\rangle$
  - "similarity" between **x** and **z** in terms of features ϕ

Valid kernels

- A valid kernel $k(\mathbf{x},\mathbf{z})$ is symmetric and positive semi-definite (Mercer's theorem):

$$\forall n \geq 0, \forall c_1 \ldots c_n \in \Re : \sum_{i,j \in \{1,\ldots,n\}} c_i c_j k(x_i, x_j) \geq 0$$

  - In that case, the left-hand side of the inequality can be interpreted as a linear combination of feature vectors: $\sum_{i,j} c_i c_j k(x_i, x_j) = \left\| \sum_i c_i \phi(x_i) \right\|^2$

Our approach

- Individual-centred domains: structure within but not between individuals
- Represent individuals by terms in a higher-order logic
- Default kernel is defined recursively on type signature of individuals
- Can be adapted to domain by modifiers and domain-specific kernels on atomic terms

Basic terms (aka ground terms)

DEFINITION 3.1 (Basic terms). *The set of* basic terms, $\mathfrak{B}$, *is defined inductively as follows.*

1. *If* C *is a data constructor having signature* $\sigma_1 \to \cdots \to \sigma_n \to (T\ a_1 \ldots a_k)$, $t_1, \ldots, t_n \in \mathfrak{B}$ $(n \geq 0)$, *and* t *is* $C\ t_1 \ldots t_n \in \mathfrak{L}$, *then* $t \in \mathfrak{B}$.

2. *If* $t_1, \ldots, t_n \in \mathfrak{B}$, $s_1, \ldots, s_n \in \mathfrak{B}$ $(n \geq 0)$, $s_0 \in \mathfrak{D}$ *and* t *is*

$$\lambda x.\mathit{if}\ x = t_1\ \mathit{then}\ s_1\ \mathit{else}\ \ldots\ \mathit{if}\ x = t_n\ \mathit{then}\ s_n\ \mathit{else}\ s_0 \in \mathfrak{L},$$

   then $t \in \mathfrak{B}$.

3. *If* $t_1, \ldots, t_n \in \mathfrak{B}$ $(n \geq 0)$ *and* t *is* $(t_1, \ldots, t_n) \in \mathfrak{L}$, *then* $t \in \mathfrak{B}$.

Kernel for basic terms

DEFINITION 4.1 (Default kernel for basic terms). *The function* $k : \mathfrak{B} \times \mathfrak{B} \to \mathbb{R}$ *is defined inductively on the structure of terms in* $\mathfrak{B}$ *as follows.*

1. *If* $s, t \in \mathfrak{B}_\alpha$, *where* $\alpha = T\ \alpha_1 \ldots \alpha_k$, *for some* $T, \alpha_1, \ldots, \alpha_k$, *then*

$$k(s,t) = \begin{cases} \kappa_T(C,D) & \text{if } C \neq D \\ \kappa_T(C,C) + \sum_{i=1}^{n} k(s_i,t_i) & \text{otherwise} \end{cases}$$

   where s *is* $C\ s_1 \ldots s_n$ *and* t *is* $D\ t_1 \ldots t_m$.

2. *If* $s, t \in \mathfrak{B}_\alpha$, *where* $\alpha = \beta \to \gamma$, *for some* β, γ, *then*

$$k(s,t) = \sum_{\substack{u \in supp(s) \\ v \in supp(t)}} k(V(s\ u), V(t\ v)) \cdot k(u,v).$$

Kernel for basic terms (cont.)

3. *If* $s,t \in \mathfrak{B}_\alpha$, *where* $\alpha = \alpha_1 \times \cdots \times \alpha_n$, *for some* $\alpha_1, \ldots, \alpha_n$, *then*

$$k(s,t) = \sum_{i=1}^{n} k(s_i,t_i),$$

where s *is* $(s_1, \ldots, s_n)$ *and* t *is* $(t_1, \ldots, t_n)$.

4. *If there does not exist* $\alpha \in \mathfrak{S}^c$ *such that* $s,t \in \mathfrak{B}_\alpha$, *then* $k(s,t) = 0$.

Example for simple lists

EXAMPLE 4.1 (Default kernel on lists). *Let M be a nullary type constructor and $A,B,C,D : M$. Let $\#$ and $[]$ be the usual data constructors for lists. Choose κ_M and κ_{List} to be the matching kernel. Let s be the list $[A,B,C] \in \mathfrak{B}_{List\ M}$, $t = [A,D]$, and $u = [B,C]$. Then*

$$
\begin{aligned}
k(s,t) &= \kappa_{List}((\#),(\#)) + k(A,A) + k([B,C],[D]) \\
&= 1 + \kappa_M(A,A) + \kappa_{List}((\#),(\#)) + k(B,D) + k([C],[]) \\
&= 1 + 1 + 1 + \kappa_M(B,D) + \kappa_{List}((\#),[]) \\
&= 3 + 0 + 0 \\
&= 3.
\end{aligned}
$$

Similarly, $k(s,u) = 2$ and $k(t,u) = 3$.

Example for simple sets

EXAMPLE 4.2 (Default kernel on sets). *Let M be a nullary type constructor and $A,B,C,D : M$. Choose κ_M and κ_Ω to be the matching kernel. Let s be the set $\{A,B,C\} \in \mathfrak{B}_{M\to\Omega}$, $t = \{A,D\}$, and $u = \{B,C\}$. Then*

$$\begin{aligned}
k(s,t) &= k(A,A)k(\top,\top) + k(A,D)k(\top,\top) + k(B,A)k(\top,\top) \\
&\qquad + k(B,D)k(\top,\top) + k(C,A)k(\top,\top) + k(C,D)k(\top,\top) \\
&= \kappa_M(A,A) + \kappa_M(A,D) + \kappa_M(B,A) + \kappa_M(B,D) \\
&\qquad + \kappa_M(C,A) + \kappa_M(C,D) \\
&= 1 + 0 + 0 + 0 + 0 + 0 \\
&= 1.
\end{aligned}$$

Similarly, $k(s,u) = 2$ and $k(t,u) = 0$.

Example for simple multisets

EXAMPLE 4.3 (Default kernel on multisets). *Let M be a nullary type constructor and $A,B,C,D : M$. Choose κ_M to be the matching kernel, and κ_{Nat} to be the product kernel. Let s be $\langle A,A,B,C,C,C\rangle \in \mathfrak{B}_{M\to Nat}$ (i.e., s is the multiset containing two occurrences of A, one of B, and three of C), $t = \langle A,D,D\rangle$, and $u = \langle B,B,C,C\rangle$. Then*

$$\begin{aligned} k(s,t) &= k(2,1)k(A,A) + k(2,2)k(A,D) \\ &\qquad + k(1,1)k(B,A) + k(1,2)k(B,D) \\ &\qquad + k(3,1)k(C,A) + k(3,2)k(C,D) \\ &= 2\times 1 + 4\times 0 + 1\times 0 + 2\times 0 + 3\times 0 + 6\times 0 \\ &= 2. \end{aligned}$$

Similarly, $k(s,u) = 8$ and $k(t,u) = 0$.

Modifiers

By default, no modifier is used, i.e.,

$$\kappa_{\text{default}}(k)(x, x') = k(x, x').$$

Instead, a polynomial version of the default kernel can be used:

$$\kappa_{\text{polynomial}}(p, l)(k)(x, x') = (k(x, x') + l)^p. \qquad (l \geq 0, \;\; p \in \mathbb{Z}^+)$$

Or a Gaussian version:

$$\kappa_{\text{gaussian}}(\gamma)(k)(x, x') = e^{-\gamma[k(x,x)-2k(x,x')+k(x',x')]}. \qquad (\gamma > 0)$$

Another frequently used modification is the normalisation kernel:

$$\kappa_{\text{normalised}}(k)(x, x') = \frac{k(x, x')}{\sqrt{k(x, x)k(x', x')}}.$$

Example: East-West challenge

```
eastbound :: Train -> Bool
type Train = Car -> Bool with modifier gaussian 0.1
type Car = (Shape,Length,Roof,Wheels,Load)
data Shape = Rectangle | Oval
data Length = Long | Short
data Roof = Flat | Peaked | None with kernel roofK
type Wheels = Int with kernel discreteKernel
type Load = (LShape,LNumber)
data LShape = Rectangle | Circle | Triangle
type LNumber = Int
```

$$k_\delta(x, x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$$

```
roofK :: Roof -> Roof -> Real
roofK x x = 1
roofK Flat Peaked = 0.5
roofK Peaked Flat = 0.5
roofK x y = 0
```

From kernels to distances

DEFINITION 5.1 (Distances from kernels). *Let* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *be a kernel on* $\mathcal{X}$. *The* distance measure induced by k *is defined as*

$$d_k(x,x') = \sqrt{k(x,x) - 2k(x,x') + k(x',x')}$$

PROPOSITION 5.2 (Valid kernels induce pseudo-metrics). *Let* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *be a kernel on* $\mathcal{X}$, *and let* $d_k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *be the distance induced by* k. *If* k *is positive definite, then* d_k *is a pseudo-metric.*

Examples of distances

EXAMPLE 5.1 (Default distance on lists). *We continue Example 4.1, with* $s = [A,B,C]$, $t = [A,D]$, *and* $u = [B,C]$. *We have* $k(s,s) = 7$, $k(t,t) = 5$, *and* $k(u,u) = 5$. *Then,* $d_k(s,t) = \sqrt{k(s,s) - 2k(s,t) + k(t,t)} = \sqrt{7-6+5} = 2.45$, $d_k(s,u) = \sqrt{7-4+5} = 2.83$, *and* $d_k(t,u) = \sqrt{5-6+5} = 2$.

EXAMPLE 5.2 (Default distance on sets). *We continue Example 4.2, with* $s = \{A,B,C\}$, $t = \{A,D\}$, *and* $u = \{B,C\}$. *We have* $k(s,s) = 3$, $k(t,t) = 2$, *and* $k(u,u) = 2$. *Then,* $d_k(s,t) = \sqrt{3-2+2} = 1.73$, $d_k(s,u) = \sqrt{3-4+2} = 1$, *and* $d_k(t,u) = \sqrt{2-0+2} = 2$.

EXAMPLE 5.3 (Default distance on multisets). *We continue Example 4.3, with* $s = \langle A,A,B,C,C,C\rangle$, $t = \langle A,D,D\rangle$, *and* $u = \langle B,B,C,C\rangle$. *We have* $k(s,s) = 14$, $k(t,t) = 5$, *and* $k(u,u) = 8$. *Then,* $d_k(s,t) = \sqrt{14-4+5} = 3.87$, $d_k(s,u) = \sqrt{14-16+8} = 2.45$, *and* $d_k(t,u) = \sqrt{5-0+8} = 3.61$.

Application: spectroscopy

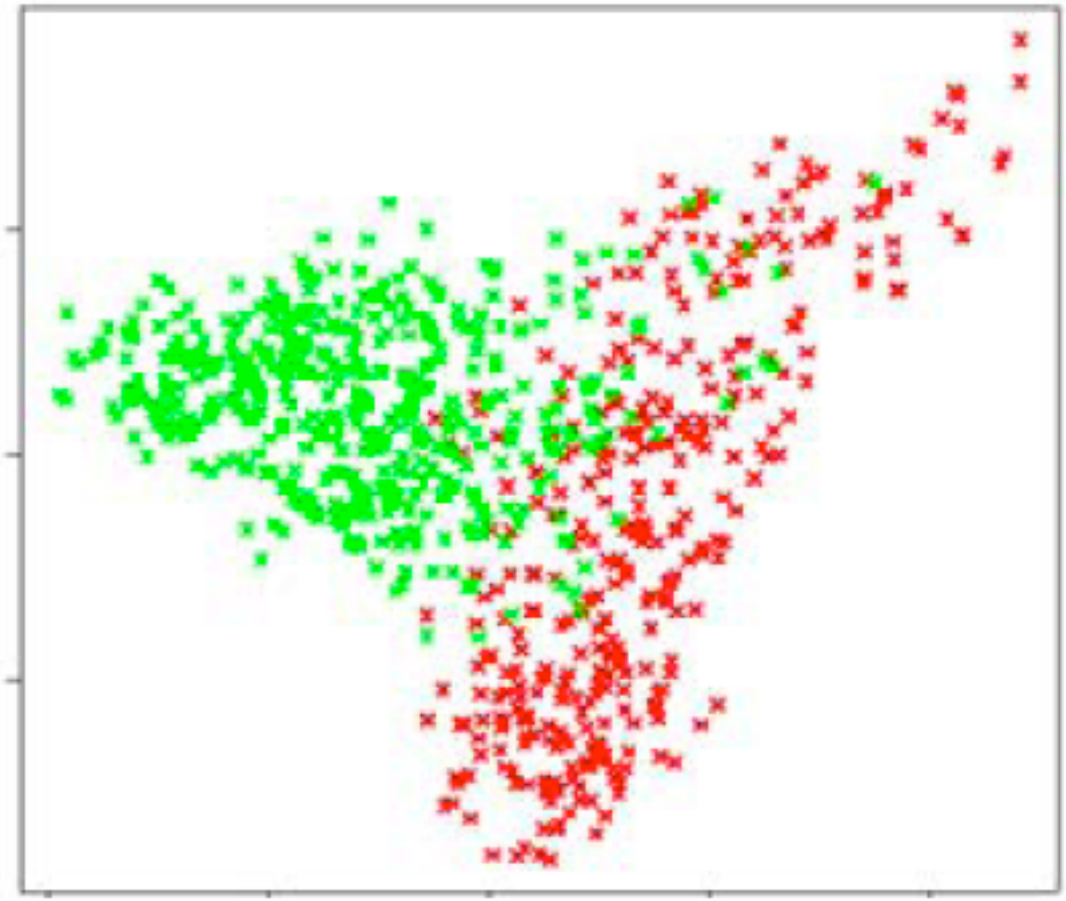
The dataset consists of 1503 spectra of diterpenes, classified into 23 different classes according to their skeleton structure as follows (number of examples per class in brackets): Trachyloban (9), Kauran (353), Beyeran (72), Atisiran (33), Ericacan (2), Gibban (13), Pimaran (155), 6,7-seco-Kauran (9), Erythoxilan (9), Spongian(10), Cassan (12), Labdan (448), Clerodan (356), Portulan (5), 5,10-seco-Clerodan (4), 8,9-seco-Labdan (6), and seven classes with only one example each.

```
type Spectrum = Frequency -> Multiplicity
type Frequency = Real with modifier gaussian 0.6
data Multiplicity = s | d | t | q | 0 with default 0
```

Table II. Classification accuracy (in %) on diterpene data

FOIL	ICL	TILDE	RIBL	KES	DES
46.5	65.3	81.6	86.5	94.7	97.1

Feature-space PCA



Application: spatial clustering

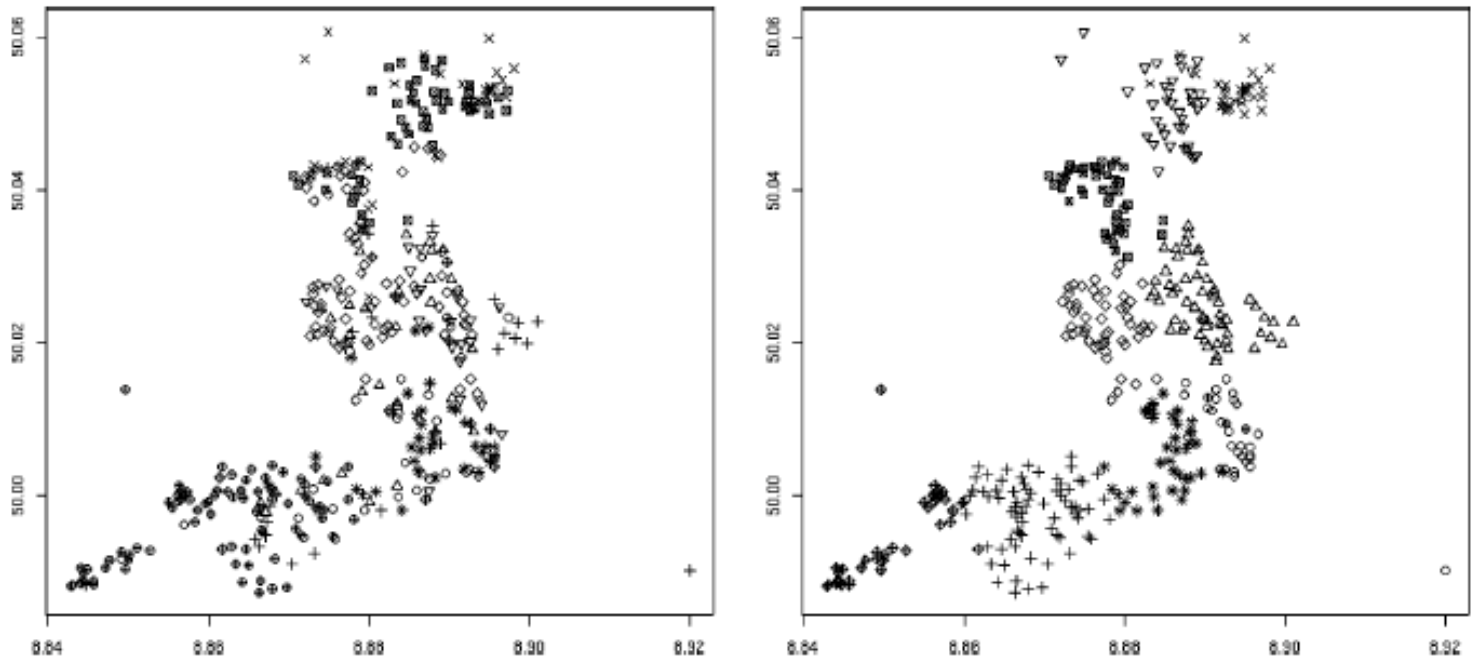


Figure 2. Plots of households spatially clustered with different parameters (left: 0.1, right: 0.02). Spatial compactness increases as the parameter is decreased.

```
type Neighbourhood = Coords -> Statistics
type Coords=(Real,Real) with modifier gaussian 0.1
type Statistics = (Real,Real,...,Real)
                                with modifier normalised
```