# Support Vector Machines - II

Prof. Dan A. Simovici

UMB

# What is a Hilbert Space?

Hilbert spaces are generalizations of Euclidean spaces.
A Hilbert space is a linear space that is equipped with an inner product such that the metric space generated by the inner product is complete.
The inner product of two elements $x, y$ of a Hilbert space $H$ is denoted by $(x, y)$. Note that in the case of $\mathbb{R}^n$ (which is a special case of a Hilbert space) the inner product of $\mathbf{x}, \mathbf{y}$ was denoted by $\mathbf{x}'\mathbf{y}$.

### Definition

A kernel over $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ such that there exists a function $\Phi : \mathcal{X} \longrightarrow H$ that satisfies the condition

$$K(u, v) = \langle \Phi(u), \Phi(v) \rangle,$$

where $H$ is a Hilbert space called the feature space.

Recall the general form of the dual optimization problem for SVMs:

$$\begin{aligned} &\textit{maximize for } \mathbf{u} \ \textstyle\sum_{i=1}^{m} u_i - \frac{1}{2} u_i u_j y_i y_j \mathbf{x}_i' \mathbf{x}_j \\ &\textit{subject to } 0 \leqslant u_i \leqslant C \textit{ and } \textstyle\sum_{i=1}^{m} u_i y_i = 0 \\ &\textit{for } 1 \leqslant i \leqslant m. \end{aligned}$$

Note the presence of the inner product $\mathbf{x}_i' \mathbf{x}_j$. This is replaced by the inner product $(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$, in the Hilbert feature space, that is, by $K(\mathbf{x}_i, \mathbf{x}_j)$, where $K$ is a suitable kernel function.

# A More General SVM Formulation

$$\begin{aligned} \textit{maximize for } \mathbf{u} \ & \sum_{i=1}^{m} u_i - \frac{1}{2} u_i u_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & \textit{subject to } 0 \leqslant a_i \leqslant C \textit{ and } \sum_{i=1}^{m} u_i y_i = 0 \\ & \textit{for } 1 \leqslant i \leqslant m. \end{aligned}$$

The hypothesis returned by the SVM algorithm is now

$$h(\mathbf{x}) = \textit{sign}\left(\sum_{i=1}^{m} u_i y_i K(\mathbf{x}_i, \mathbf{x}) - a\right).$$

with $a = y_i - \sum_{j=1}^{m} u_j y_j K(x_j, x_i)$ for any $\mathbf{x}_i$ with $0 < u_i < C$.
Note that we do not work with the feature mapping $\Phi$; instead we use the kernel only!

## Definition

Let $S$ be a non-empty set. A function $K : S \times S \longrightarrow \mathbb{C}$ is of *positive type* if for every $n \geqslant 1$ we have:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i K(x_i, x_j)\overline{a_j} \geqslant 0$$

for every $a_i \in \mathbb{C}$ and $x_i \in S$, where $1 \leqslant i \leqslant n$.

- If $K : S \times S \longrightarrow \mathbb{C}$ is of positive type, then taking $n = 1$ we have $aK(x, x)\overline{a} = K(x, x)|a|^2 \geqslant 0$ for every $a \in \mathbb{C}$ and $x \in S$, so $K(x, x) \geqslant 0$ for $x \in S$.
- If $K : S \times S \longrightarrow \mathbb{C}$ is of positive type if for every $n \geqslant 1$ and for every $x_1, \ldots, x_s$ the matrix $A_{n,K}(x_1, \ldots, x_n) = (K(x_i, x_j))$ is positive semidefinite.

## Example

The function $K : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ given by $K(x, y) = \cos(x - y)$ is of positive type because

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} a_i K(x_i, x_j)\overline{a_j} &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i \cos(x_i - x_j)\overline{a_j} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i(\cos x_i \cos x_j + \sin x_i \sin x_j)\overline{a_j} \\
&= \left|\sum_{i=1}^{n} a_i \cos x_i\right|^2 + \left|\sum_{i=1}^{n} a_i \sin x_i\right|^2.
\end{aligned}$$

for every $a_i \in \mathbb{C}$ and $x_i \in S$, where $1 \leqslant i \leqslant n$.

### Example

Let $S = \mathbb{R}^k$ and let $K : \mathbb{R}^k \times \mathbb{R}^k \longrightarrow \mathbb{R}$ be given by $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y})^d$, where $d \in \mathbb{N}$ and $d \geqslant 1$. The function $K$ is of positive type.

# Justification

We prove the existence of a function $\phi : \mathbb{R}^k \longrightarrow \mathbb{R}^m$, where $m = \binom{k+d-1}{d}$ such that $K(\mathbf{x}, \mathbf{y}) = (\phi(x), \phi(y))$.
Note that if

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix},$$

then

$$K(\mathbf{x}, \mathbf{y}) = (x_1 y_1 + \cdots + x_k y_k)^d$$

The expansion of $(x_1 y_1 + \cdots + x_k y_k)^d$ results into a sum of $m$ monomials of the form

$$\binom{d}{n_1 \ \cdots \ n_k} x_1^{n_1} y_1^{n_1} \cdots x_k^{n_k} y_k^{n_k},$$

where $n_1 + \cdots + n_k = d$ and $n_j \geqslant 0$. The number $m$ of these monomials equals the number of non-negative solutions of the equation $n_1 + \cdots + n_k = d$.

To evaluate this number we start from a sequence of $d + k - 1$ binary digits that contains $d$ ones and $k - 1$ zeroes:

$$\underbrace{11\cdots1}_{n_1}0\underbrace{11\cdots1}_{n_2}0\cdots0\underbrace{11\cdots1}_{n_k}$$

A solution of $n_1 + \cdots + n_k = d$ and $n_j \geqslant 0$ is given by the lengths of the sequences of ones determined by the positions of the $k - 1$ zeroes. Note that the total length of the sequence is $d + k - 1$. Since the positioning of the zeroes is defined by a subset containing $k - 1$ elements of the set of $d + k - 1$ positions it follows that the number of solutions is

$$\binom{d+k-1}{k-1} = \binom{d+k-1}{d}.$$

# Justification

Thus, $\phi$ is a sum of $\binom{d+k-1}{d}$ monomials:

$$\phi(\mathbf{x}) = \left( \cdots , \sqrt{\binom{d}{n_1 \ \cdots \ n_k}} x_1^{n_1} \cdots x_k^{n_k}, \cdots \right),$$

and $K(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}), \phi(\mathbf{y}))$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$.

# A Special Case

For $d = 2$ we have $\binom{d+k-1}{d} = \binom{3}{2} = 3$ and we can write:

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= (x_1 y_1 + x_2 y_2)^2 \\ &= x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2 \\ &= \left( \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{pmatrix}, \begin{pmatrix} y_1^2 \\ \sqrt{2} y_1 y_2 \\ y_2^2 \end{pmatrix} \right), \end{aligned}$$

so

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{pmatrix}$$

for $\mathbf{x} \in \mathbb{R}^2$.

# Other Functions of Positive Type

## Example

- Let $K : \mathbb{R}^k \times \mathbb{R}^k \longrightarrow \mathbb{C}$ be the function defined by $K(\mathbf{x}, \mathbf{y}) = ((\mathbf{x}, \mathbf{y}) + a)^d$, where $d \in \mathbb{N}$, $d \geqslant 1$ and $a > 0$. This is the non-homogeneous polynomial kernel.
- The radial basis function is the function $K : \mathbb{R}^k \times \mathbb{R}^k \longrightarrow \mathbb{C}$ defined by $K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$. $K$ is of positive type and $K(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}), \phi(\mathbf{y}))$, where $\phi : \mathbb{R}^k \longrightarrow \ell^2(\mathbb{R})$. Note that for this example $\phi$ ranges over an infinite-dimensional Hilbert space.

# Mercer's Theorem

## Theorem

*Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a compact set and let $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ be a continuous and symmetric function. Then $K$ admits a uniformly convergent expression*

$$K(u, v) = \sum_{n=0}^{\infty} a_n \phi_n(u) \phi_n(v)$$

*with $a_n > 0$ if and only if for every square integrable function $c \in L_2(\mathcal{X})$ we have*

$$\int \int_{\mathcal{X} \times \mathcal{X}} c(u) c(v) K(u, v) \, du \, dv \geqslant 0$$

This is qquivalent to saying that the kernel is positive definite symmetric (PDS).

# Saitoh's Theorem

**Theorem**

*A kernel* $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ *is PDS if for any* $\{x_1, \ldots, x_m\} \subseteq \mathcal{X}$ *the matrix* $\mathbf{K} = (K(x_i, x_j))$ *is symmetric and positive semidefinite.*

A symmetric matrix $\mathbf{K}$ is positive semidefinite if one of the equivalent conditions:

- the eigenvalues of $\mathbf{K}$ are non-negative, or
- for any $\mathbf{c} \in \mathbb{R}^m$, $\mathbf{c}'\mathbf{K}\mathbf{c} \geqslant 0$

hold.

From the equality for a PDS kernel

$$K(u, v) = \sum_{n=0}^{\infty} a_n \phi_n(u) \phi_n(v)$$

with $a_n > 0$ we can constract a mapping $\Phi$ into a feature space (in this case the potentially infinite $\ell_2$) as

$$\Phi(u) = \sum_{n=0}^{\infty} \sqrt{a_n} \phi_n(u)$$

## Example

For $c > 0$ a polynomial kernel of degree $d$ is the kernel defined over $\mathbb{R}^n$ by

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}'\mathbf{v} + c)^d.$$

As an example, consider $n = 2$, $d = 2$ and the kernel $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}'\mathbf{v} + c)^2$. We have

$$\begin{aligned} K(\mathbf{u}, \mathbf{v}) &= (u_1 v_1 + u_2 v_2 + c)^2 \\ &= u_1^2 v_1^2 + u_2^2 v_2^2 + c^2 + 2u_1 v_1 u_2 v_2 + 2u_1 v_1 c + 2u_2 v_2 c, \end{aligned}$$

# Example (cont'd)

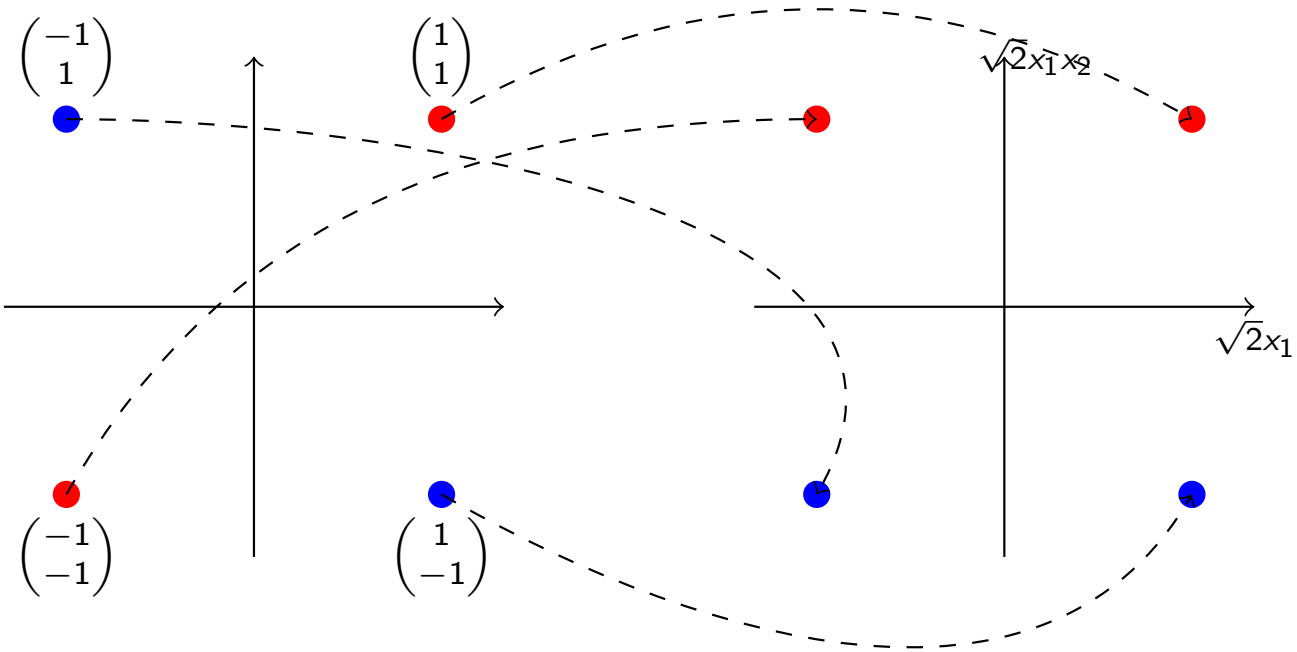
Feature space is $\mathbb{R}^6$

$$K(\mathbf{u},\mathbf{v}) = \begin{pmatrix} u_1^2 \\ u_2^2 \\ \sqrt{2}u_1u_2 \\ \sqrt{2c}u_1 \\ \sqrt{2c}u_2 \\ c \end{pmatrix}' \begin{pmatrix} v_1^2 \\ v_2^2 \\ \sqrt{2}v_1v_2 \\ \sqrt{2c}v_1 \\ \sqrt{2c}v_2 \\ c \end{pmatrix} = \Phi(\mathbf{u})'\Phi(\mathbf{v}) \text{ and } \Phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{pmatrix}$$

In general, features associated to a polynomial kernel of degree $d$ are all monomials of degree $d$ associated to the original features. It is possible to show that polynomial kernels of degree $d$ on $\mathbb{R}^n$ map the input space to a space of dimension $\binom{n+d}{d}$.

For the kernel $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}'\mathbf{v} + 1)^2$ we have

$$\Phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ 1 \end{pmatrix}.$$




For the kernel $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}'\mathbf{v} + 1)^2$ we have

$$\Phi\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\\sqrt{2}\\\sqrt{2}\\\sqrt{2}\\1\end{pmatrix}, \Phi\begin{pmatrix}-1\\-1\end{pmatrix} = \begin{pmatrix}1\\1\\\sqrt{2}\\-\sqrt{2}\\-\sqrt{2}\\1\end{pmatrix}, \Phi\begin{pmatrix}-1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\-\sqrt{2}\\-\sqrt{2}\\\sqrt{2}\\1\end{pmatrix}, \Phi\begin{pmatrix}1\\-1\end{pmatrix} = \begin{pmatrix}1\\1\\-\sqrt{2}\\\sqrt{2}\\-\sqrt{2}\\1\end{pmatrix}$$

For this set of points differences occur in the third,fourth, and fifth features.

## Definition

To any kernel $K$ we can associate a normalized kernel $K'$ defined by

$$K'(u,v) = \begin{cases} 0 & \text{if } K(u,u) = 0 \text{ or } K(v,v) = 0, \\ \frac{K(u,v)}{\sqrt{K(u,u)}\sqrt{K(v,v)}} & \text{otherwise.} \end{cases}$$

If $K(u,u) \neq 0$, then $K'(u,u) = 1$.

## Example

Let $K$ be the kernel

$$K(\mathbf{u}, \mathbf{v}) = e^{\frac{\mathbf{u}'\mathbf{v}}{\sigma^2}},$$

where $\sigma > 0$. Note that $K(\mathbf{u}, \mathbf{u}) = e^{\frac{\|\mathbf{u}\|^2}{\sigma^2}}$ and $K(\mathbf{v}, \mathbf{v}) = e^{\frac{\|\mathbf{v}\|^2}{\sigma^2}}$, hence its normalized kernel is

$$\begin{aligned} K'(\mathbf{u}, \mathbf{v}) &= \frac{K(u, v)}{\sqrt{K(u, u)}\sqrt{K(v, v)}} \\ &= \frac{e^{\frac{\mathbf{u}'\mathbf{v}}{\sigma^2}}}{e^{\frac{\|\mathbf{u}\|^2}{2\sigma^2}} e^{\frac{\|\mathbf{v}\|^2}{2\sigma^2}}} \\ &= e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}} \end{aligned}$$

### Example

For a positive constant $\sigma$ a Gaussian kernel or a radial basis function is the function $K : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ defined by

$$K(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}}.$$

**Theorem**

*Let $K$ be a PDS kernel. For any $u, v \in \mathcal{X}$ we have*

$$K(u,v)^2 \leqslant K(u,u)K(v,v).$$

**Proof:** Consider the matrix

$$\mathbf{K} = \begin{pmatrix} K(u,u) & K(u,v) \\ K(v,u) & K(v,v) \end{pmatrix}$$

$\mathbf{K}$ is positive semidefinite, so its eigenvalues $\lambda_1, \lambda_2$ must be non-negative. Its characteristic equation is

$$\begin{vmatrix} K(u,u) - \lambda & K(u,v) \\ K(v,u) & K(v,v) - \lambda \end{vmatrix} = 0$$

Equivalently,

$$\lambda^2 - (K(u,u) + K(v,v))\lambda + \det(\mathbf{K}) = 0$$

Therefore, $\lambda_1\lambda_2 = \det(\mathbf{K}) \geqslant 0$ and this implies

$$K(u,u)K(v,v) - K(u,v)^2 \geqslant 0.$$

**Theorem**

*Let $K$ be a PDS kernel. Its normalized kernel is PDS.*

**Proof:** Let $\{x_1, \ldots, x_m\} \subseteq \mathcal{X}$ and $\mathbf{c} \in \mathbb{R}^m$. We prove that $\sum_{i,j} c_i c_j K'(x_i, x_j) \geqslant 0$.
If $K(x_i, x_i) = 0$, then $K(x_i, x_j) = 0$ and, thus, $K'(x_i, x_j) = 0$ for $1 \leqslant j \leqslant m$. Thus, we may assume that $K(x_i, x_i) > 0$ for $1 \leqslant i \leqslant m$. We have

$$\begin{aligned}
\sum_{i,j} c_i c_j K'(x_i, x_j) &= \sum_{i,j} c_i c_j \frac{K(x_i, x_j)}{\sqrt{K(x_i, x_i) K(x_j, x_j)}} \\
&= \sum_{i,j} c_i c_j \frac{\langle \Phi(x_i), \Phi(x_j) \rangle}{\| \Phi(x_i) \|_H \| \Phi(x_j) \|_H} \\
&= \Big\| \sum_i \frac{c_i \Phi(x_i)}{\| \Phi(x_i) \|_H} \Big\| \geqslant 0,
\end{aligned}$$

where $\Phi$ is the feature mapping associated to $K$.

**Theorem**

*Let $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ be a PDS kernel. Then, there exists a Hilbert space $H$ of functions and a feature mapping $\Phi : \mathcal{X} \longrightarrow H$ such that $K(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}), \Phi(\mathbf{y}))$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Furthermore, $H$ has the reproducing property which means that for every $h \in H$ we have*

$$h(\mathbf{x}) = (h, K(\mathbf{x}, \cdot)).$$

The function space $H$ is called a reproducing Hilbert space associated with $K$.

- One of the reasons for the success of support vector machines is the possibility of constructing classifiers for data where the separation cannot be achieved by hyperplanes.
- The general idea is to map data in the input spaces using a non-linear map $\phi : \mathbb{R}^n \longrightarrow H$ into a Hilbert spaces $H$ referred to as the feature space.
- This is made possible by the fact that the objective function of the dual problem of SVM:

$$g(\mathbf{u}) = \sum_{i=1}^{m} u_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} u_i u_j y_i y_j \mathbf{x}_i' \mathbf{x}_j$$

depends just on the inner products $\mathbf{x}_i' \mathbf{x}_j$ of the vectors from $\mathbb{R}^n$, in other words, on the entries of the Gram matrix of the set of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$.

If a tansformation $\phi : \mathbb{R}^n \longrightarrow H$ is applied the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$, these vectors are transformed into members of the space Hilbert $H$, $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_m)$. To construct an SVM in $H$ we need to maximize an objective function of the form

$$g(\mathbf{u}) = \sum_{i=1}^{m} u_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} u_i u_j y_i y_j (\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)),$$

which depends on the values of the inner products $(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j))$ in $H$.

# Letter Image Recognition Data

Data was created by D. J. Slate and used in P. W. Frey and D. J. Slate (Machine Learning Vol 6 #2 March 91) "Letter Recognition Using Holland-style Adaptive Classifiers".

# The Structure of Data

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet.

- The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli.
- Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15.
- Training is made for the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000.

# Attribute Information

| | | | |
|---|---|---|---|
| 1. | lettr | capital letter | (26 values from A to Z) |
| 2. | x-box | horizontal position of box | (integer) |
| 3. | y-box | vertical position of box | (integer) |
| 4. | width | width of box | (integer) |
| 5. | high | height of box | (integer) |
| 6. | onpix | total # on pixels | (integer) |
| 7. | x-bar | mean x of on pixels in box | (integer) |
| 8. | y-bar | mean y of on pixels in box | (integer) |
| 9. | x2bar | mean x variance | (integer) |
| 10. | y2bar | mean y variance | (integer) |
| 11. | xybar | mean x y correlation | (integer) |
| 12. | x2ybr | mean of x * x * y | (integer) |
| 13. | xy2br | mean of x * y * y | (integer) |
| 14. | x-ege | mean edge count left to right | (integer) |
| 15. | xegvy | correlation of x-ege with y | (integer) |
| 16. | y-ege | mean edge count bottom to top | (integer) |
| 17. | yegvx | correlation of y-ege with x | (integer) |

# Class Distribution

| | | | | | | |
|---|---|---|---|---|---|---|
| 789 A | 766 B | 736 C | 805 D | 768 E | 775 F | 773 G |
| 734 H | 755 I | 747 J | 739 K | 761 L | 792 M | 783 N |
| 753 O | 803 P | 783 Q | 758 R | 748 S | 796 T | 813 U |
| 764 V | 752 W | 787 X | 786 Y | 734 Z | | |

# Data Structure of the object letters

```
> letters <- read.csv("letter-recognition.csv",header=TRUE,sep=",")
> str(letters)
'data.frame': 20000 obs. of 17 variables:
$ lettr : Factor with 26 levels "A","B","C","D",..: 20 9 4 14 7 19 2 1 10 13 ...
$ x.box : int 2 5 4 7 2 4 4 1 2 11 ...
$ y.box : int 8 12 11 11 1 11 2 1 2 15 ...
$ width : int 3 3 6 6 3 5 5 3 4 13 ...
$ high : int 5 7 8 6 1 8 4 2 4 9 ...
$ onpix : int 1 2 6 3 1 3 4 1 2 7 ...
$ x.bar : int 8 10 10 5 8 8 8 8 10 13 ...
$ y.bar : int 13 5 6 9 6 8 7 2 6 2 ...
$ x2bar : int 0 5 2 4 6 6 6 2 2 6 ...
$ y2bar : int 6 4 6 6 6 9 6 2 6 2 ...
$ xybar : int 6 13 10 4 6 5 7 8 12 12 ...
$ x2ybr : int 10 3 3 4 5 6 6 2 4 1 ...
$ xy2br : int 8 9 7 10 9 6 6 8 8 9 ...
$ xletters.ede: int 0 2 3 6 1 0 2 1 1 8 ...
$ xegvy : int 8 8 7 10 7 8 8 6 6 1 ...
$ y.ege : int 0 4 3 2 5 9 7 2 1 1 ...
$ yegvx : int 8 10 9 8 10 7 10 7 7 8 ...
```

R Packages specialized in SVMs:

- kernlab
- svmlight
- libsvm
- e1071

We shall use kernlab.

```
>
> local(pkg <- select.list(sort(.packages(all.available = TRUE)),graphics=TRUE)
+ if(nchar(pkg)) library(pkg, character.only=TRUE))
Warning message:
package kernlab was built under R version 3.0.2
```

# Construction of Training Set and Test Set

```
>
> letters_train ← letters[1:16000, ]
> letters_test ← letters[16001:20000, ]
>
```

```
> letter_classifier → ksvm(lettr ∼ .,data= letters_train,kernel = "vanilladot")
Setting default kernel parameters
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Linear (vanilla) kernel function.
Number of Support Vectors : 7037
Objective Function Value : -14.1746 -20.0072 -23.5628 -6.2009 -7.5524 -32.7694 -49.9786 -18.1824 -62.1111 -32.7
-16.2209 -32.2837 -28.9777 -51.2195 -13.276 -35.6217 -30.8612 -16.5256 -14.6811 -32.7475 -30.3219 -7.7956 -11.8
-32.3463 -13.1262 -9.2692 -153.1654 -52.9678 -76.7744 -119.2067
...
Training error : 0.130062
```

```
> letter_prediction ← predict(letter_classifier,letters_test)
> head(letter_prediction)
[1] U N V X N H
Levels: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
> table(letter_prediction,letters_test$lettr)
letter_prediction A B C D E F G
A 144 0 0 0 0 0 0
B 0 121 0 5 2 0 1
C 0 0 120 0 4 0 10
D 2 2 0 156 0 1 3
E 0 0 5 0 127 3 1
F 0 0 0 0 0 138 2
G 1 1 2 1 9 2 123
(in abbreviated form)
```

```
> agreement ← letter_prediction == letters_test$lettr
> table(agreement)
agreement
FALSE TRUE
643 3357
>
```

# Data Set Description

Attribute Information:

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class:
  - Iris Setosa
  - Iris Versicolour
  - Iris Virginica

# Data Presentation

Data contains 150 records: 50 records for each class value: *setosa*, *versicolor*, and *virginica*.

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
.
.
.
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
.
.
.
6.3,2.5,5.0,1.9,Iris-virginica
6.5,3.0,5.2,2.0,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica
```

# Uniform Distribution Generation

The data set is already grouped on class values; this requires a random rearrangement of the record in order to extract the training set and the test set.
Uniform distribution generation
The function runif generates $n$ values of a random variable uniformly distributed in the interval $[m, M]$.
It is called using

$$> \text{runif}(n, m, M)$$

```
> runif(10,12, 20)
[1] 14.81854 13.33863 17.58722 15.75252 17.11880 13.99228 19.8
12.95395 [9] 18.50042 12.46879
```

If called with one argument $n$ it produces $n$ random values in the interval $[0, 1]$.

# Ordering Permutation

The function order returns a permutation which rearranges its first argument into ascending or descending order, breaking ties by further arguments.

```
> iris_rand <- iris[order(runif(150)), ]
```

# Classifier Generation

```
> iris_train <- iris_rand[1:120,]
> iris_test <- iris_rand[121:150,]
> iris_classifier <- ksvm(class ~ .,
+ data = iris_train, kernel = "vanilladot")
> iris_prediction <- predict(iris_classifier,iris_test)
> table(iris_prediction,iris_test$class)
```

# Kernels available in kernlab

- The linear `vanilladot` is the simplest and is given by $K(\mathbf{u}, \mathbf{v}) = \mathbf{u}'\mathbf{v}$; this is useful when dealing with large sparse data vectors (typically text cathegorization).
- the Gaussian radial basis kernel `rbfdot` is $K(\mathbf{u}, \mathbf{v}) = e^{-\sigma\|\mathbf{u}-\mathbf{v}\|^2}$; a typical invocation is

$$\text{rbf} \ < -\text{rbfdot}(\text{sigma} = 0.05)$$

  This is a general kernel and is used when no further prior knowledge exists about data.
- The polynomial kernel `polydot` $K(\mathbf{u}, \mathbf{v}) = (k\mathbf{u}'\mathbf{v} + c)^d$ frequently used in image classification.

- The hyperbolic tangent kernel `tanhdot` is

$$K(\mathbf{u}, \mathbf{v}) = \tanh(k\mathbf{u}'\mathbf{v} + c)$$

  mainly used as an alternative to neural networks.
- The Laplace radial basis kernel `laplacedot`

$$K(\mathbf{u}, \mathbf{v}) = e^{-\sigma\|\mathbf{u}-\mathbf{v}\|}$$

  is a general purpose kernel.
- the ANOVA radial basis kernel `anovadot`

$$K(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{n} e^{-\sigma(u_i - v_i)^2}\right)^d$$

  used in multidimensional regression problems.

Example

```
> letter <- read.csv("letter-recognition.csv",header=TRUE,sep=",")
> letters_train <- letter[1:16000,]
> letters_test <- letter[16001:20000,]
> letter_classifier <- ksvm(lettr  .,data = letters_train,kernel="rbfdot")
Using automatic sigma estimation (sigest) for RBF or laplace kernel
> letter-classifier
Error: object 'classifier' not found
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Gaussian Radial Basis kernel function.
Hyperparameter : sigma = 0.0474609039404198
Number of Support Vectors : 8680
Objective Function Value : -43.1068 -33.8779 -59.0838 -27.2155 -34.6708 -46.8762 ....
Training error : 0.051625
```

Example

```
> letter_classifier <- ksvm(lettr  .,data = letters_train,kernel="polydot")
Setting default kernel parameters
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Polynomial kernel function.
Hyperparameters : degree = 1 scale = 1 offset = 1
Number of Support Vectors : 7035
Objective Function Value : -14.1746 -20.0072 -23.5628 -6.2009 -7.5524 -32.7694 ....
Training error : 0.130125
```

Example

```
> letter_classifier <- ksvm(lettr  .,data = letters_train,kernel="tanhdot")
Setting default kernel parameters
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Hyperbolic Tangent kernel function.
Hyperparameters : scale = 1 offset = 1
Number of Support Vectors : 15696
Objective Function Value : -15157.29 -1786.306 -15642.6 -5531.012 -1218.474 -14029.91 ...
Training error : 0.910875
```

Example

```
> letter_classifier <- ksvm(lettr  .,data = letters_train,kernel="laplacedot")
Using automatic sigma estimation (sigest) for RBF or laplace kernel
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Laplace kernel function.
Hyperparameter : sigma = 0.0477332265453678
Number of Support Vectors : 11331
Objective Function Value : -101.5121 -67.578 -131.9846 -70.7183 -77.3382 -109.682 ...
Training error : 0.084875
```

Example

```
> letter_classifier <- ksvm(lettr  .,data = letters_train,kernel="anovadot")
Setting default kernel parameters
> letter_classifier
Support Vector Machine object of class "ksvm"
SV type: C-svc (classification)
parameter : cost C = 1
Anova RBF kernel function.
Hyperparameter : sigma = 1 degree = 1
Number of Support Vectors : 6636
Objective Function Value : -8.7926 -9.3741 -12.0187 -6.6614 -5.8274 -16.8295 ...
Training error : 0.032687
```