Kernel Methods

Other sources (slides)

http://www.igi.tugraz.at/lehre/MLA/WS07/MLA_kernelmethods.pdf

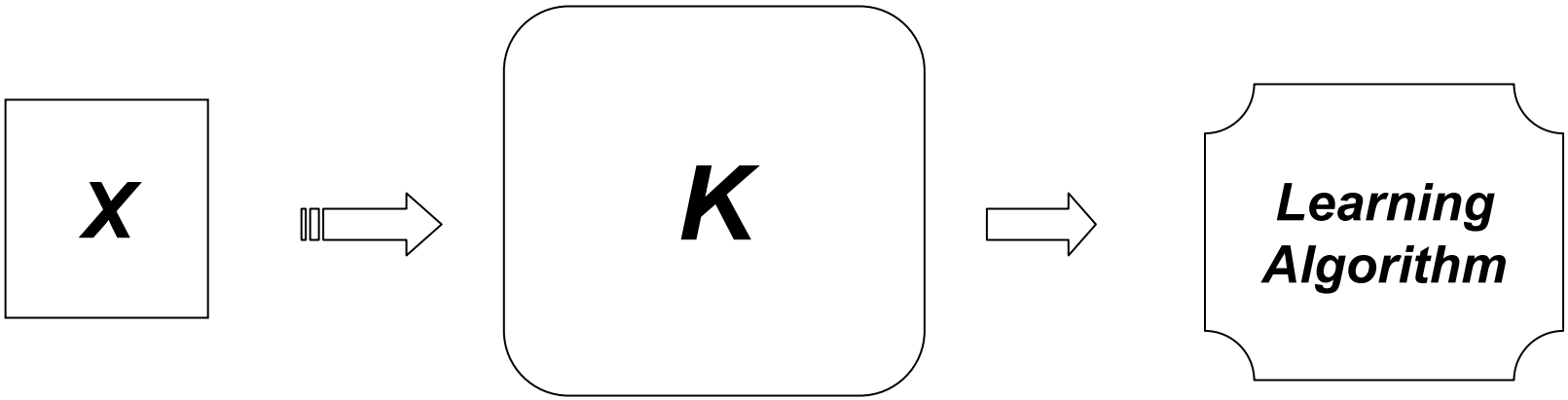
http://www.datamodelling.group.shef.ac.uk/winterschool2008/talks/Kernel%20Based%20Methods%20-%20Colin%20Campbell.pdf

Contents

- What is a kernel method?
- Kernel functions
- Some standard kernels for numerical data
- Properties of kernels
- Kernel manipulations
- More examples of kernels (strings, trees …)

A **kernel method** comprises two parts:

- Mapping into feature space
- A learning algorithm that uses that mapping



Some linear algebra

Matrix notation
http://www.purplemath.com/modules/matrices2.htm

Matrix operations
http://math.jct.ac.il/~naiman/linalg/lay/slides/c02/sec2_1ov.pdf

Vector dot products (from Wikipedia)
http://en.wikipedia.org/wiki/Dot_product

Some notation that I will use: [a b c d]' is the *transpose of the vector* containing elements a,b, c and d. (on occasion, 'T' will be used to indicate the transpose). In matrices, ';' indicates a new line of elements.

Classwork

Q1: If v1=[5 4 7 1] and v2=[3 1 2 9], find <v1,v2>, ||v1|| and ||v2||.

Q2: Find the product of the following two matrices:

[1 1 0; 2 1 5; 3 2 2] and [3 1 1; -2 0 8; -1 -3 0]

What is a kernel function?

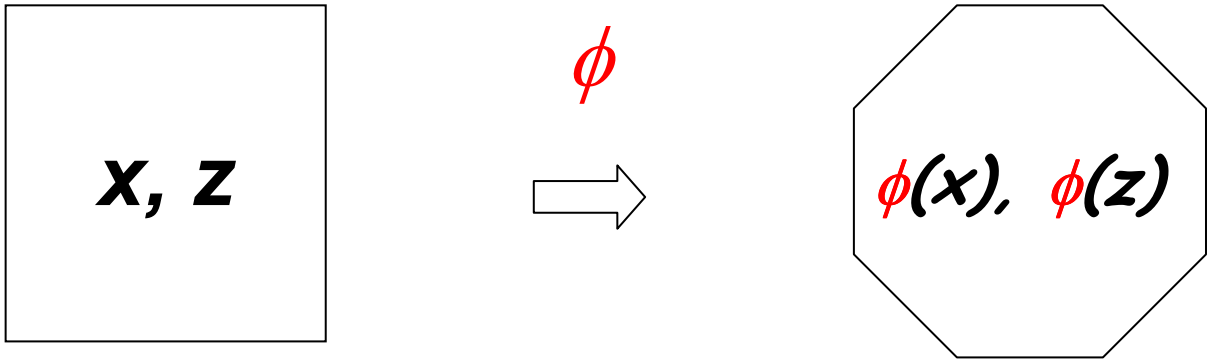
Definition 2.8, Page 34, 'Kernel Methods for Pattern Analysis' (KMPA)

A kernel is a function κ such that $\forall\ x, z \in X$

$$\kappa(x, z) = < \phi(x), \phi(z) >$$

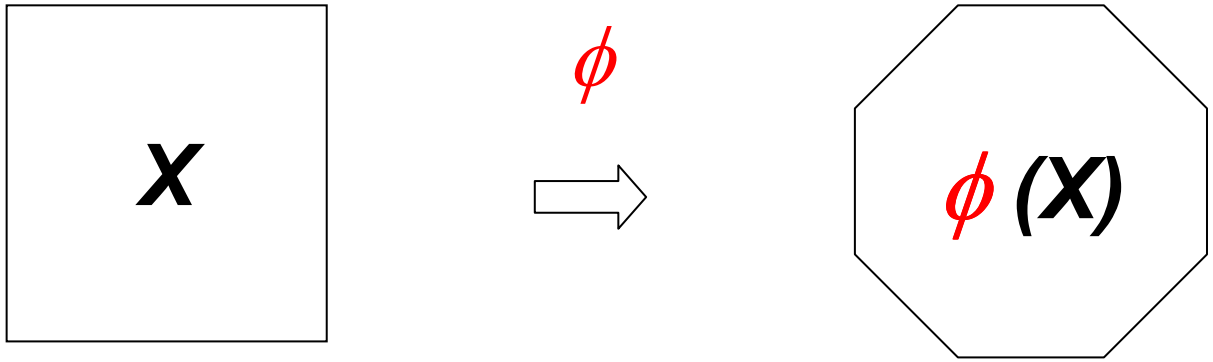
where ϕ is a mapping from X to an (inner product) features space F

$$\phi : x \rightarrow \phi(x) \in F$$



$$\kappa(x,z) =< \phi(x), \phi(z) >$$

Example 2.9 – KMPA (page 34)



$$x = (x_1, x_2) \qquad \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\Downarrow$$

$$\kappa(x, z) =< \phi(x), \phi(z) >$$

$$=< [x_1^2, x_2^2, \sqrt{2}x_1x_2], [z_1^2, z_2^2, \sqrt{2}z_1z_2] >$$

$$\ldots =< x, z >^2$$

Kernels on real numbers

The linear kernel is defined to be

$$< x_i, x_j > .$$

In matrix notation ,

$$\kappa(x) = X^T X.$$

The polynomial kernel

$$\kappa_d(x, z) = (< x, z > +R)^d.$$

The Gaussian kernel is defined to be

$$\kappa(x, z) = exp(\frac{-||x - z||^2}{2\sigma^2})$$

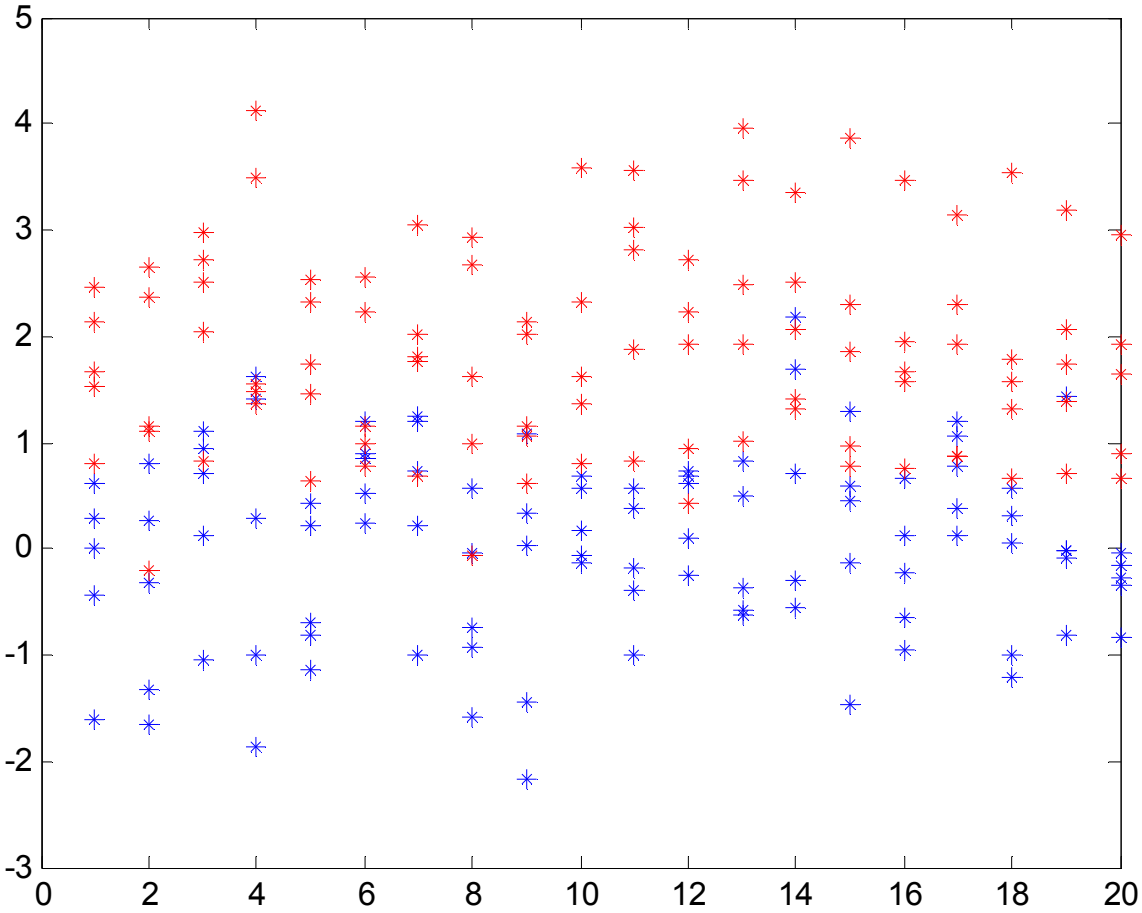
where σ is a choice parameter.

(aka the Radial Basis Function RBF)

Open question: what is the optimal σ parameter?

Kernels … so what?

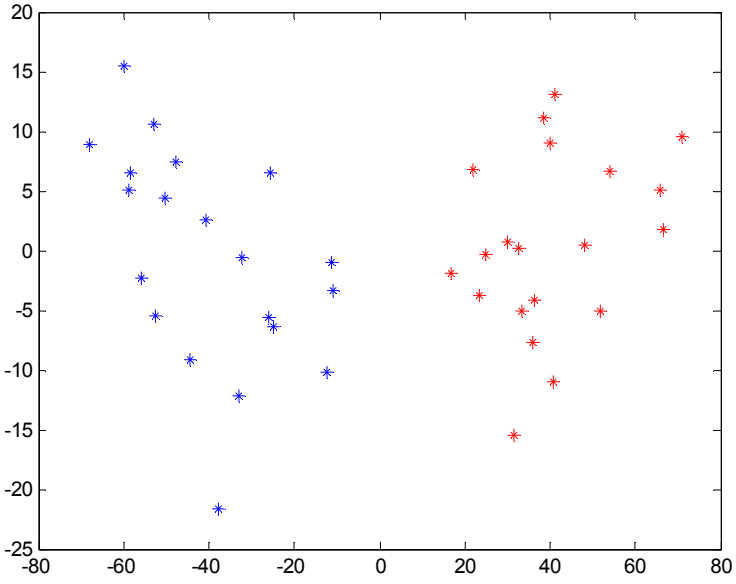
Some random 2-dim data …



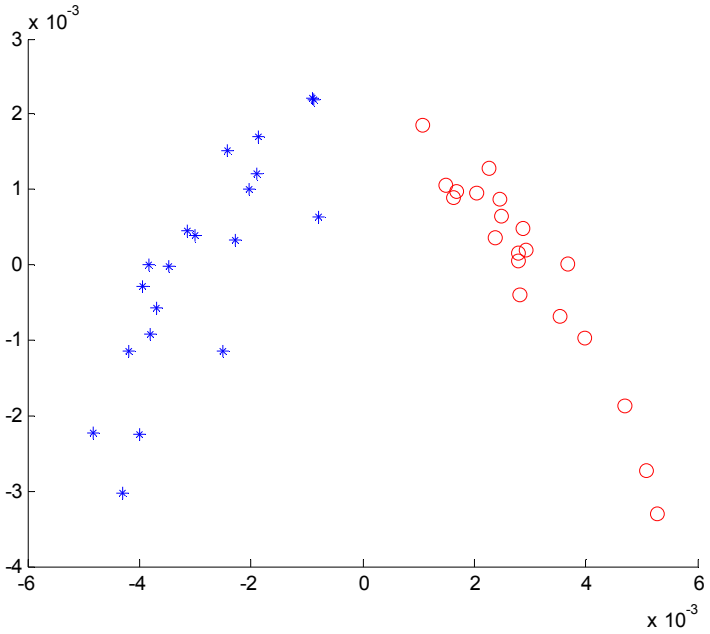
This data appears to be linearly inseparable.

But if we use kernels …

Kernelize the data …



Linear kernel



Gaussian kernel (s=100)

… data is now linearly separable in feature space.

This is actually a projection of the data from the high-dimensional feature space to 2 dimensions using PCA (more later on PCA).

psd matrices

A symmetric matrix is **p**ositive **s**emi**d**efinite if its eigenvalues are all non-negative.
(positive definite if its eigenvalues are all positive)

OR

If $v'Av \geq 0$ for all vectors v. ($v'Av > 0$ for pd)

What are eigenvalues? --- more linear algebra

- Singular matrices
- Eigenvectors
- Eigenvalues

A bit more linear algebra

A matrix A is **singular** if there exists a non-zero vector x such that $Ax=0$.

In such cases, the **determinant** is zero.

Example: Consider [6 3; -2 -1] and the vector [1 -2].

If an nxn matrix A is non-singular then the columns are said to be **linearly independent** and **span a space** of dimension n.

If $Ax= \lambda x$ for some real number λ and a vector x, then λ is said to be an **eigenvalue** of A and x is said to be the corresponding **eigenvector** of A.

Example: Consider [1 -3 3; 3 -5 3; 6 -6 4] and [-0.5 -0.5 1]'.
Multiply to get [4 4 4]. Therefore the eigenvector x=[-0.5 -0.5 1]' corresponds to the eigenvalue 4.
Note that the set of eigenvectors that span A are linearly independent (so [-1 -1 2] is equivalent to x above)

Finding the eigenvalues and eigenvectors

Since $Ax-\lambda x=0$ for an eigenvector x and matrix A, then to find the eigenvalues of A, we need to find solutions to it's **characteristic equation**, which is

$$\det(A-\lambda I)=0$$

where det is the **determinant** of the matrix A.

Once we find the eigenvalues, we can find the **eigenvectors**.

Example: Find the eigenvalues and eigenvectors of $\begin{bmatrix} 2 & -4 \\ -1 & -1 \end{bmatrix}$

$\lambda = 3, -2$ … eigenvalues

$(A-\lambda I)[x_1\ x_2]' = 0$ … solve to find eigenvectors

$x=[-4\ 1]'$ for $\lambda=3$, $x=[1\ 1]'$ for $\lambda=-2$

Gram matrix and kernels are psd

Given a set of vectors $x_1, \ldots x_\ell$, the **Gram matrix** is defined as

$$G_{ij} =< x_i, x_j >,$$

that is, the dot product between all pairs of vectors.

If we use a mapping function ϕ to project the data into some feature space, then this type of Gram matrix

$$\kappa(x_i, x_j) =< \phi(x_i), \phi(x_j) >$$

is called a **kernel**.

Gram and kernel matrices are symmetric and are **positive semidefinite**.

Kernel closure properties

Prop 3.7 (KMPA page 75)

- $\kappa(x,z) = \kappa_1(x,z) + \kappa_2(x,z)$
- $\kappa(x,z) = \alpha\kappa_1(x,z)$
- $\kappa(x,z) = \kappa_1(x,z)\kappa_2(x,z)$
- $\kappa(x,z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x,z) = \kappa_3(\phi(x),\phi(z))$
- $\kappa(x,z) = x^T Bz$ where B is a psd matrix.

This leads to a world of possibilities with kernel operations …

Normalizing kernels

The kernel is normalized as follows:

$$\kappa(x,z) = < \frac{\phi(x)}{\| \phi(x) \|}, \frac{\phi(z)}{\| \phi(z) \|} >$$

$$= \frac{< \phi(x), \phi(z) >}{\| \phi(x) \| \| \phi(z) \|}$$

$$= \frac{\kappa(x,z)}{\sqrt{\kappa(x,x)\kappa(z,z)}}$$

Centering

The idea is to move the origin of your data in the feature space to the center of mass of the data.

Centering data by kernel operations is easier than doing it in feature space.

Kernel Methods for Pattern Analysis, Ch 5, more details and code provided

Subspace projection

Often, data in high dimensional space can be well approximated in a carefully chosen low dimensional subspace.

Eg. Projection into subspace spanned by the first eigenvectors (corresponding to largest eigenvalues) … more later

Multiple kernels

What if you have multiple sets of features describing your data?

For example, you have a set of proteins that can be described in terms of their hydrophobicity profile (Kh), amino acid composition (Ka) and secondary structure (Ks). Suppose you want to predict the behavior of your proteins in certain conditions. Suppose the biologists insist that the most influential features are the protein secondary structures, but that the other features are also influential.

How can we model this using multiple kernels?

One possibility is to use weights:

$$\kappa = \alpha_h \kappa_h + \alpha_a \kappa_a + \alpha_s \kappa_s$$

where $\alpha_h + \alpha_a + \alpha_s = 1$ and α_s is the largest weight.

Sculpting the feature space …

Move the points in feature space by …

… adding a constant to K
… adding constant to diagonal
… normalizing the data

You can also use known labels for your training data to make your kernel fit the data … lots of details in Part III of *Kernel Methods for Pattern Analysis*

Kernels for structured data: Strings and Trees

Data D is said to be **structured** if it is possible to decompose it into smaller parts. (Eg. Strings can be decomposed into substrings and subsequences, Trees into subtrees)

A **decomposition structure** for such data is specified by a relation R between an element x in D and a finite set of sub-components.

A kernel can then be derived using counts of these subcomponents (aka convolution kernel, more details in Ch 11, Kernel Methods for Pattern Analysis)

String kernels

- Definitions and terms
- Spectrum kernels
- All-subsequences kernels
- Fixed length subsequences kernels
- Gap-weighted subsequences kernels

Strings: terms and definitions

Alphabet A is a finite set of symbols

A string s is any finite sequence of symbols from A

$$s = s_1 \dots s_n \text{ where } s_i \in A$$

We denote the set of all strings of length p as A^p.

A substring s' of s consists of a sequence of contiguous (one after the other) characters in s. Example: chin is a substring of machine

Denote the empty string as ε

Spectrum kernels

Define the **spectrum** of order *p* of a sequence *s* to be the histogram of frequencies of all its substrings of length *p*.

The *p*-spectrum kernel: The feature space F associated with the *p*-spectrum kernel is given by

$$\phi_u^p(s) = |\{(v_1, v_2) : s = v_1 u v_2\}|, u \in A^p$$

The associated kernel for two strings *s* and *t* is defined as

$$\kappa_p(s,t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in A^*} \phi_u^p(s) \phi_u^p(t)$$

A spectrum kernel example

Consider the words 'bar', 'bat', 'car', 'cat'. If p=2, then must consider the following substrings

ø	ar	at	ba	ca
bar	1	0	1	0
bat	0	1	1	0
car	1	0	0	1
cat	0	1	0	1

Now take the dot product.

ø	bar	bat	car	cat
bar	2	1	1	0
bat	1	2	0	1
car	1	0	2	1
cat	0	1	1	2

Now consider the two words "class" and "blast" and let p=3.

Substrings to consider are cla las ass bla las ast. Dot product is 1.

What happens with p=2?

All-subsequences kernels

A *u* is a subsequence of string s if *u* is composed of a sequence of characters in s that do not necessarily occur contiguously in s.
Example: ex is a subsequence of example, and so is xp

Let $\phi_u(s)$ denote the count of the number of times string u occurs as a subsequence in string s.
The all-subsequences kernel is defined as

$$\kappa(s,t) = \langle \phi(s), \phi(t) \rangle = \sum_u \phi_u(s)\phi_u(t)$$

for all possible subsequences u using the alphabet A.

All-subs example

empty string

ø	ε	a	b	c	r	t	aa	ar	at	ba	br	bt	ca	cr	ct	bar	baa	car	cat
bar	1	1	1	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0
baa	1	2	1	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	0
car	1	1	0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	1	0
cat	1	1	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	0	1

Take dot product … what is the kernel matrix?

	bar	baa	car	Cat
bar	8	6	4	2
baa	6	12	3	3
car	4	3	8	4
cat	2	3	4	8

Fixed length subsequences kernels

Similar to all-subsequences kernel, except that now you are interested only in subsequences of a certain length.

Gap-weighted subsequences kernels

Associate a weight λ so that the feature space for string s with a subsequence u is defined by the number of characters in s over which u is spread.

Example: gone going galleon

$$\phi(\text{gone}) = \lambda^3,\ \phi(\text{going}) = \lambda^4,\ \phi(\text{galleon}) = \lambda^7$$

Comparing Trees

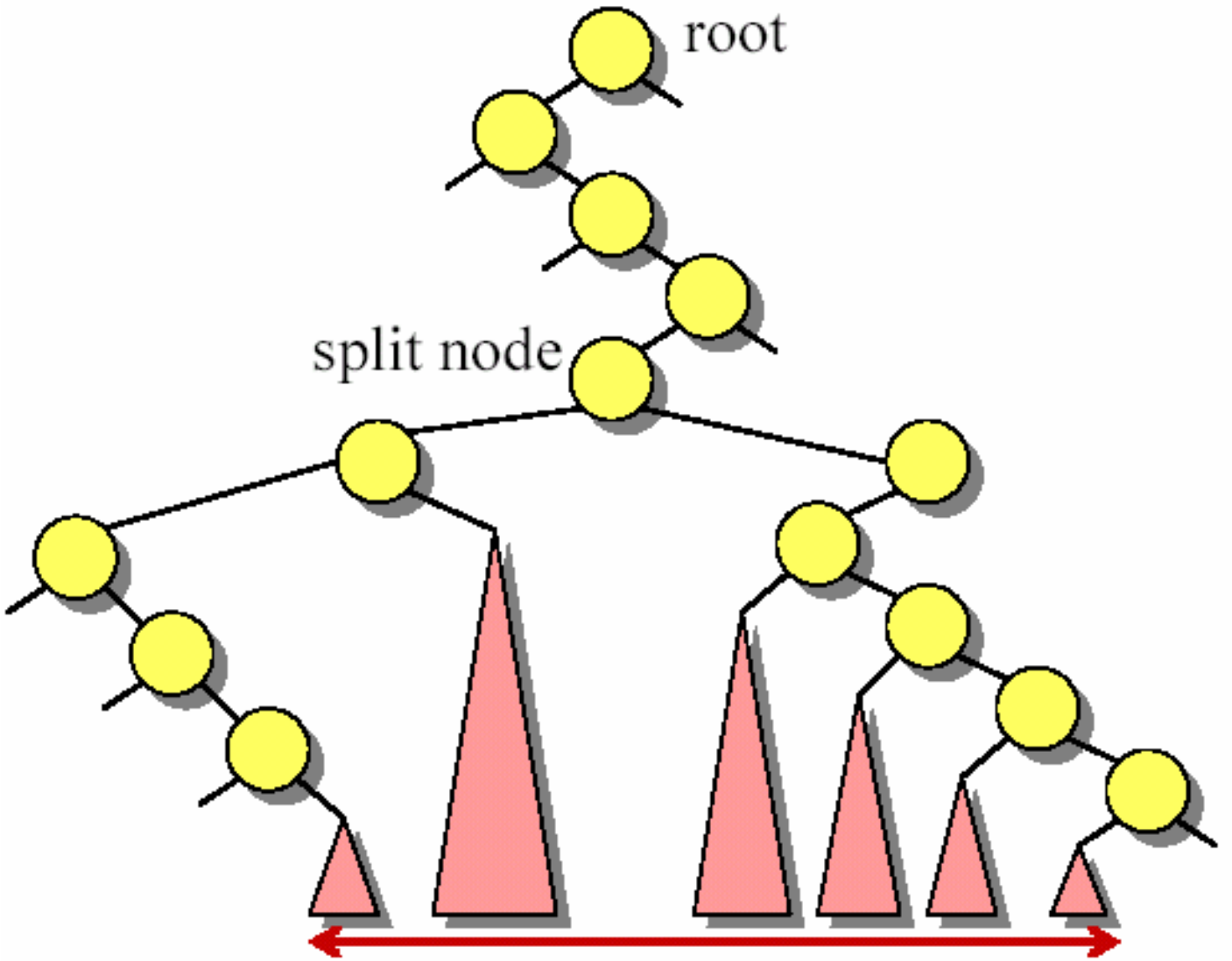
A **tree** is a directed connected acyclic graph in which each vertex (**node**) except for the **root** has in-degree one.

A **complete subtree** of a tree at a node v is a tree obtained by taking the node v together with all vertices and edges reachable from v.

A **co-rooted subtree** of a tree is the tree resulting after removing a sequence of complete subtrees and replacing their roots

A **general subtree** of a tree is any co-rooted subtree of a complete subtree.

General 1D range query



Co-rooted subtree kernel

Define the feature space associated with the co-rooted subtree kernel as the set of all proper trees with the embedding given by

$$\phi_S^r(T) = \begin{cases} 1 & \text{if } S \text{ is a subtree of } T \\ 0 & \text{otherwise} \end{cases}$$

Then the kernel is

$$\kappa_r(T_1, T_2) = \langle \phi^r(T_1), \phi^r(T_2) \rangle = \sum_S \phi_S^r(T_1)\phi_S^r(T_1)$$

All-subtree kernel

Define the feature space associated with the all-subtree kernel as the set of all proper trees with the embedding given by

$$\phi_S(T) = \begin{cases} 1 & \text{if } S \text{ is a subtree of } T \\ 0 & \text{otherwise} \end{cases}$$

Then the kernel is

$$\kappa(T_1, T_2) = \langle \phi(T_1), \phi(T_2) \rangle = \sum_S \phi_S(T_1)\phi_S(T_1)$$

Okay okay … enough about kernels.

What about kernel METHODS?

SVMs
PCA (principal component analysis)
CCA (canonical correlation analysis)
Fisher's linear discriminant analysis
Ridge regression
…
…