Graph kernels

Luc Brun

GREYC – CNRS UMR 6072, University of Caen, ENSICAEN
Luc.Brun@ensicaen.fr

In close collaboration with:
Francois Xavier Dupé (GREYC)
Benoit Gauzere (GREYC)
Didier Villemin (LCMT)
Pasquale Foggia (MIVIA)
Donatello Conte (MIVIA)
Mario Vento (MIVIA)
Graph based Representation in Pattern Recognition 2011

Motivation

Structural Pattern Recognition

- ☺ Rich description of objects
- ☹ Poor properties of graph's space does not allow to readily generalize/combine sets of graphs

Statistical Pattern Recognition

- ☹ Global description of objects
- ☺ Numerical spaces with many mathematical properties (metric, vector space, ...).

Motivation

Analyse large famillies of structural and numerical objects using a **unified** framework based on pairwise similarity.

Outline

1. Basics about Kernels
2. Graph Kernels

Definition

- A kernel k is a **symmetric** similarity measure on a set χ

$$\forall (x,y) \in \chi^2,\ k(x,y) = k(y,x)$$

- k is said to be **definite positive** (d.p.) iff k is symmetric and iff:

$$\left.\begin{array}{lcl} \forall (x_1,\dots,x_n) & \in & \chi^n \\ \forall (c_1,\dots,c_n) & \in & \mathbb{R}^n \end{array}\right\} \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i,x_j) \geq 0$$

- $K = (k(x_i,x_j))_{(i,j)\in\{1,\dots,n\}}$ is the Gramm matrix of k. k is d.p. iff:

$$\forall c \in \mathbb{R}^n - \{0\},\ c^t K c \geq 0$$

Examples

If $\chi = \mathbb{R}^n$, classical kernels include:

- Linear kernel:
$$K(x, y) = x^t y$$
- Polynomial kernel
$$K(x, y) = (x^t y)^d + c,\ c \in \mathbb{R}, d \in \mathbb{N}$$
- Cosinus kernel:
$$K(x, y) = \frac{x^t y}{\|x\| \|y\|}$$
- Rational kernel:
$$K(x, y) = 1 - \frac{\|x - y\|^2}{\|x - y\|^2 + b}, b \in \mathbb{R} - \{0\}$$
- Gaussian Kernel
$$K(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \sigma \in \mathbb{R} - \{0\}$$

Aronszajn 1950 :

A kernel k is d.p. on a space χ
if and only if
it exists

- one Hilbert space $\mathcal{H}$ and
- a function $\varphi : \chi \rightarrow \mathcal{H}$

such that:

$$k(x, y) =< \varphi(x), \varphi(y) >$$

A basic example

- Let $\chi = \mathbb{R}^2$ and $k(x, y) = (x^t y)^2 + 1$
- For any $x = (x_1, x_2)$ and $y = (y_1, y_2)$ we have:

$$\begin{array}{rcl} k(x,y) & = & (x_1 y_1 + x_2 y_2)^2 + 1 \\ & = & 1 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2 \end{array}$$

- The function φ from $\mathbb{R}^2$ to $\mathbb{R}^4$ defined by:

$$\varphi(x) = \begin{pmatrix} 1 \\ x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{pmatrix}$$

- Satisfies

$$k(x, y) =< \varphi(x), \varphi(y) >$$

- Remark: An hyperplane in $\mathcal{H} = \mathbb{R}^4$ corresponds to a quadric of $\mathbb{R}^2$.

$$< \varphi(x), n >= K \Rightarrow n_1 + n_2 x_1^2 + n_3 x_2^2 + n_4 \sqrt{2} x_1 x_2 = K$$

The Kernel Trick

- Linear classifier become non-linear using kernels

- Problem: φ is usually unknown.
- Many methods only need scalar product between data (not explicit coordinates) $\Rightarrow$ replace scalar product by kernel.
- E.g. k-NN:

$$\begin{aligned}
d_K^2(x_1,x_2) &= \|\varphi(x_1)-\varphi(x_2)\|^2 \\
&= < \varphi(x_1)-\varphi(x_2), \varphi(x_1)-\varphi(x_2) > \\
&= < \varphi(x_1),\varphi(x_1) > + < \varphi(x_2),\varphi(x_2) > -2 < \varphi(x_1),\varphi(x_2) > \\
d_K(x_1,x_2) &= k(x_1,x_1)+k(x_2,x_2)-2k(x_1,x_2)
\end{aligned}$$

- Kernel trick
  - Algorithm defined in $\mathcal{H} \Rightarrow$ (linear methods,non linear separation),
  - Data stored in χ.

Interesting but so what...

Outline

Kernel and strutured data

The kernel trick provides an implicit embedding whose metric is defined from our similarity criterion (the kernel).

1. Basics about Kernels
2. Graph Kernels
   - Graph Edit Distance
   - Kernels based on infinite Bags
   - Kernels based on finite Bags

Graph Edit distance

- Edit Path $h = e_1 \ldots, e_n \in P(G, G')$

$$G \underset{e_1}{\rightarrow} \ldots \underset{e_n}{\rightarrow} G'$$

  e_i: removal/insertion/relabelling operation.

- Cost of an edit path:

$$C(h) = \sum_{i=1}^{n} c(e_i)$$

- Graph Edit distance

$$d(G, G') = min_{h=e_1,\ldots,e_n \in P(G,G')} C(h)$$

A work intensively explored by Neuhauss & Bunke (see citations on last slide)

Graph prototypes Let $\{G_1, \ldots, G_n\}$

$$G \rightarrow vect(G) = (d(G, G_i))_{i=\{1,\ldots,n\}} \Rightarrow K(G, G') =< Vect(G), Vect($$

Diffusion Kernels

- Let B^S be a similarity matrix over S=$\{G_1, \ldots, G_n\}$. E.g.:

$$B^S(G_i, G_j) = exp(-\frac{d(G_i, G_j)^2}{2\sigma})$$

- “Force” definite positiveness:

$$K^S = exp(\lambda B^S) = \sum_{k=1}^{+\infty} \frac{\lambda^k}{k!}(B^S)^k$$

  with an appropriate choice of λ is definite positive.

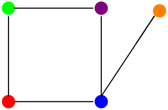
$$k^S(G_i, G_j) = K^S_{i,j}$$

  is thus a valid kernel defined on $\{G_1, \ldots, G_n\}$.

- Drawback: Any incoming data G defines a new problem on $S \cup \{G\}$.

Walk kernels

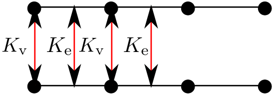
- Walks: Let $G = (V, E)$. $W = (v_1, \dots, v_n)$ is a walk iff $(v_i, v_{i+1}) \in E, \forall i \in \{1, \dots, n-1\}$.



is a walk

- Kernel between walks

$$K(h, h') = \begin{cases} 0 & \text{if } |h| \neq [h'| \text{ and} \\ K_v(v_1, v'_1).\Pi_{i=1}^{|h|} K_e(e_i, e'_i) K_v(v_{i+1}, v'_{i+1}) & \text{otherwize} \end{cases}$$



Walks kernels

- Walk kernels :

$$K(G_1,G_2)=\sum_{h\in\mathcal{W}(G_1)}\sum_{h'\in\mathcal{W}(G_2)}K(h,h')\lambda_{G_1}(h)\lambda_{G_2}(h')$$

- Covers different Graph kernels[Vert 2007,Vishwanathan et al. 2010]:

$$If\ \lambda_G(h)=\begin{cases}1\ iff\ |h|=n & K\text{is a nth order walk kernel}\\ P_G(h)(Markov\ RW) & K\text{is a random walk/marginalized kern}\\ \beta^{|h|} & K\text{is a geometric kernel}\end{cases}$$

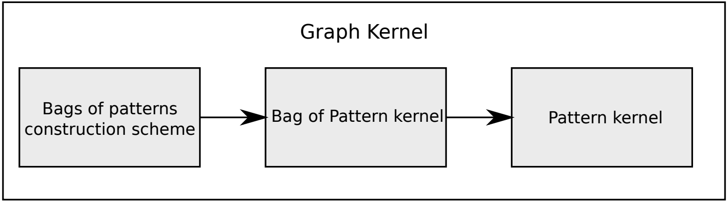
$$P_G(h)=p_s(h_1)\Pi_{i=1}^n p_t(h_i|h_{i-1})p_q(h_n)\text{ with }|h|=n$$

- Connection with diffusion algorithms on product graphs. May be computed “efficiently” using matrix inversion.
- Walks may induce totering problems: Walks with arbitrary length on the same set of edges and vertices.
- Framework extended to tree-pattern [Vert 2006, Bach 2007]

Finite Bag kernels

$$\left.\begin{array}{ccc} G & \rightarrow & B(G) \\ G' & \rightarrow & B(G') \end{array}\right\} K(G,G') = K(B(G),B(G'))$$

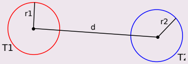
Three independent step to design a graph kernel.

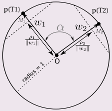


Haussler99

$$K_{\text{mean}}(T_1,T_2) = \frac{1}{|T_1|}\frac{1}{|T_2|}\sum_{t\in T_1}\sum_{t\in T_2} K_{\text{pattern}}(t,t'),$$

More complex kernels





(Desobry 2005)

$$K'(x,y) = \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}}$$
$$K'(x,x) = \|\varphi'(x)\|^2 = 1$$

Normalized kernel

Weighted mean kernel

$$\begin{aligned} K_{\text{weighted}}(T_1,T_2) &= \frac{1}{|T_1|}\frac{1}{|T_2|}\sum_{t\in T_1}\sum_{t'\in T_2}\lambda_{T_1}(t),\lambda_{T_2}(t')K_{\text{pattern}}(t,t'), \\ \lambda_{T_i}(t) &= <\varphi(t),\mu_{T_i}>^d \end{aligned}$$

Conclusion

Kernel

Direct comparison

- Vector Description
- Edit Distance
- Regularization of similarity criterion

Bags of Patterns

Infinite

- Type of Pattern (Walks,Trees)
- Walk attenuation
  - Markov,
  - geometric,
  - n^{th} order.

Finite

- Type of Pattern (Paths, Trails, Trees)
- Bag selection,
- Bag comparison
- Pattern kernel

- Graph kernels provide an implicit embedding of graphs,
- Many statistical tools may be used using the kernel trick,
- The interpretation of operations performed in the implicit Hilbert space is controled by the similarity criterion defined by the kernel (e.g. mean).

Thank you for your attention !

Questions ?

References

Indefinite Kernels Feature Space Interpretation of SVM with Indefinite Kernels, Bernard Haasdonk, PAMI 27(4) 2005

Kernels and Graph edit distance Bridgining the Gap Between Graph Edit Distance and kernel machines, Michel Neuhauss and Horst Bunke (Machine perception, AI, vol. 68).

Convolution Kernel Convolution Kernels on Discrete Structures TR: UCSC-CRL-99-10, David Haussler

Diffusion kernels Diffusion Kernels on graphs and Other Discrete Structures, Risi Imre Kondor, John Lafferty. ICML 2002

Kernels and infinite Bags

- Graph kernels, Journal of Machine Learning Research 11(2010) 1201-1242, S.V.N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, Karsten M. Borgwadt
- Marginalized Kernels Between Labeled Graphs, H. Kashima, K. Tsuda and A. Inokuchi, ICML-2003
- Graph kernels based on tree patterns for molecules, Pierre Mahé and Jean-Philippe Vert, Machine Learning 75(1) 3-35

Kernels and finite Bags

- Dupe, F. -X. & Brun, L.. Edition within a graph kernel framework for shape recognition. In Graph Based Representation in Pattern Recognition 2009 , pages 11-21 2009 .
- Next talk.