Graph Kernels

Seminário de Aprendizado de Máquina em Grafos - 2019/1

Fabricio Murai

Motivation

In many learning problems from *bio-informatics, chemo-informatics, drug discovery, web data mining and social networks*, data instances come in the form of graphs **(learning graphs)** or in the form of vertices of a given fixed graph **(learning on graphs)**.

Questions:

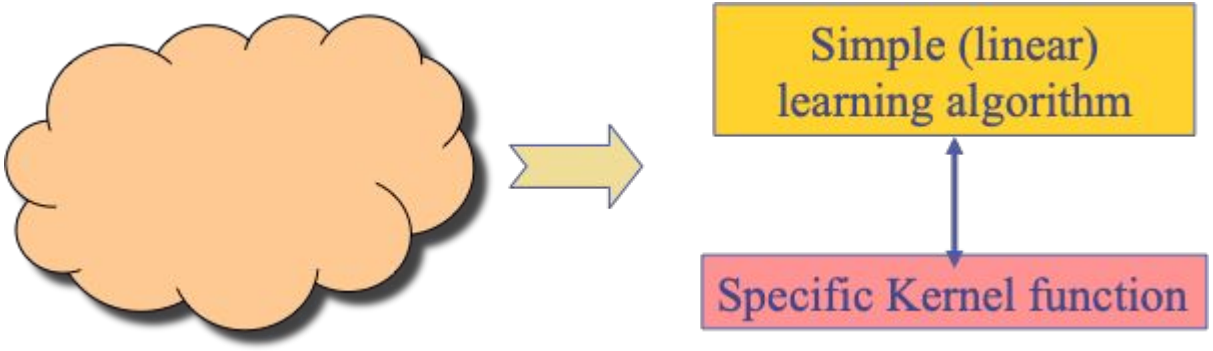
1. How similar are two graphs to each other?
2. How similar are two nodes in a given graph?

    E.g: is this protein an enzyme or not? Toxicity of a chemical molecule? Finding web pages with similar content (one step further: detecting mirrored sets of pages)

Kernels and Learning

In Kernel-based learning algorithms, problem solving is decoupled into:

- A general purpose learning algorithm (e.g., SVM, PCA) -- often linear
- A problem specific kernel



Simple (linear) learning algorithm

Specific Kernel function

Credits: Koji Tsuda

Kernel methods

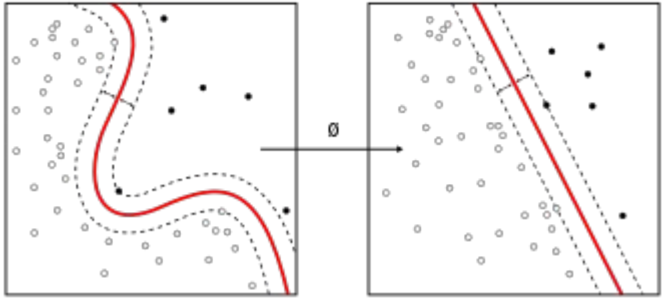
Is a class of **instance-based learners**. Best known member is SVM.

Classic SVM

$$\hat{y} = \text{sign}(\sum_{i=1}^{n}[y_i\mathbf{x}_i^\top\mathbf{x}'] + b) = \text{sign}(\mathbf{w}^\top\mathbf{x}' + b)$$

SVM is a simple linear model. Can be made more powerful by non-linear transformations of inputs into some space z.

$$\hat{y} = \text{sign}(\sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}')) = \text{sign}(\sum_{i=1}^{n}\alpha_i y_i \mathbf{z}_i^\top\mathbf{z}')$$



Kernel trick

The "trick" comes from the fact that we can define a function K(**x**,**x**') = **z**[⊤]**z**' that corresponds to the inner product of the images of **x** and **x**' in the z space without having to compute (or even define) ϕ explicitly.

Example

$$\mathbf{x} = (x_1, x_2)$$

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^2 = (1 + 2(x_1 x_1' + x_2 x_2') + x_1^2 x_1'^2 + 2x_1 x_1' x_2 x_2' + x_2^2 x_2'^2)$$

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)$$

$$\phi(\mathbf{x}') = (1, \sqrt{2}x_1', \sqrt{2}x_2', x_1'^2, \sqrt{2}x_1' x_2', x_2'^2)$$

More examples of kernels

$$\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

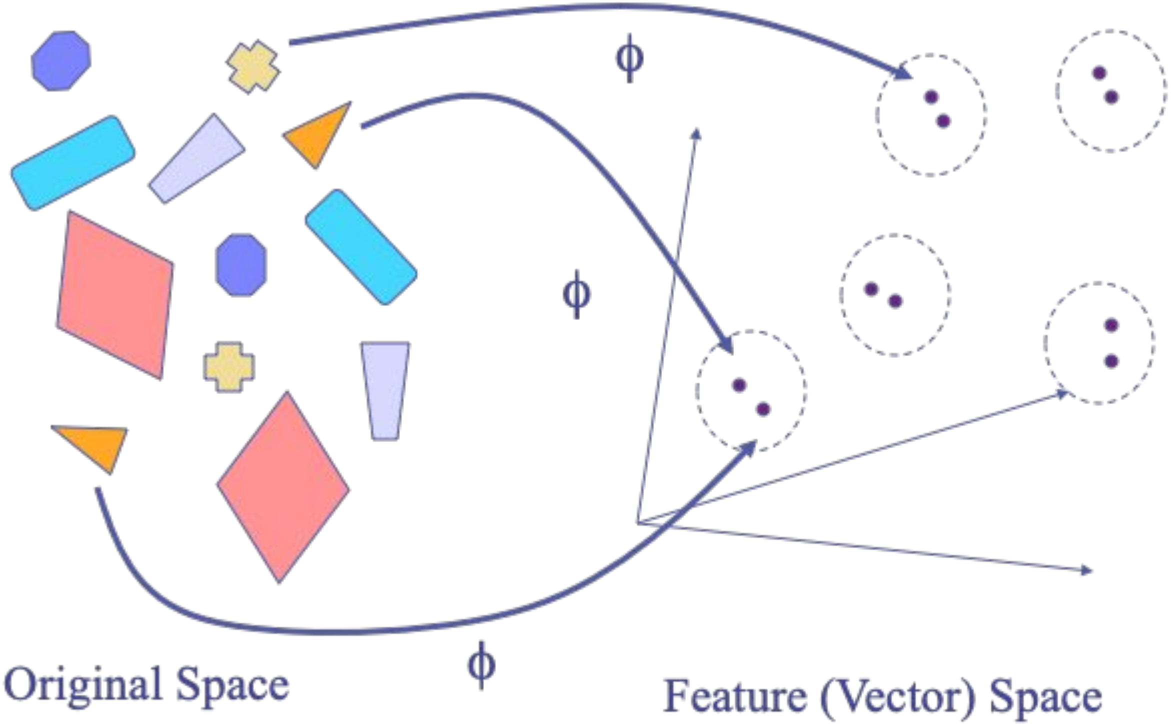
$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^Q = (1 + x_1 x_1' + \ldots + x_d x_d')^Q$$

For d=10, Q=100, what is the dimension of ϕ?

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|_2^2)$$

In this case, ϕ has infinite dimension

Kernel methods: the mapping

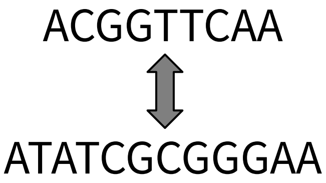


Kernels for structures

How do we define kernels for other kinds of objects?

Example

Similarity between sequences of different lengths?



ACGGTTCAA

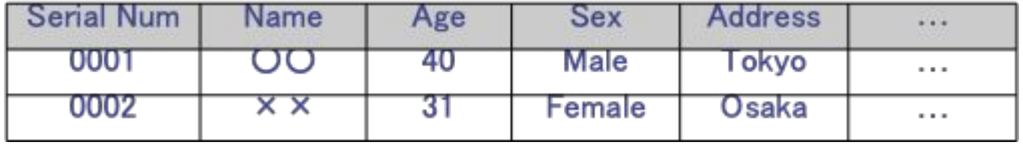
ATATCGCGGGAA

Idea (Count kernel): counts the number of symbols and take the inner product. Not good for sequences with frequent context change (e.g., coding/non-coding regions of DNA).

Marginalized kernels have been used for this purpose (but won't be covered here).

(More) motivation for graph kernels

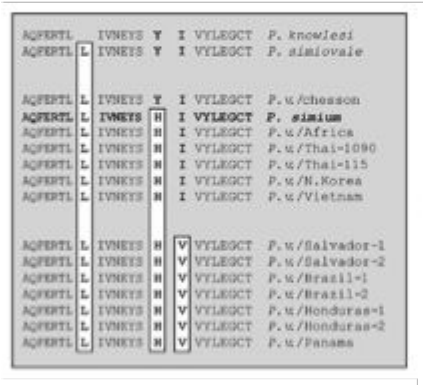
- Existing methods assume "tables"



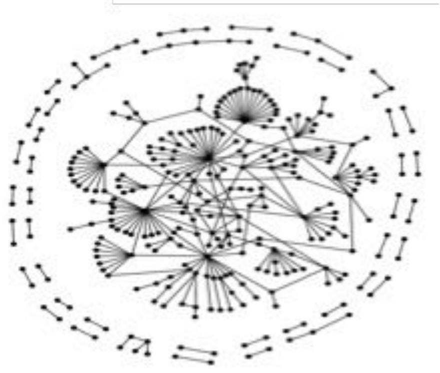
Serial Num	Name	Age	Sex	Address	…
0001	○○	40	Male	Tokyo	…
0002	× ×	31	Female	Osaka	…

- Structure data beyond this framework

  New methods for analysis

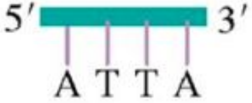


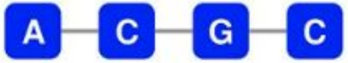
```
AQFERTL   IVNEYS Y  I VYLEGCT  P. knowlesi
AQFERTL L IVNEYS Y  I VYLEGCT  P. simiovale

AQFERTL L IVNEYS Y  I VYLEGCT  P.v/chesson
AQFERTL L IVNEYS H  I VYLEGCT  P. simium
AQFERTL L IVNEYS H  I VYLEGCT  P.v/Africa
AQFERTL L IVNKYS H  I VYLEGCT  P.v/Thai-1090
AQFERTL L IVNKYS H  I VYLEGCT  P.v/Thai-115
AQFERTL L IVNKYS H  I VYLEGCT  P.v/N.Korea
AQFERTL L IVNKYS H  I VYLEGCT  P.v/Vietnam

AQFERTL L IVNEYS H  V VYLEGCT  P.v/Salvador-1
AQFERTL L IVNEYS H  V VYLEGCT  P.v/Salvador-2
AQFERTL L IVNKYS H  V VYLEGCT  P.v/Brazil-1
AQFERTL L IVNKYS H  V VYLEGCT  P.v/Brazil-2
AQFERTL L IVNKYS H  V VYLEGCT  P.v/Honduras-1
AQFERTL L IVNKYS H  V VYLEGCT  P.v/Honduras-2
AQFERTL L IVNKYS H  V VYLEGCT  P.v/Panama
```



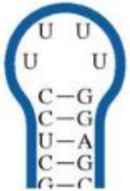
Graph structures in biology

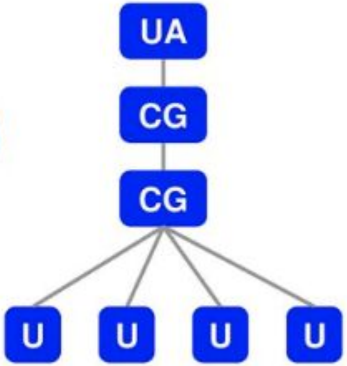
- DNA Sequence



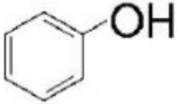


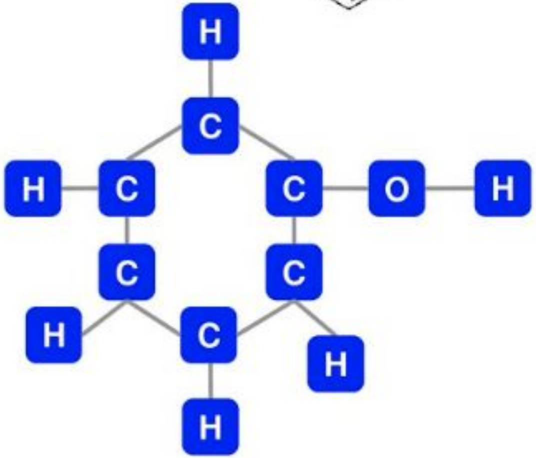
- RNA





◆ Compounds





Kernel methods

A natural framework to study these questions.

A kernel K(x,x') is a measure of similarity between two objects x and x'.

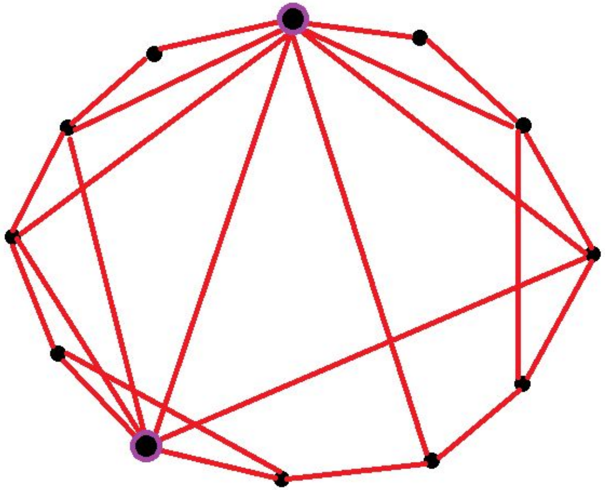
Set of objects X and a kernel K(.,.) define the Gram matrix K, which must satisfy two properties:

- Symmetric
- Positive semi-definite

Kernels on graphs

How to define similarity between 2 nodes u and v?

- Geodesic distance?
- Probability that a l-length random starting from u visits v?



Exponential kernels

H: symmetric matrix called generator

Exponential kernel

$$K = \exp(\beta H) = \lim_{n \to \infty} \left(\mathbb{I} + \frac{\beta H}{n} \right)^n = \mathbb{I} + \beta H + \frac{\beta^2 H^2}{2!} + \frac{\beta^3 H^3}{3!} + \dots$$

Where β is the diffusion parameter, and **exp** is matrix exponential, not element-wise.

It is well known that any power of a symmetric matrix is symmetric and positive semidefinite. Replacing n by 2n, we show that K is positive definite.

Exponential kernels: a dynamic process

Taking the derivative of K yields

$$\frac{d}{d\beta}K_\beta = HK_\beta$$

Examining the equation with initial condition K(0)=I, lends to interpretation that

- K(β) is the result of a continuous process,
- gradually transforming identity matrix I to a kernel with stronger and stronger off-diagonal effects as β increases.

Diffusion kernels on graphs

A: adjacency matrix

D: diagonal matrix of degrees

L = D-A: Graph Laplacian matrix

Diffusion kernel matrix

$$K = \exp(\beta H) = \lim_{n \to \infty} \left(\mathbb{I} + \frac{\beta H}{n} \right)^n = \mathbb{I} + \beta H + \frac{\beta^2 H^2}{2!} + \frac{\beta^3 H^3}{3!} + \ldots$$

Where β is the diffusion parameter, H=-L, and **exp** is matrix exponential, not element-wise.

Relationship with heat equation in physics.

Intuition: "K(i,j) is that amount of heat transferred from node i to node j"

Relationship to random walks

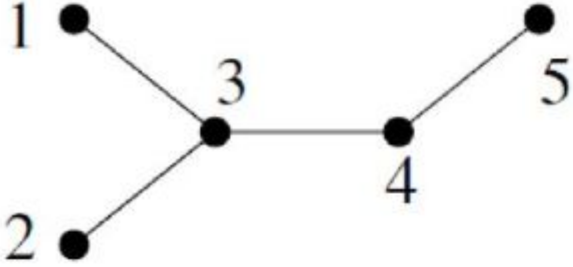
A lazy random walk on graph G with parameter $\beta_0 < 1/\text{max degree}$ is a stochastic process $S=\{i_0,\ldots,i_N\}$ over V. At step t, let the state of the process be $i_t=v$. Then, it

- Goes to one of v's neighbors with probability β_0
- Stays at v with probability $1 - \beta_0 * \text{degree}$.

Considering the distribution $\Pr(z_N \mid z_0)$ in the limit $\Delta t \to 0$ with $N=(1/\Delta t)$ and $\beta=\beta_0 \Delta t$ yields exactly the diffusion equation.

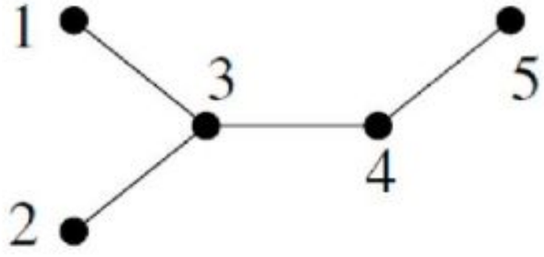
Hence, diffusion kernel is the continuous time limit of lazy random walks.

Adjacency matrix and degree matrix



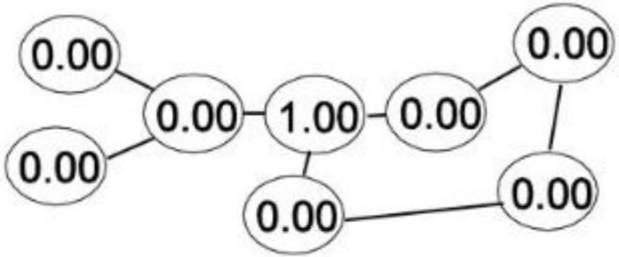
$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Graph Laplacian Matrix L

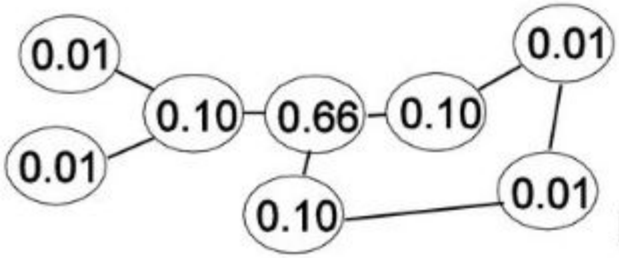


$$L = D - A = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

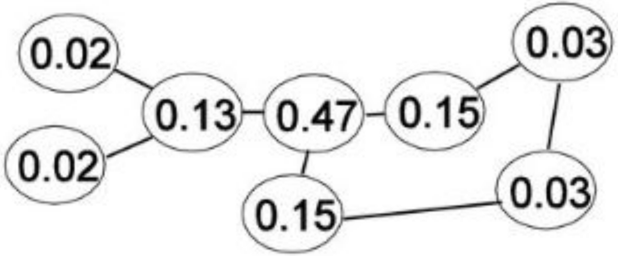
Actual values of diffusion kernels



β=0



β=0.15



β=0.3

Closeness from the "central node"

How to compute exp(A)?

When A is a (n,n) matrix, its eigendecomposition is

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

$$\exp(A) = \sum_{i=1}^{n} \exp(\lambda_i) v_i v_i^T$$

Cost of eigendecomposition is O(n^3). 😱

Special cases where K can be computed directly

- K-regular trees
- Complete graphs
- Closed chains
- The hypercube: $x=(x_1,\ldots,x_m)$ onde $x_i \in \{0,1\}$. Two sequences x and x' are neighbors if they differ only in a single digit.

$$K(x,x') \propto \left(\frac{1-e^{-2\beta}}{1+e^{-2\beta}}\right)^{d(x,x')} = (\tanh\beta)^{d(x,x')}$$

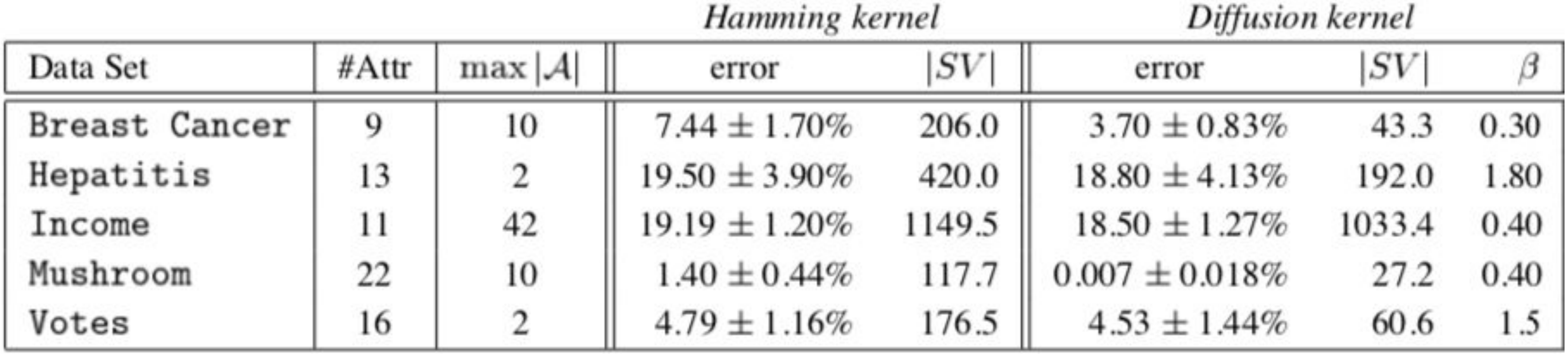
More generally, for alphabet A,

$$K(x,x') \propto \left(\frac{1-e^{|A|\beta}}{1+(|A|-1)e^{-|A|\beta}}\right)^{d(x,x')}$$

Using kernels for classification

- Goal: Compare with kernel methods for classifying categorical data
- Use a large margin classifier based on the voted perceptron
- Methods:
  - Diffusion kernel
  - Kernel based on hamming distance: $K_H(x, x') = n - \sum_{i=1}^{n} \delta(x_i, x'_i)$
- Continuous features were ignored

Results



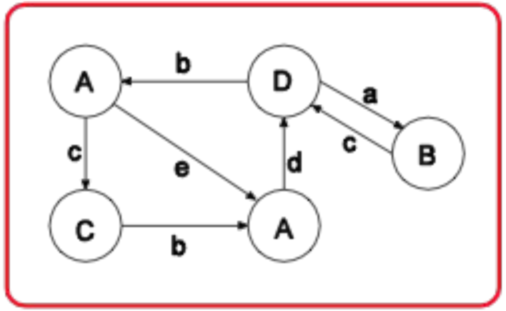
			Hamming kernel		Diffusion kernel		
Data Set	#Attr	max $\|\mathcal{A}\|$	error	$\|SV\|$	error	$\|SV\|$	β
Breast Cancer	9	10	7.44 ± 1.70%	206.0	3.70 ± 0.83%	43.3	0.30
Hepatitis	13	2	19.50 ± 3.90%	420.0	18.80 ± 4.13%	192.0	1.80
Income	11	42	19.19 ± 1.20%	1149.5	18.50 ± 1.27%	1033.4	0.40
Mushroom	22	10	1.40 ± 0.44%	117.7	0.007 ± 0.018%	27.2	0.40
Votes	16	2	4.79 ± 1.16%	176.5	4.53 ± 1.44%	60.6	1.5

Add some description of datasets

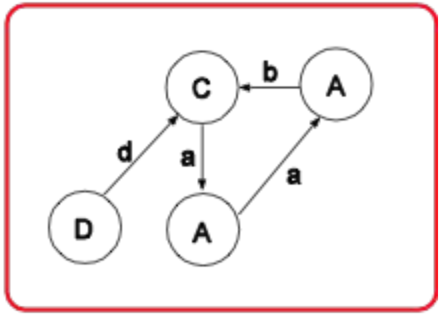
Graph Kernels

We now want to compare a set of graphs.

- Need to define kernel function K(G,G')
- Both vertex and edges can be labeled



G



G'

Generalized Random Walk Graph Kernels

Defined for a graph G = (V,E) that can

- Be directed/undirected
- Be weighted/unweighted
- Have (or not) edge labels

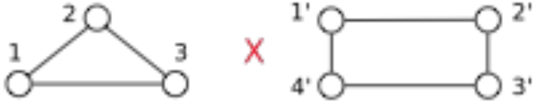
Two special cases:

- Random walk graph kernels
- Marginalized graph kernels

Generalized Random Walk Graph Kernels

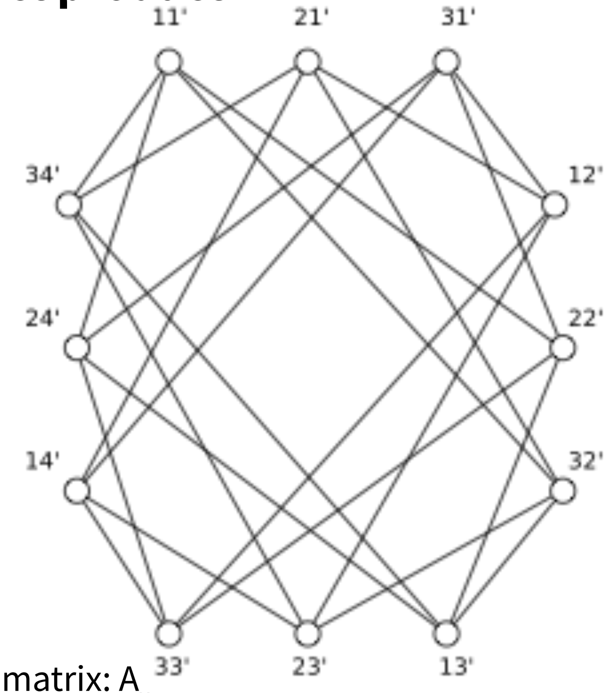
Idea: given a pair of graphs, perform random walks on both and count the number of matching walks

Two graphs



Adj. matrix: A and A'

Direct product



Adj. matrix: A_x

Random walks

Let

- p and p' be respec. the starting probabilities
- q and q' be respec. the starting probabilities

Hence

- $p_x = p \otimes p'$
- $q_x = q \otimes q'$

Weight matrix

$$W_\times = \Phi(X) \otimes \Phi(X').$$

- Continuous weights case:

  $W_x = A_x$
- Categorical weights case (finite set of size d):

$$W_\times = \sum_{l=1}^{d} {}^{l}A \otimes {}^{l}A'.$$

Kernel definition

$$k(G,G') := \sum_{k=0}^{\infty} \mu(k)\, q_{\times}^{\top} W_{\times}^{k} p_{\times}\,.$$

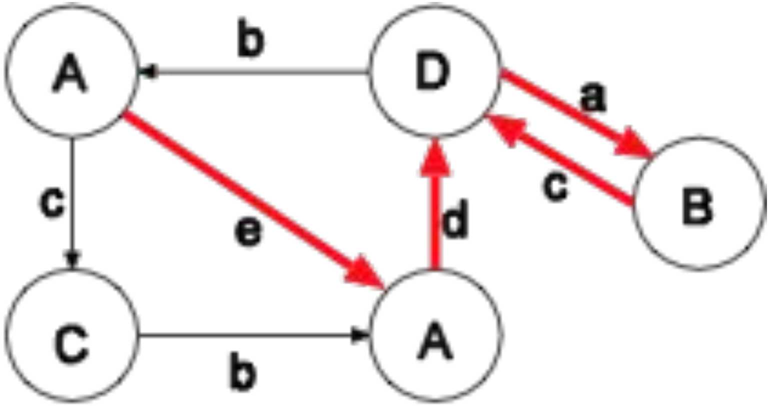
- Choice of μ(k) allows to (de-)emphasize walks of different lengths

Two special cases:

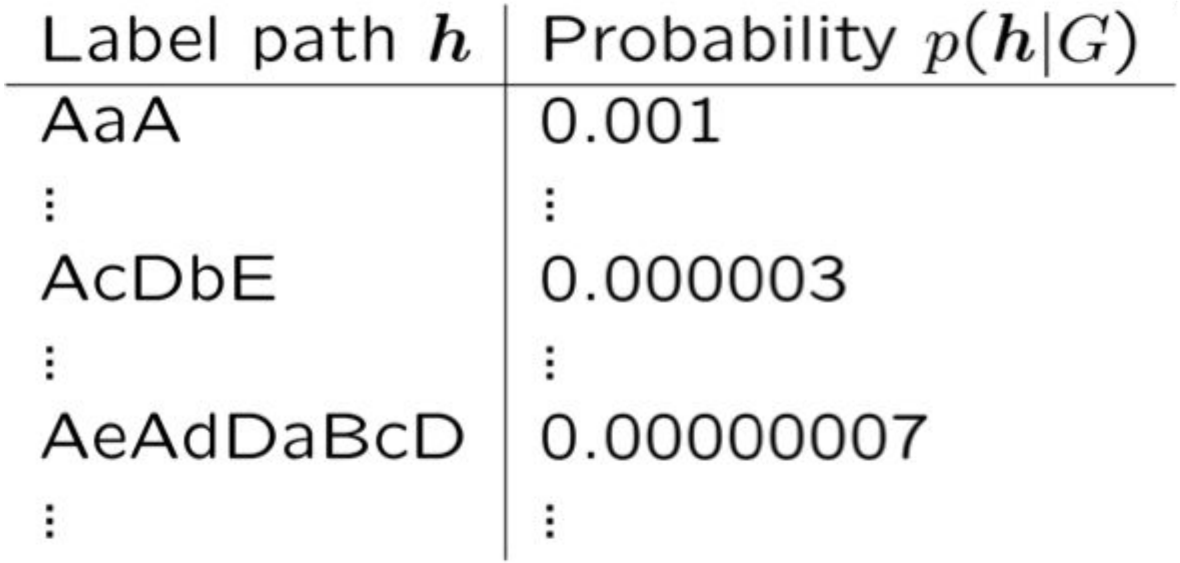
- Marginalized graph kernel (Kashima et al., 2004)
- Random walk graph kernel

Label path

- Sequence of node and edge labels
  H = (A, e, A, d, D, a, B, c, D)
- Generated by random walks
- Simplest case: uniform initial, transition and terminal probabilities



Path-probability vector



Label path $\boldsymbol{h}$	Probability $p(\boldsymbol{h}\|G)$
AaA	0.001
⋮	⋮
AcDbE	0.000003
⋮	⋮
AeAdDaBcD	0.00000007
⋮	⋮

Kernel definition

- Kernels for paths

$$K(\boldsymbol{h}, \boldsymbol{h}') = \begin{cases} 0 & (|\boldsymbol{h}| \neq |\boldsymbol{h}'|) \\ k_v(h_1, h_1')k_e(h_2, h_2') \cdots k_v(h_\ell, h_\ell') & (|\boldsymbol{h}| = |\boldsymbol{h}'|) \end{cases}$$

- Take expectation over all possible paths!
- Marginalized graph kernels

$$K(G, G') = \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p(\boldsymbol{h}|G)p(\boldsymbol{h}'|G')K(\boldsymbol{h}, \boldsymbol{h}')$$

In the "generalized" version...

They do not (cannot?) consider node labels

Path h = $(i_1, i_2, \ldots, i_t)$

$$p(h|G) := q_{i_{t+1}} \prod_{j=1}^{t} P_{i_j, i_{j+1}} p_{i_1}.$$

$$\kappa(h, h') := \prod_{i=1}^{t} \kappa(h_i, h'_i) = \prod_{i=1}^{t} \langle \hat{\phi}(h_i), \hat{\phi}(h'_i) \rangle$$

$$K(G, G') = \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p(\boldsymbol{h}|G) p(\boldsymbol{h}'|G') K(\boldsymbol{h}, \boldsymbol{h}')$$

In the "generalized" version...

They do not (cannot?) consider node labels

Path h = (i_1,i_2,...,i_t)

$$p(h|G) := q_{i_{t+1}} \prod_{j=1}^{t} P_{i_j,i_{j+1}} \, p_{i_1}.$$

$$\kappa(h,h') := \prod_{i=1}^{t} \kappa(h_i,h'_i) = \prod_{i=1}^{t} \langle \hat{\phi}(h_i), \hat{\phi}(h'_i) \rangle$$

$$K(G,G') = \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p(\boldsymbol{h}|G) p(\boldsymbol{h}'|G') K(\boldsymbol{h},\boldsymbol{h}')$$

Graph Kernel Applications

- Chemical Compounds (Mahe et al., 2005)
- Protein 3D structures (Borgwardt et al., 2005)
- RNA graphs (Karklin et al., 2005)
- Pedestrian detection
- Signal Processing

Concluding remarks

- Idea of measuring similarity between nodes via random walks is still very important
- Ideas from word2vec were very influential in area of graph representation
- Embedding techniques construct actual feature maps Φ and similarity is measured on that space
- Do graph embeddings build on graph kernels?

References

Kernels on graphs (between nodes of a single graph):
R. Kondor and J. Lafferty: Diffusion kernels on graphs and other discrete input spaces (ICML 2002) (winner of "test of time" award)
A. Smola and R. Kondor: Kernels and regularization on graphs (COLT 2003)

Graph kernels (between graphs):
Thomas Gartner, Peter A. Flach, and Stefan Wrobel: On graph kernels: Hardness results and efficient alternatives (COLT 2003)
Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schonauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels (ISMB 2005)

References

Learning on graphs:

R. Kondor and K. M. Borgwardt: The skew spectrum of graphs (ICML 2008)

R. Kondor, N. Shervashidze and K. M. Borgwardt: The graphlet spectrum (ICML 2009)

Survey:

S. V. N. Vishwanathan, N. N. Schraudolf, R. Kondor and K. M. Borgwardt: Graph kernels (Journal of Machine Learning Research 11, 2010)

Multi-scale structure of large graphs:

R. Kondor and H. Pan: The Multiscale Laplacian Graph Kernel (NIPS 2016)

References

Applications

R. Kondor and J.-P. Vert: Diffusion kernels in "Kernel Methods in Computational Biology" ed. B. Scholkopf, K. Tsuda and J.-P. Vert, (The MIT Press, 2004)

Hermansson, L., Kerola, T., Johansson, F., Jethava, V. and Dubhashi, D.. Entity disambiguation in anonymized graphs using graph kernels (ACM CIKM 2013)

References

Fast kernels:

Kang, U., Tong, H. and Sun, J.. Fast random walk graph kernel (SDM 2012)

Koutra, D., Vogelstein, J.T. and Faloutsos, C. Deltacon: A principled massive-graph similarity function (SDM 2013 and TKDD 2014?)

Hu, B., Lu, Z., Li, H. and Chen, Q.. Convolutional neural network architectures for matching natural language sentences (NIPS 2014)