http://alex.smola.org/talks/coltgraph2003.pdf

Kernels and Regularization on Discrete Domains

Alexander J. Smola and Risi I. Kondor
Alex.Smola@anu.edu.au and risi@cs.columbia.edu

Machine Learning Program
Australian National University and National ICT Australia

Department of Computer Science
Columbia University

Outline

- Learning Problem
- The Graph Laplacian
  - Definition and Properties
  - Invariance Theorem
- Regularization and Greens Functions on Graphs
  - Regularization by the Graph Laplacian
  - Kernels
  - Connections to Clustering
- Approximate and Fast Computation
  - Products of Graphs
  - Iterative Expansions and Polynomial Approximation
- Summary and Outlook

Learning Problem

Estimation Problem

Given some observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, find estimator $f : \mathcal{X} \to \mathcal{Y}$ which minimizes some cost of misprediction.

Specifically, f is a member of a Reproducing Kernel Hilbert Space, so we need a kernel $k(x, x')$.

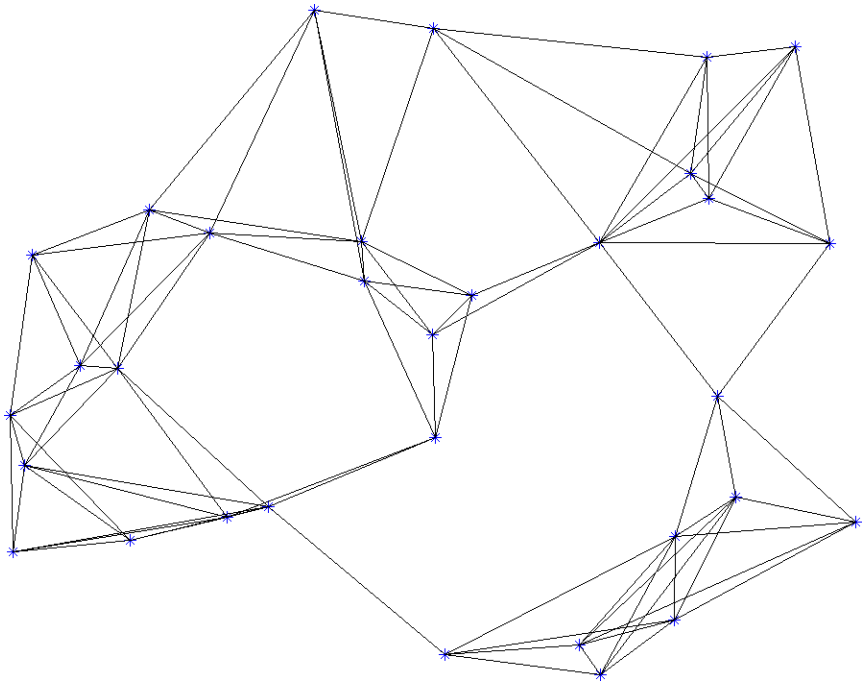
Unreal Data:

- Discrete data categorical variables, e.g. `(English, high school, butcher, unemployed)`
- Similarity between pairs of observations, e.g. set of k-nearest neighbours.
- Web pages
- Regulatory networks

Problem

We need a measure of smoothness on functions f, defined on $\mathcal{X}$, where $\mathcal{X}$ is a **discrete** set.

Graphs



Graph

Define $G(V, E)$ as a set of vertices V and edges E.

Connectivity Matrix

$W \in \mathbb{R}^{|V|^2}$ where $W_{ij} = 1$ if i, j share an edge and 0 otherwise.

Also $W \in [0, \infty)$.

Random Walk

From vertex i to j with probability

$$p(j|i) = \frac{W_{ij}}{\sum_l W_{il}} = \frac{W_{ij}}{D_{ii}}$$

Graph Laplacian

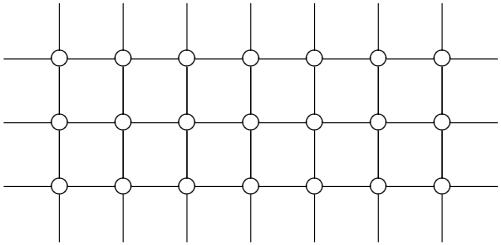
Smoothness on Graph

A possible criterion for smooth functions is that variations between adjacent values should be small:

$$\sum_{i\sim j}(f_i - f_j)^2 = 2\sum_i f_i^2 D_i - 2\sum_{i\sim j} f_i f_j = 2f^\top \underbrace{(D-W)}_{:=L} f.$$

where $D_i = \sum_j W_{ij}$ is the diagonal normalization.

Special Case: Lattice in 2D



For regular lattices, $-L$ is the discretization of the continuous Laplace operator

$$\Delta = \sum_i \partial_{x_i}^2.$$

Normalized Graph Laplacian

We rescale L by D to obtain $\tilde{L} := \mathbf{1} - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.

Note that $\mathbf{1} \succeq \frac{1}{2}\tilde{L} \succeq 0$.

Invariance Theorem

Theorem

Denote by $L \in \mathbb{R}^{n^2}$ a symmetric matrix, given as a linear permutation invariant function of the adjacency matrix W, i.e. $L = \mathfrak{T}[W]$ with

$$\Pi_\pi^\top \mathfrak{T}[W] \Pi_\pi = \mathfrak{T}\left[\Pi_\pi^\top W \Pi_\pi\right] \text{ for all } \pi \in S_n$$

Then L is related to W by a linear combination of the following operations:

- identity
- row/column sums and overall sum
- row/column sum restricted to the diagonal of L

Consequence

This essentially only leaves the (normalized) graph Laplacian. An analogous result exists for the Laplace Operator in $\mathbb{R}^n$ with respect to the Galilei group.

Proof Idea

Specifying the Operator $\mathcal{T}$

$$L_{i_1 i_2} = \mathfrak{T}[W]_{i_1 i_2} := \sum_{i_3, i_4}^{n} T_{i_1 i_2 i_3 i_4} W_{i_3 i_4}$$

Permutation invariance implies

$$T_{\pi(i_1)\pi(i_2)\pi(i_3)\pi(i_4)} = T_{i_1 i_2 i_3 i_4} \text{for any } \pi \in S_n.$$

Picking matching terms

For every matching set of indices, the corresponding entries in the tensor T have to agree, e.g. if the first and second index ($i_1 = i_2$) in T agree, then they also agree in $T_{\pi(i_1)\pi(i_2)\pi(i_3)\pi(i_4)}$, that is $\pi(i_1) = \pi(i_2)$.

Matching

Interpret the remaining terms of T as per theorem.

Regularization and Kernels

Regularization on f

Given $f \in \mathbb{R}^n$ we want some matrix $M \succeq 0$ to define the regularizer $f^\top M f$.

Self-Consistency Condition

In an RKHS we have the condition

$$\langle k(x, \cdot), Mk(x', \cdot) \rangle = k(x, x')$$

In matrix notation this can be rewritten as

$$KMK = K \text{ and therefore } K = M^\dagger$$

Here $M^\dagger$ is the pseudoinverse of M.

"Kernel Expansion"

For the expansion $f = K\alpha$ we have

$$f^\top M f = \alpha^\top K M K \alpha = \alpha^\top K \alpha$$

Using the Laplacian

Designing M from $L, \tilde{L}$

We want to penalize quickly varying functions on the graph more severely. The eigensystem of L or $\tilde{L}$ is a good guess for that.

Eigenvectors with small eigenvalues split the graph into large coherent clusters (e.g. Fiedler vector).

Analogy from Regularization with Laplace Operators

$$\langle f, Mf \rangle = \left\langle f, \exp\left(\frac{\sigma^2}{2}\Delta\right) f \right\rangle$$

yields Gaussian kernels $k(x, x') = \exp(-\frac{1}{2\sigma^2}\|x - x'\|^2)$.

$$\langle f, Mf \rangle = \langle f, \exp(\sigma L) f \rangle$$

yields Diffusion kernels $K = \exp(-\sigma L)$.

Eigenvalue Remapping

General Connection

Use monotonic $r(\lambda)$ to define $M = r(L)$. K is then given by $K = r^{-1}(L)$. **Big Gain: $r^{-1}(\lambda)$ may be cheap.**

Examples of $r(\lambda)$

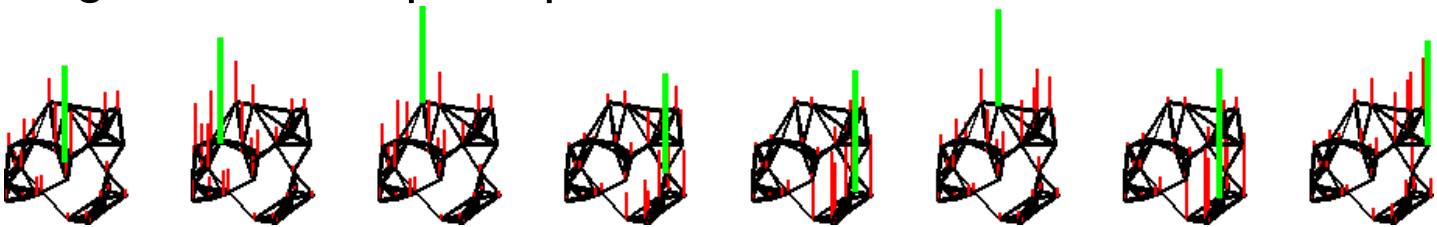
$r(\lambda) = \mathbf{1} + \sigma\lambda$	(Regularized Laplacian)
$r(\lambda) = \exp(\sigma\lambda)$	(Diffusion Process)
$r(\lambda) = (a\mathbf{1} - \lambda)^{-p}$ with $a \geq 2$	(p-Step Random Walk)
$r(\lambda) = (\cos \lambda\pi/4)^{-1}$	(Inverse Cosine)

Examples of $K = r^{-1}(L)$

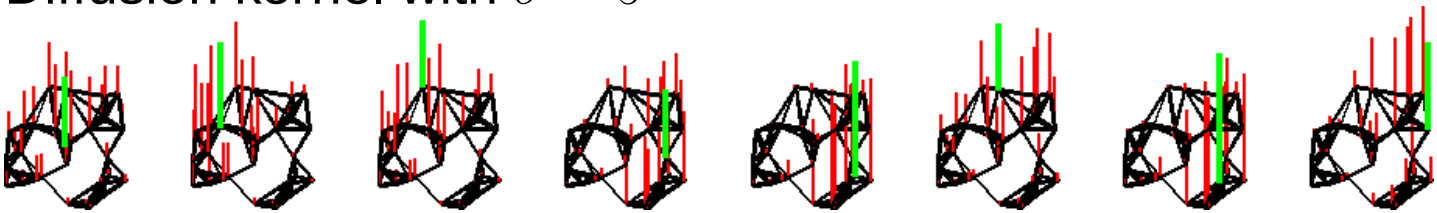
$K = (\mathbf{1} + \sigma L)^{-1}$	(Regularized Laplacian)
$K = (\vec{1}\vec{1}^\top + \sigma L)^{-1}$	("Google)"
$K = \exp(-\sigma L)$	(Diffusion Process)
$K = (a\mathbf{1} - L)^p$ with $a \geq 2$	(p-Step Random Walk)

Examples

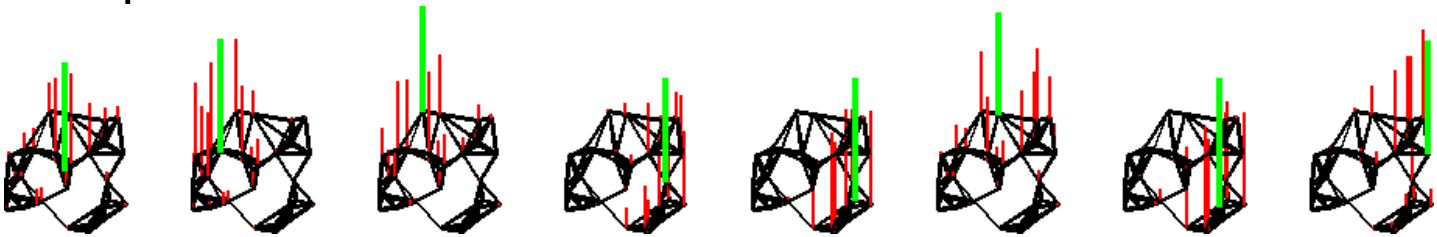
Regularized Graph Laplacian



Diffusion kernel with $\sigma = 5$



4-step Random Walk



Connections to Clustering

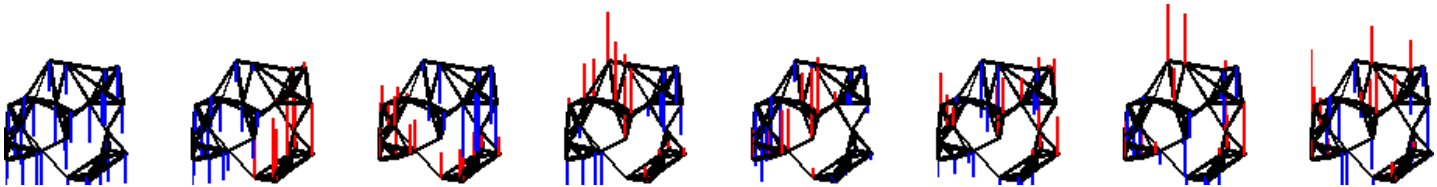
Eigenvectors

In spectral clustering one decomposes $G(V, E)$ according to the smallest eigenvectors of the Graph Laplacian:

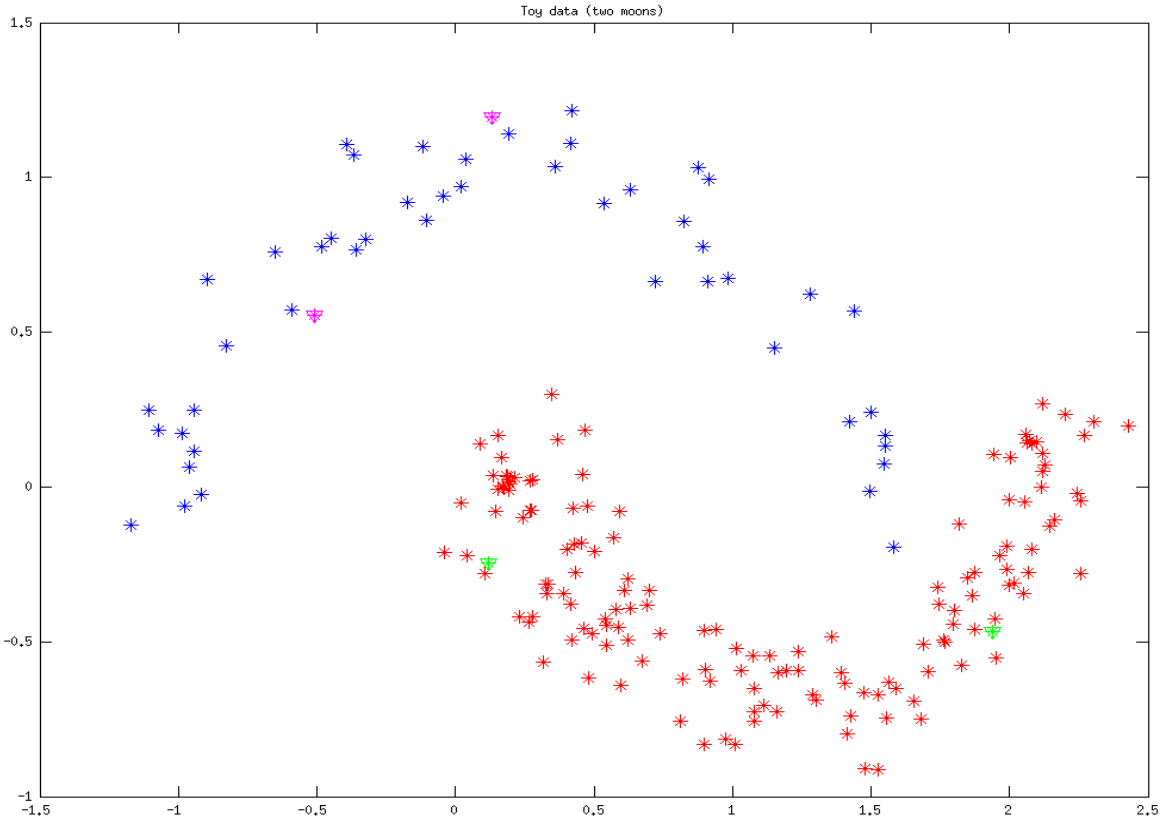
- Small eigenvalues/eigenvectors correspond to large coherent parts of the graph.
- Large eigenvalues/eigenvectors yield incoherent components.

Kernels

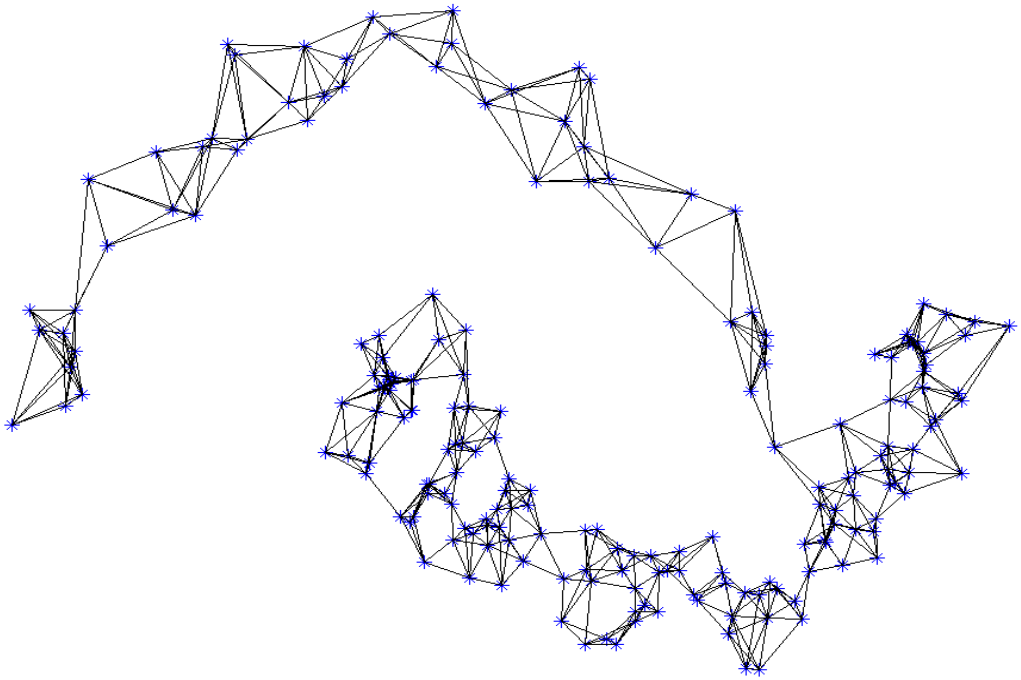
The order of the eigenvalues is **reversed**. So Kernel-PCA on a Graph-Kernel finds **small eigenvectors** of the Graph Laplacian.



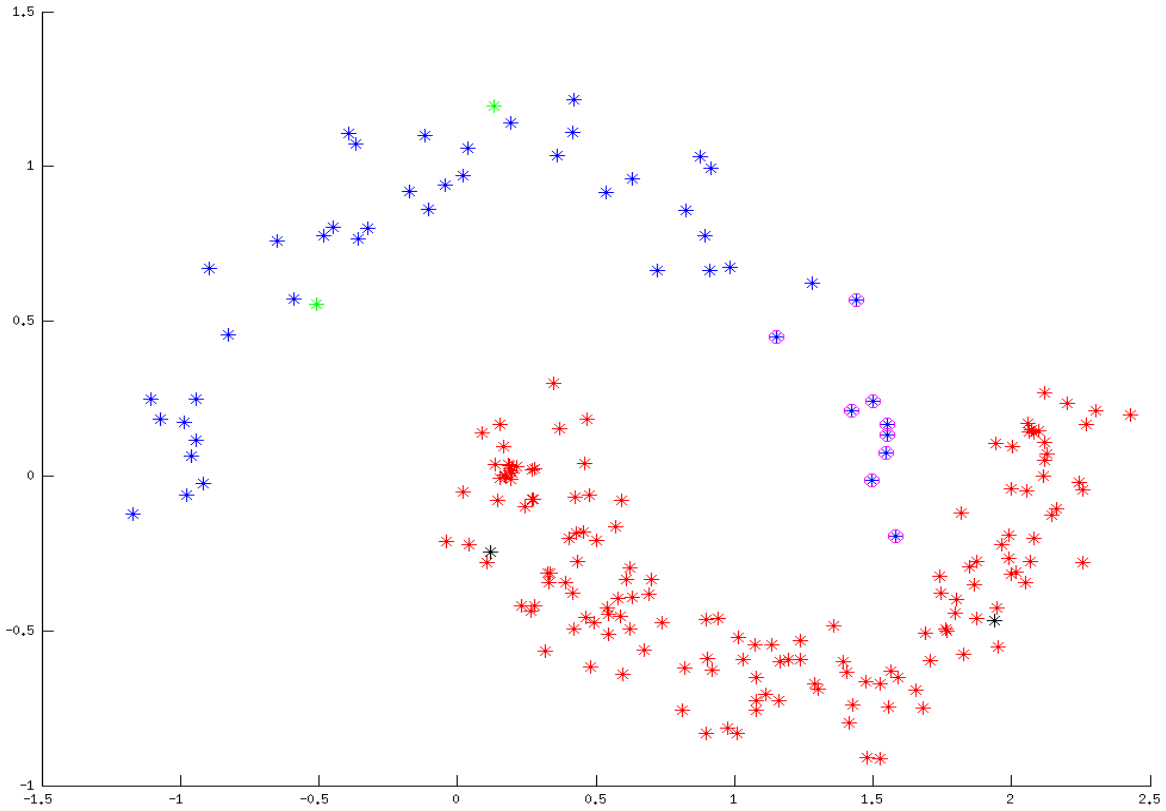
Two Moons



Nearest Neighbor Graph



Inverse Graph Laplacian



Products of Graphs

Motivation

Often graphs are composed of simple parts, e.g. as products of simpler graphs. Example: hypercubes.

Goal

Compute K without paying the price of the larger graph.

Spectral Properties

For regular graphs, we can simply **multiply the eigenvalues** of the factors of the graph.

$$\lambda_{j,l}^{\text{fact}} = \frac{d}{d+d'}\lambda_j + \frac{d'}{d+d'}\lambda'_l$$

Likewise, the eigenvectors are the **cartesian product** of the eigenvectors of the factors:

$$e_{(i,i')}^{j,l} = e_i^j e_{i'}'^l$$

Product Tricks

Analytic Expressions

$$\exp(-\beta(a+b)) = \exp(-\beta a)\exp(-\beta b)$$

$$(A-(a+b))^p = \sum_{n=0}^{p} \binom{p}{n} \left(\frac{A}{2}-a\right)^n \left(\frac{A}{2}-b\right)^{p-n}$$

So for diffusion processes we can simply take the product of the kernels over the factors.

Brute Force Theorem If we can solve parts more cheaply, we can compute the overall kernel by

$$K_{(j,j'),(l,l')} = \frac{1}{2\pi i}\int_C K_\alpha(j,l)G'_{-\alpha}(j',l')d\alpha = \sum_v K_{\lambda_v}(j,l)e^v_{j'}e^v_{l'}$$

Open Problem

What to do if we do not have regular graphs.

Outlook and Summary

What we have

- Extending regularization operators to discrete domain.
- Connections to spectral clustering.
- Extensions of the diffusion kernel setting.

To Do

- Extensions of the regularization framework to directed graphs (e.g. for Smola & Vishwanathan).
- Stability results for vertex/edge removal.
- Approximate computation for large graphs and scale-free networks.

We are hiring. For details see www.nicta.com.au or Alex.Smola@anu.edu.au