

Graph Structure, Complexity and Learning

Edwin Hancock

Department of Computer Science

University of York



THE ROYAL
SOCIETY

Supported by a Royal Society
Wolfson Research Merit Award

Royal Society

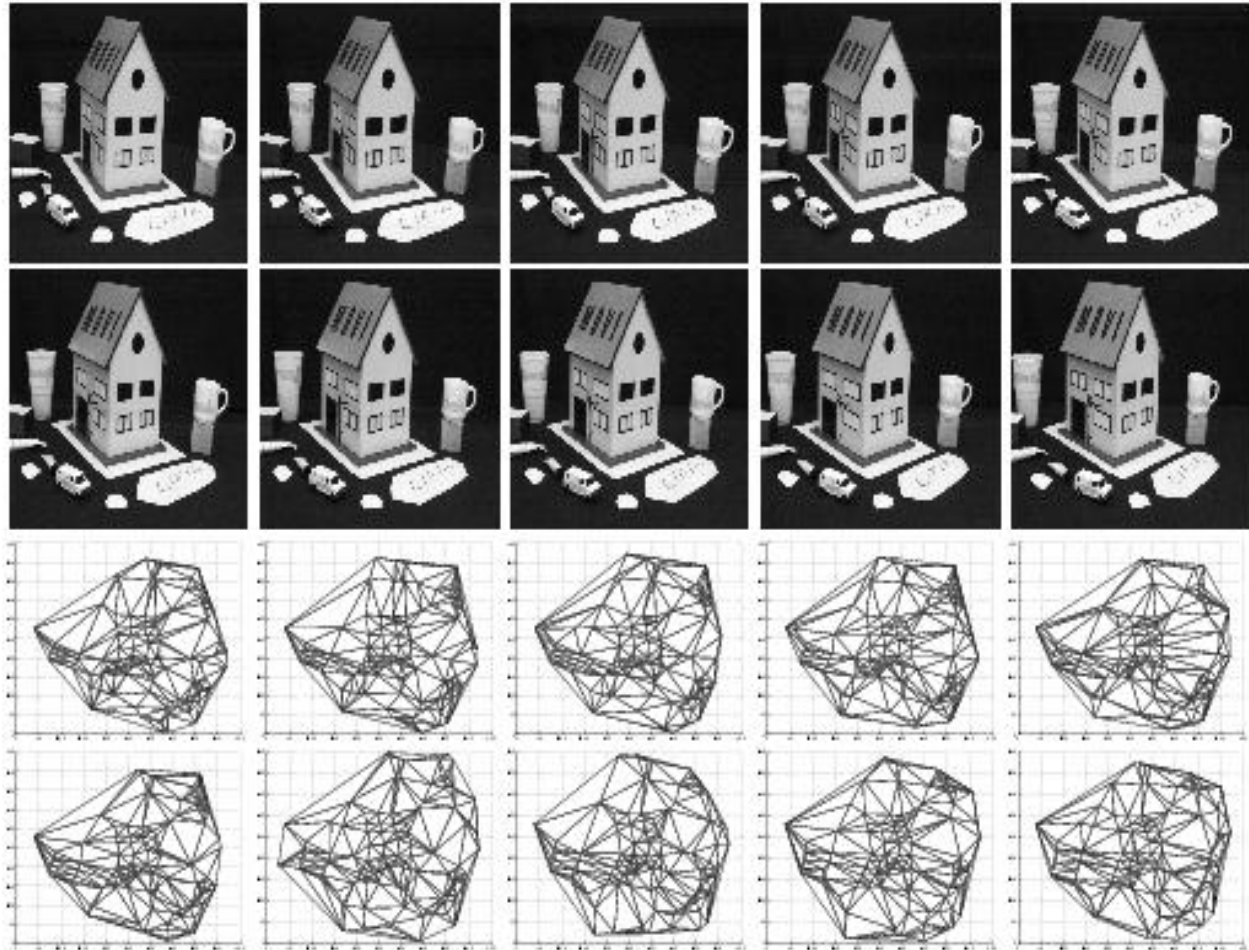
- Chicheley Hall meeting May 2012



- Video proceedings:

http://videolectures.net/complexnetworks2012_london/

Structural Variations



Protein-Protein Interaction Networks

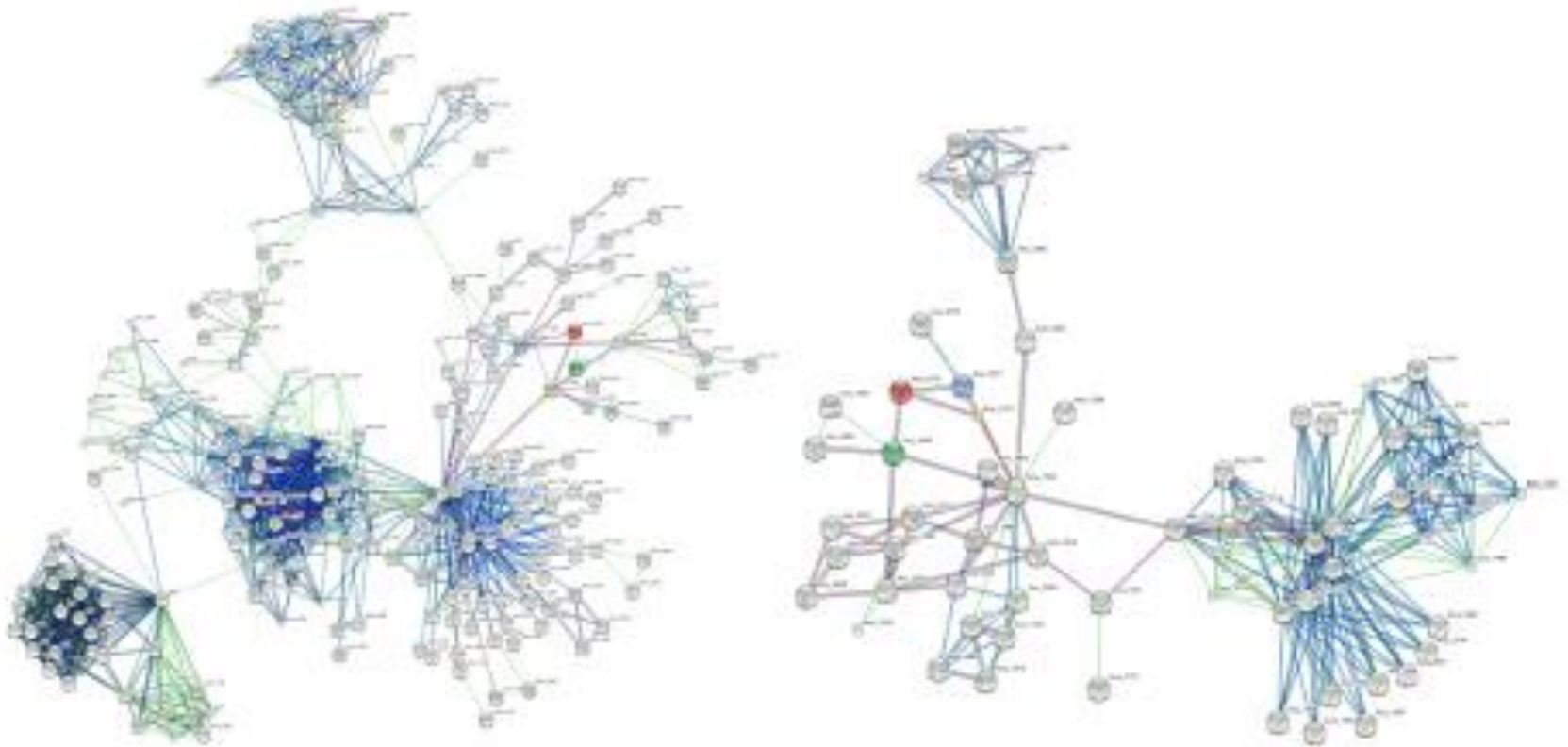


Figure: STRING Protein-Protein Interaction Networks

Graph data

- Problems based on graphs arise in areas such as language processing, proteomics/chemoinformatics, data mining, computer vision and complex systems.
- Relatively little methodology available, and vectorial methods from statistical machine learning not easily applied since there is no canonical ordering of the nodes in a graph.
- Can make considerable progress if we develop permutation invariant characterisations of variations in graph structure.

Graph data

- Problems based on graphs arise in areas such as language processing, proteomics/chemoinformatics, data mining, computer vision and complex systems.
- Relatively little methodology available, and vectorial methods from statistical machine learning not easily applied since there is no canonical ordering of the nodes in a graph.
- Can make considerable progress if we develop permutation invariant characterisations of variations in graph structure.

Graph data

- Problems based on graphs arise in areas such as language processing, proteomics/chemoinformatics, data mining, computer vision and complex systems.
- Relatively little methodology available, and vectorial methods from statistical machine learning not easily applied since there is no canonical ordering of the nodes in a graph.
- Can make considerable progress if we develop permutation invariant characterisations of variations in graph structure.

Characterising graphs

- **Topological:** e.g. average degree, degree distribution, edge-density, diameter, cycle frequencies etc.
- **Spectral or algebraic:** use eigenvalues of adjacency matrix or Laplacian, or equivalently the co-efficients of characteristic polynomial.
- **Complexity:** use information theoretic measures of structure (e.g. Shannon entropy).

Complex systems

- **Spatial and topological indices**: node degree stats; edge density;
- **Communicability**: communities, measures of centrality, separation, etc. (Baribasi, Watts and Strogatz, Estrada).
- **Processes on graphs**: Markov process, Ising models, random walks, searchability (Kleinberg).

Links explored in this talk

- Structure: discriminate between graphs on the basis of their detailed structure.
- Complexity: determine whether different non-isomorphic structures are if similar or different intrinsic complexity.
- Learning: learn generative model of structure that gives minimum complexity description of training data (MDL).

Structure

Graph spectra and random walks

Use spectrum of Laplacian matrix
to compute hitting and commute
times for random walk on a graph

Laplacian Matrix

- Weighted adjacency matrix

$$W(u, v) = \begin{cases} w(u, v) & (u, v) \in E \\ 0 & \textit{otherwise} \end{cases}$$

- Degree matrix

$$D(u, u) = \sum_{v \in V} W(u, v)$$

- Laplacian matrix

$$L = D - W$$

Laplacian spectrum

- Spectral Decomposition of Laplacian

$$L = \Phi \Lambda \Phi^T = \sum_k \lambda_k \phi_k \phi_k^T$$

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{|V|}$$

$$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{|V|})$$

$$\Phi = (\phi_1 \mid \dots \mid \phi_{|V|})$$

- Element-wise

$$L(u, v) = \sum_k \lambda_k \phi_k(u) \phi_k(v)$$

Properties of the Laplacian

- Eigenvalues are positive and smallest eigenvalue is zero

$$0 = \lambda_1 < \lambda_2 < \dots < \lambda_{|V|}$$

- Multiplicity of zero eigenvalue is number connected components of graph.
- Zero eigenvalue is associated with all-ones vector.
- Eigenvector associated with the second smallest eigenvalue is Fiedler vector.

Continuous time random walk

Heat Kernels

- Solution of heat equation and measures information flow across edges of graph with time:

$$\frac{\partial h_t}{\partial t} = -Lh_t$$

$$L = D - W = \Phi \Lambda \Phi^T$$

- Solution found by exponentiating Laplacian eigensystem

$$h_t = \sum_k \exp[-\lambda_k t] \phi_k \phi_k^T = \Phi \exp[-\Lambda t] \Phi^T$$

Heat kernel and random walk

- State vector of continuous time random walk satisfies the differential equation

$$\frac{\partial p_t}{\partial t} = -Lp_t$$

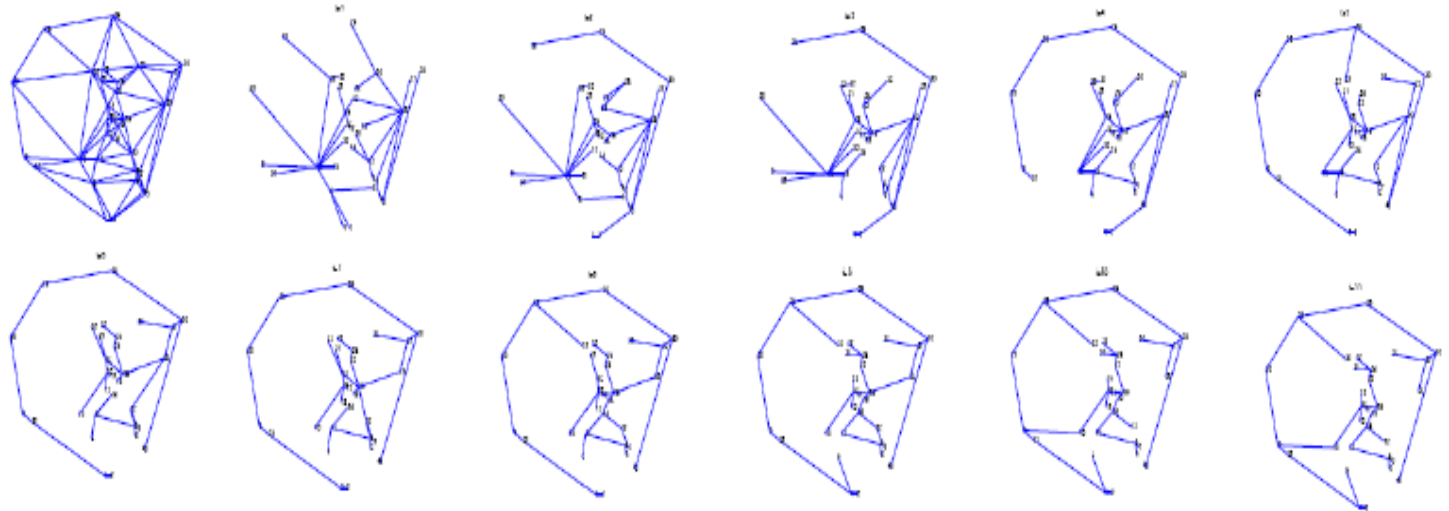
- Solution

$$p_t = \exp[-Lt]p_0 = h_t p_0$$

Example.

Graph shows spanning tree of heat-kernel. Here weights of graph are elements of heat kernel. As t increases, then spanning tree evolves from a tree rooted near centre of graph to a string (with ligatures).

Low t behaviour dominated by Laplacian, high t behaviour dominated by Fiedler-vector.



Moments of the heat-kernel trace

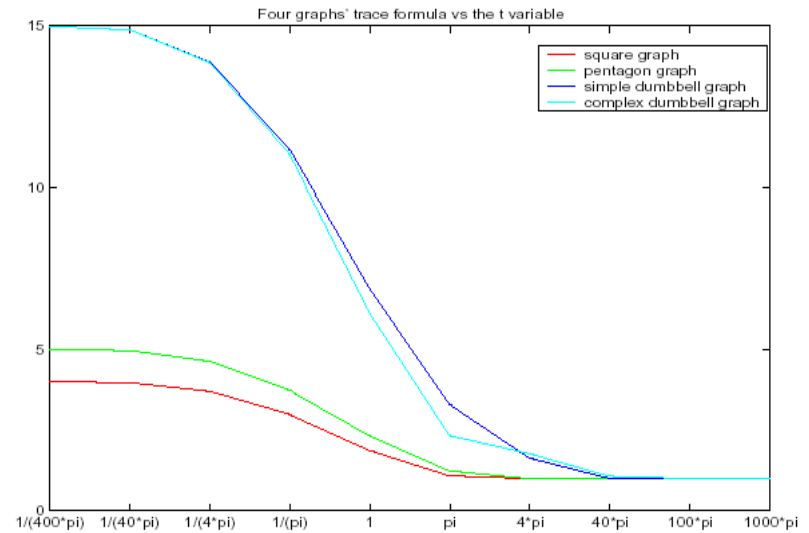
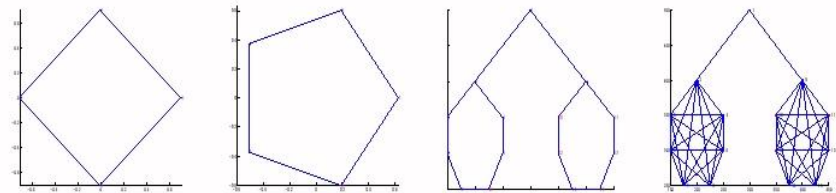
....can we characterise graph by
the shape of its heat-kernel trace
function?

Heat Kernel Trace

$$\text{Tr}[h_t] = \sum_i \exp[-\lambda_i t]$$

Shape of heat-kernel distinguishes graphs... can we characterise its shape using moments

Trace



Time (t)->

Heat Kernel Trace

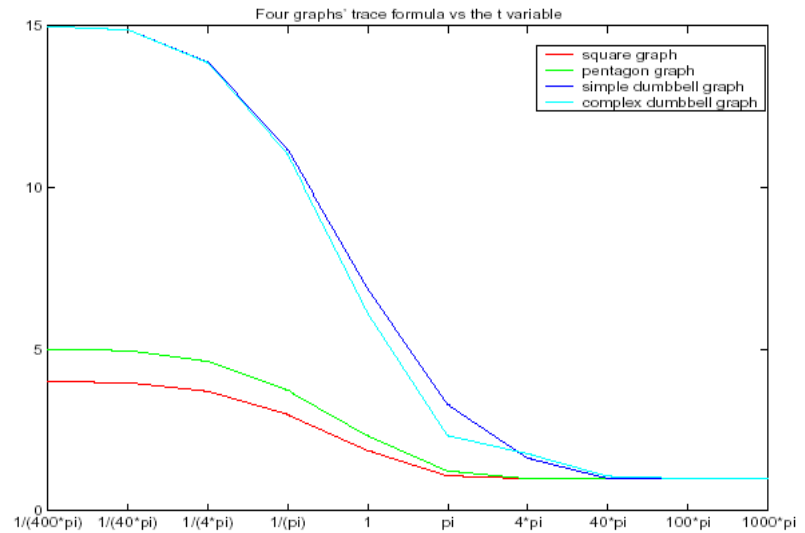
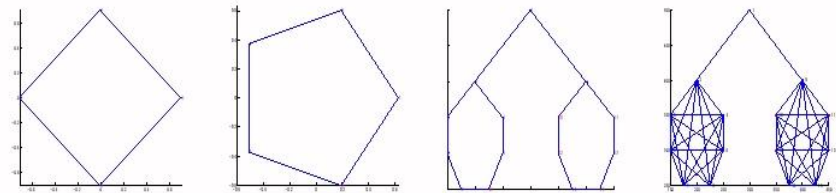
$$\text{Tr}[h_t] = \sum_i \exp[-\lambda_i t]$$

Shape of heat-kernel distinguishes graphs...can we characterise its shape using moments

Trace

Use moments of heat kernel trace:

$$\mu(s) = \int_0^\infty t^{s-1} \text{Tr}[h(t)] dt$$



Time (t)->

Rosenberg Zeta function

- Definition of zeta function

$$\zeta(s) = \sum_{\lambda_k \neq 0} (\lambda_k)^{-s}$$

Heat-kernel moments

- Mellin transform

$$\lambda_i^{-s} = \frac{1}{\Gamma(s)} \int_0^\infty t^{s-1} \exp[-\lambda_i t] dt$$

$$\Gamma(s) = \int_0^\infty t^{s-1} \exp[-t] dt$$

- Trace and number of connected components

$$\text{Tr}[h_t] = C + \sum_{\lambda_i \neq 0} \exp[-\lambda_i t]$$

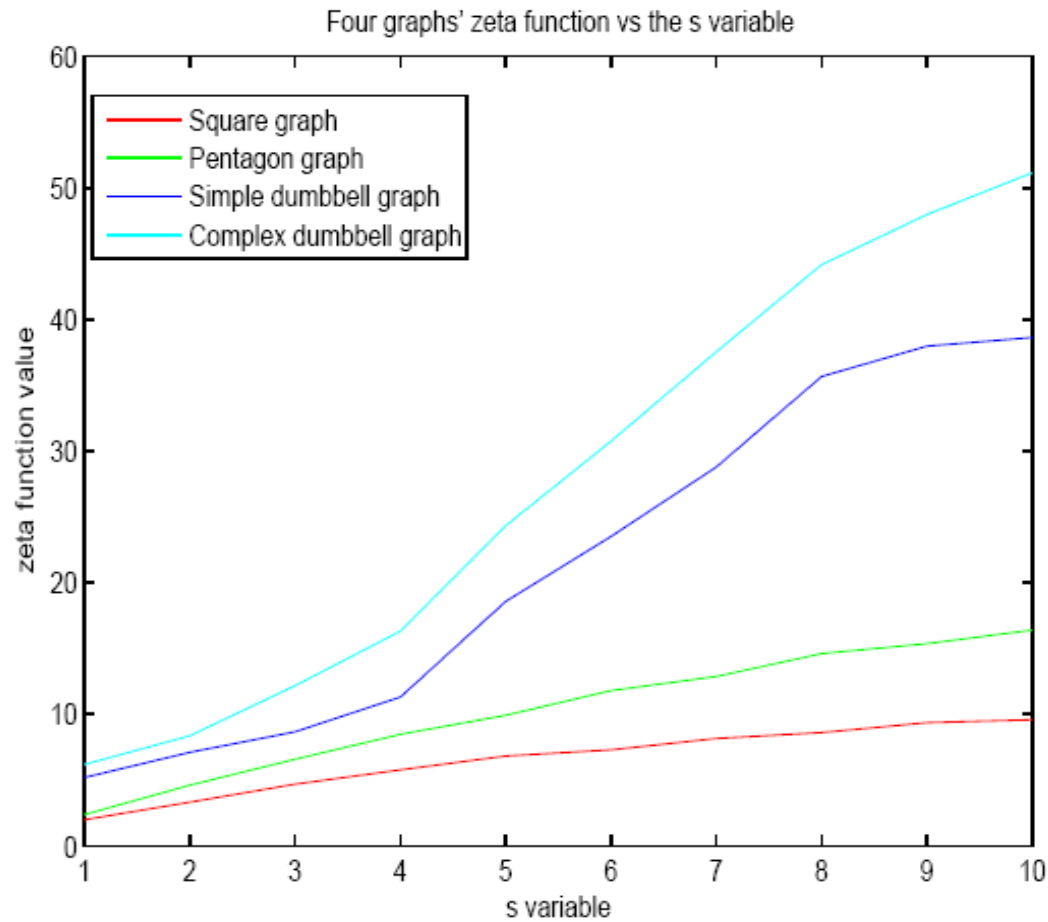
C is multiplicity of zero eigenvalue or number of connected components in graph.

- Zeta function

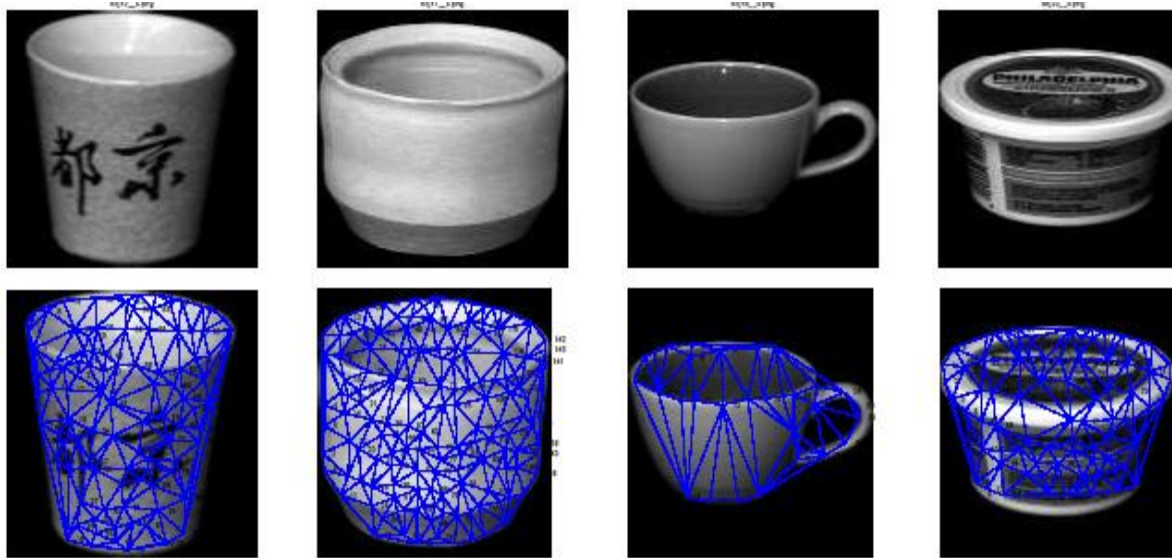
$$\zeta(s) = \sum_{\lambda_i \neq 0} \lambda_i^{-s} = \frac{1}{\Gamma(s)} \int_0^\infty t^{s-1} [\text{Tr}[h_t] - C] dt$$

Zeta-function is related to moments of heat-kernel trace.

Zeta-function behavior



Objects



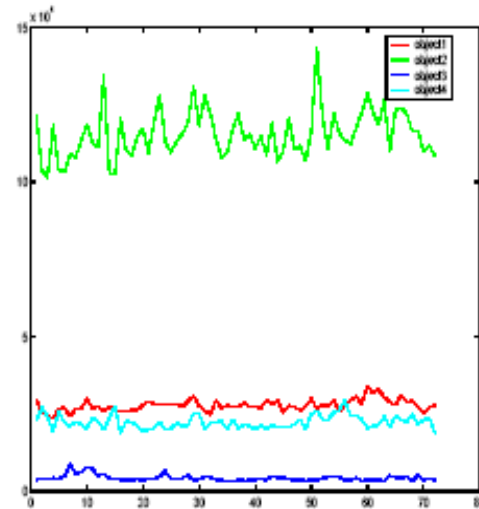
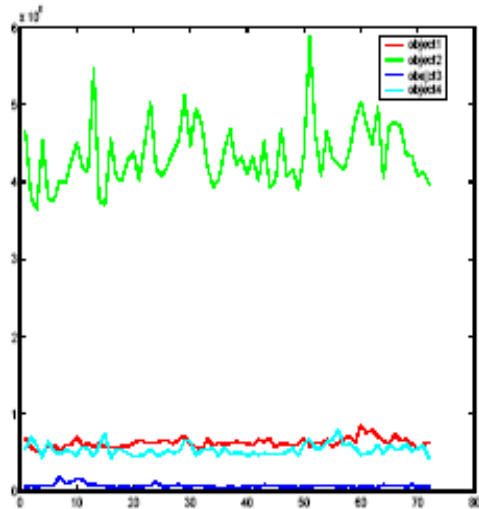
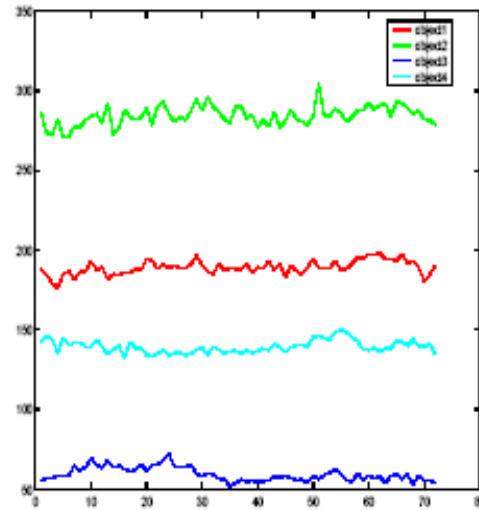
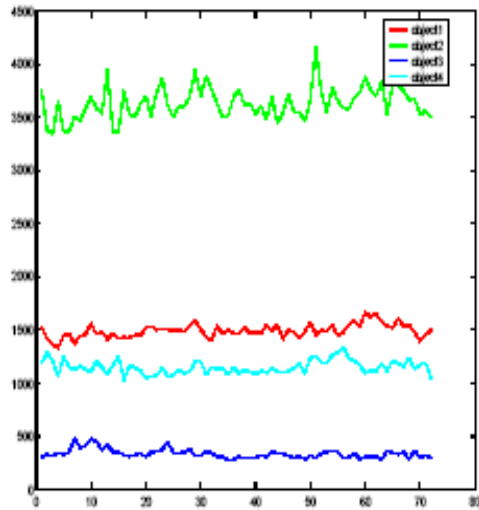
72 views of each object taken in 5 degree intervals as camera moves in circle around object.

Feature points extracted using corner detector.

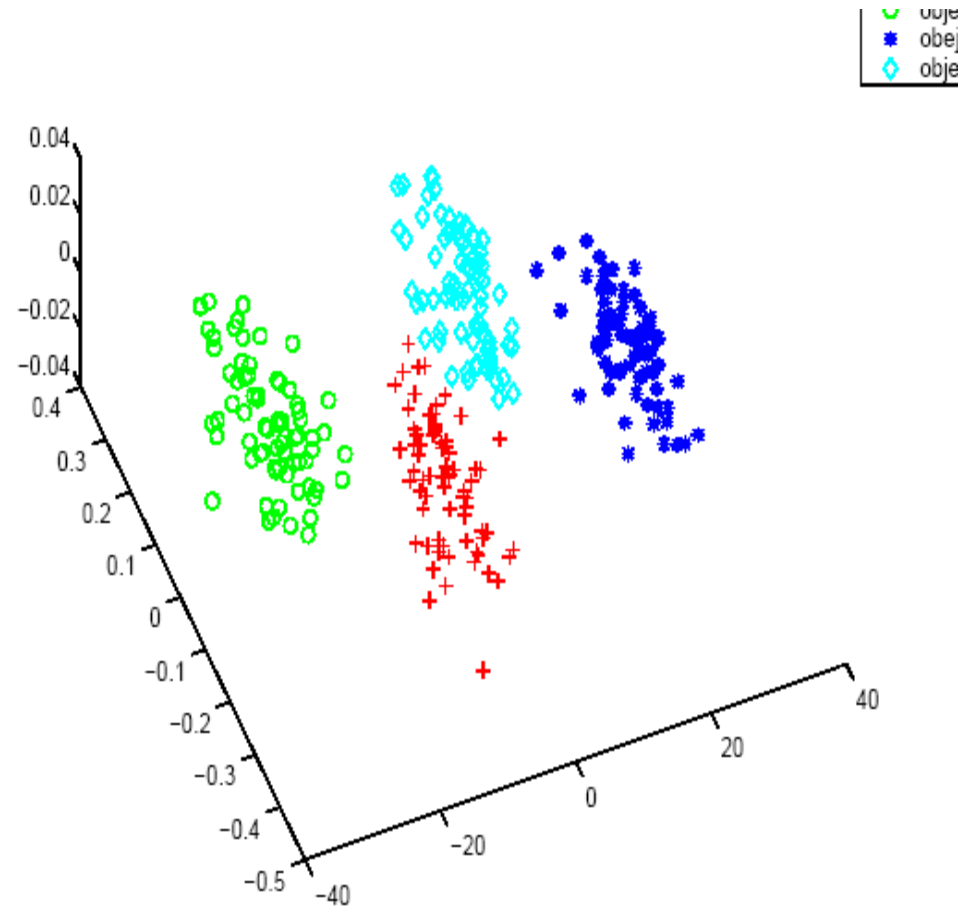
Construct Voronoi tessellation image plane using corner points as seeds.

Delaunay graph is region adjacency graph for Voronoi regions.

Heat kernel moments (zeta(s), s=1,2,3,4)



PCA using zeta(s), s=1,2,3,4)



Zeta function derivative

- Zeta function in terms of natural exponential

$$\zeta(s) = \sum_{\lambda_k \neq 0} (\lambda_k)^{-s} = \sum_{\lambda_k \neq 0} \exp[-s \ln \lambda_k]$$

- Derivative

$$\zeta'(s) = - \sum_{\lambda_k \neq 0} \ln \lambda_k \exp[-s \ln \lambda_k]$$

- Derivative at origin

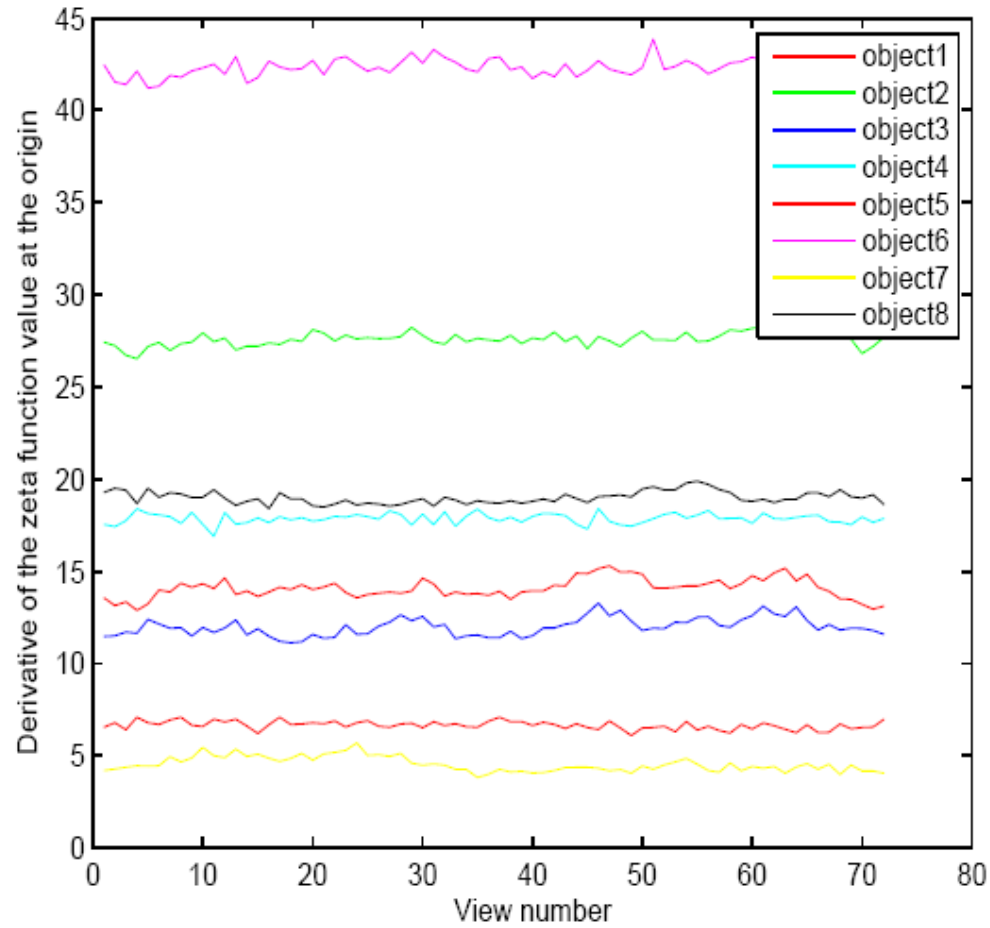
$$\zeta'(0) = - \sum_{\lambda_k \neq 0} \ln \lambda_k = \ln \frac{1}{\prod_{\lambda_k \neq 0} \lambda_k}$$

Meaning

- Number of spanning trees in graph

$$\tau(G) = \frac{\prod_{u \in V} d_u}{\sum_{u \in V} d_u} \exp[-\zeta'(0)]$$

COIL



Deeper probes of structure

Ihara zeta function

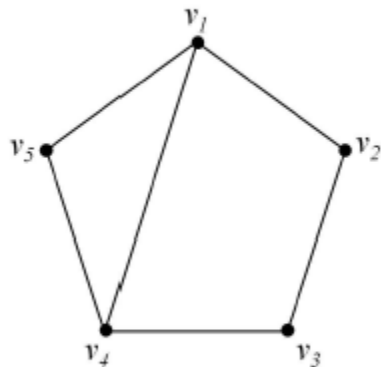
Zeta functions

- Used in number theory to characterise distribution of prime numbers.
- Can be extended to graphs by replacing notion of prime number with that of a prime cycle.

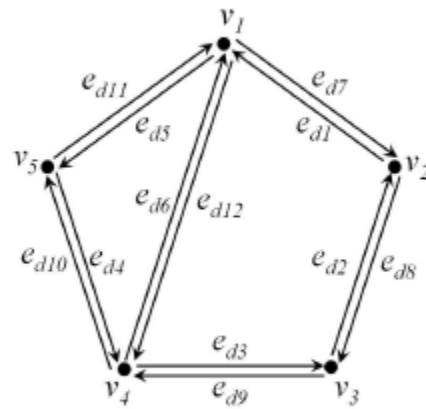
Ihara Zeta function

- Determined by distribution of prime cycles.
- Transform graph to oriented line graph (OLG) with edges as nodes and edges indicating incidence at a common vertex.
- Zeta function is reciprocal of characteristic polynomial for OLG adjacency matrix.
- Coefficients of polynomial determined by eigenvalues of OLG adjacency matrix.
- Powers of OLG adjacency matrix give prime cycle frequencies.

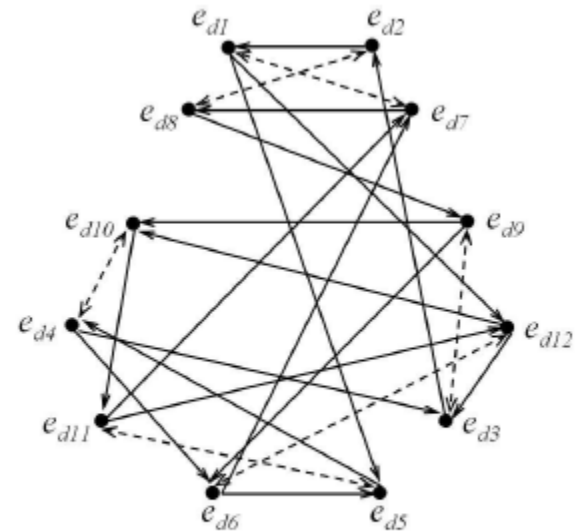
Oriented Line Graph



(a) Original Graph



(b) Digraph



(c) Oriented Line Graph

Ihara Zeta Function

- Ihara Zeta Function for a **graph** $G(V, E)$
 - Defined over **prime cycles** of graph

$$Z_G(u) = \prod_{p \in P} (1 - u^{|p|})^{-1}$$

- Rational expression in terms of **characteristic polynomial** of oriented line-graph

$$Z_G(u) = (1 - u^2)^{\chi(G)} \det (\mathbf{I}_{|V(G)|} - u\mathbf{A} + u^2\mathbf{Q})^{-1}$$

A is adjacency matrix of line digraph

Q = D-I (degree matrix minus identity)

Characteristic Polynomials from IZF

- Perron-Frobenius operator is the adjacency matrix \mathbf{T}_H of the oriented line graph

- Determinant Expression of IZF

$$\begin{aligned}\zeta_H(u) &= \det(\mathbf{I}_H - u\mathbf{T}_H)^{-1} \\ &= (c_0 + c_1u + \cdots + c_{M-1}u^{M-1} + c_Mu^M)^{-1}\end{aligned}$$

- Each coefficient, i.e. Ihara coefficient, can be derived from the elementary symmetric polynomials of the eigenvalue set $\{\lambda_1, \lambda_2, \lambda_3 \dots\}$

$$c_r = (-1)^r \sum_{k_1 < k_2 < \dots < k_r} \lambda_{k_1} \lambda_{k_2} \dots \lambda_{k_r}$$

- Pattern Vector in terms of $\vec{v} = [c_{r1} \ c_{r2} \ \dots \ c_{rN}]^T$

Analysis of determinant

- From matrix logs

$$\zeta(s) = \frac{1}{\det[I - Ts]} = \exp\left[\sum_{k>1} \text{Tr}[T^k] \frac{s^k}{k}\right]$$

- $\text{Tr}[T^k]$ is symmetric polynomial of eigenvalues of T

$$\text{Tr}[T^1] = \lambda_1 + \dots + \lambda_N$$

$$\text{Tr}[T^2] = \lambda_1^2 + \lambda_1\lambda_2 + \dots + \lambda_N^2$$

.....

$$\text{Tr}[T^N] = \lambda_1\lambda_2\dots\lambda_N$$

Distribution of prime cycles

- Frequency distribution for cycles of length l

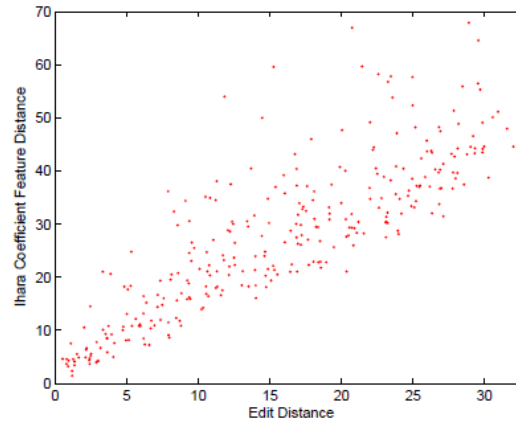
$$s \frac{d}{ds} \ln \zeta(s) = \sum_l N_l s^l$$

- Cycle frequencies

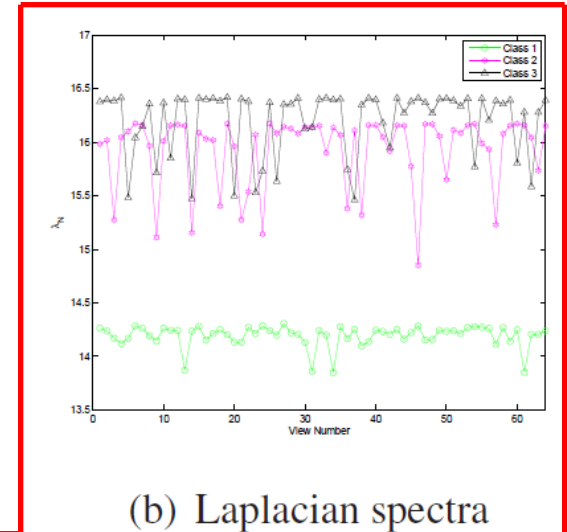
$$N_l = \frac{1}{(l-1)!} \frac{d^l}{ds^l} \ln \zeta(s) \Big|_{s=0} = \text{Tr}[T^l]$$

Experiments: Edge-weighted Graphs

Feature Distance
& Edit Distance

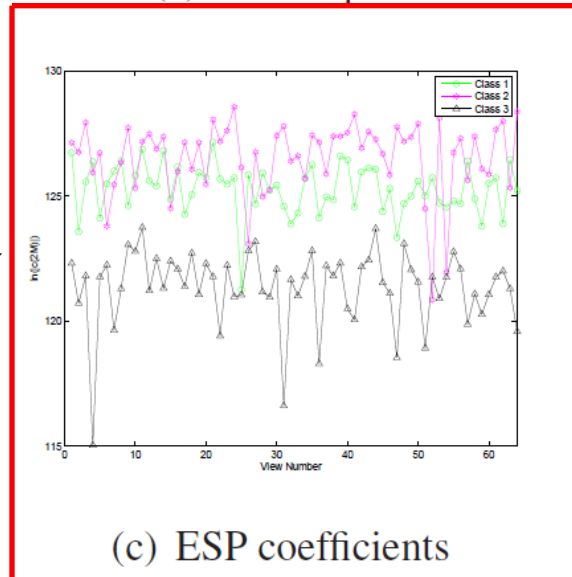


(a) Scatter plot

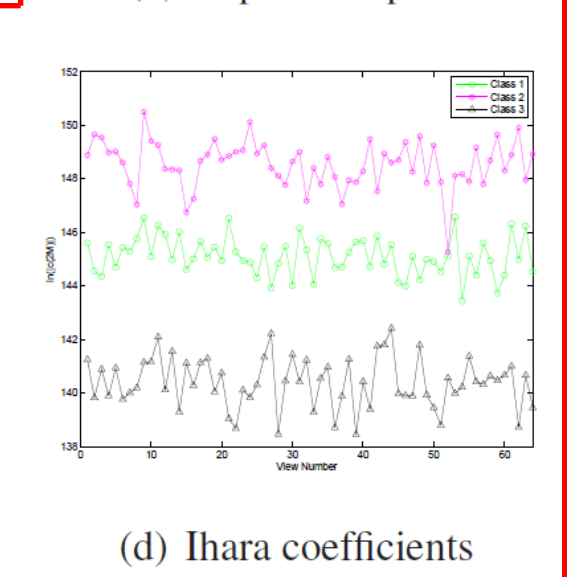


(b) Laplacian spectra

Three Classes of Randomly
Generated Graphs



(c) ESP coefficients



(d) Ihara coefficients

Experiments: Hypergraphs



(a) Houses



Pattern Vector	Number of Object Classes			
	5	6	7	8
Truncated Normalized Laplacian Spectra	0.7323	0.7074	0.7650	0.8030
Truncated Laplacian Spectra	0.8574	0.8564	0.8454	0.8449
Ihara Coefficients	0.9355	0.8859	0.8716	0.8812

Complexity

Information theory, graphs and
kernels.

Protein-Protein Interaction Networks

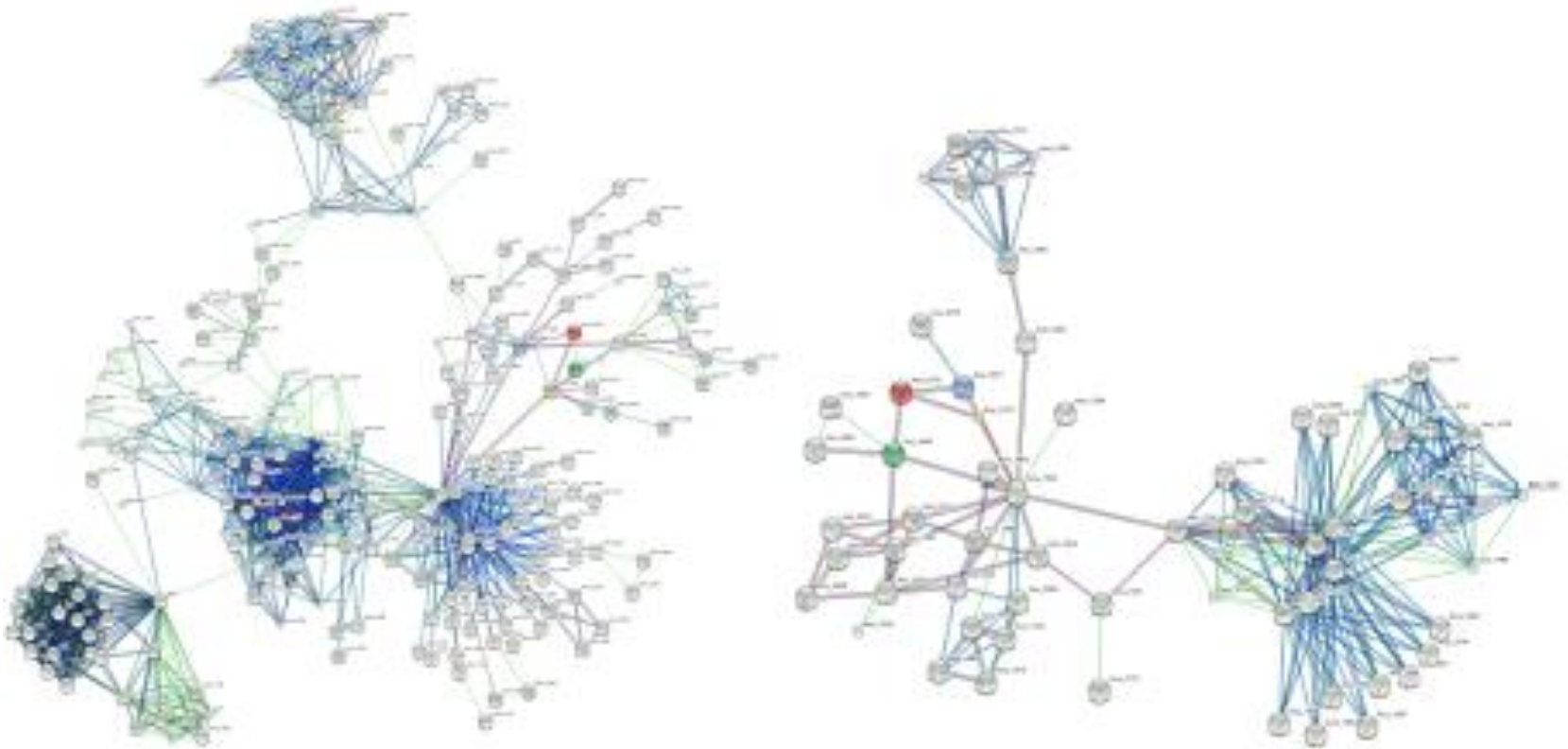


Figure: STRING Protein-Protein Interaction Networks

Characterising graphs

- **Topological:** e.g. average degree, degree distribution, edge-density, diameter, cycle frequencies etc.
- **Spectral or algebraic:** use eigenvalues of adjacency matrix or Laplacian, or equivalently the co-efficients of characteristic polynomial.
- **Complexity:** use information theoretic measures of structure (e.g. Shannon entropy).

Complexity characterisation

- **Information theory**: entropy measures
- Structural pattern recognition: graph spectral indices of structure and topology.
- Complex systems: measures of centrality, separation, searchability.

Information theory

- **Entropic measures of complexity:** Shannon , Erdos-Renyi, Von-Neumann.
- **Description length:** fitting of models to data, entropy (model cost) tensioned against log-likelihood (goodness of fit).
- **Kernels:** Use entropy to compute Jensen-Shannon divergence

Recent work

- Escolano, Hancock, Lozano: Phys. Rev. E. (Thermodynamic depth complexity using Bregman balls to compute entropy).
- Han, Wilson and Hancock: Patt. Rec. Lett. (fast entropy computation using approximation to von Neumann entropy).

Thermodynamic Depth Complexity

- Simulate heat flow on graph using continuous time random walk.
- Characterise nodes by their thermodynamic depth (time walk takes to reach node).
- Measure heat flow dependence at each node with time. Record maximum.
- Compute homogeneity statistics over thermodynamic depth.

Phase transition

- As time evolves complexity undergoes phase transition.
- Corresponds to maximum flow at a node.
- Maximum of entropy.

Von-Neumann Entropy

- Derived from normalised Laplacian spectrum

$$H_{VN} = -\sum_{i=1}^{|V|} \frac{\hat{\lambda}_i}{2} \ln \frac{\hat{\lambda}_i}{2}$$

$$\hat{L} = D^{-1/2} (D - A) D^{-1/2} = \hat{\Phi} \hat{\Lambda} \hat{\Phi}^T$$

- Comes from quantum mechanics and is entropy associated with density matrix.

Approximation

- Quadratic entropy

$$H_{VN} = \sum_{i=1}^{|\mathcal{V}|} \frac{\hat{\lambda}_i}{2} \left\{ 1 - \frac{\hat{\lambda}_i}{2} \right\} = \frac{1}{2} \sum_{i=1}^{|\mathcal{V}|} \hat{\lambda}_i - \frac{1}{4} \sum_{i=1}^{|\mathcal{V}|} \hat{\lambda}_i^2$$

- In terms of matrix traces

$$H_{VN} = \frac{1}{2} \text{Tr}[\hat{L}] - \frac{1}{4} \text{Tr}[\hat{L}^2]$$

Computing Traces

- Normalised Laplacian

$$\text{Tr}[\hat{L}] = |V|$$

- Normalised Laplacian squared

$$\text{Tr}[\hat{L}^2] = |V| + \sum_{(u,v) \in E} \frac{1}{4d_u d_v}$$

Simplified entropy

Collect terms together, von Neumann entropy reduces to

$$H_{VN} = \frac{1}{4} |V| - \sum_{(u,v) \in E} \frac{1}{4d_u d_v}$$

Homogeneity index

Based on degree statistics

$$\rho(G) = \sum_{(u,v) \in E} (d_u^{-1/2} - d_v^{-1/2})^2$$

$$\rho(G) = \frac{1}{|V| - 2\sqrt{|V| - 1}} \sum_{(u,v) \in E} \left\{ \frac{1}{d_u} + \frac{1}{d_v} - \frac{2}{\sqrt{d_u d_v}} \right\}$$

Homogeneity meaning

Limit of large degree

$$\rho(G) \sim \sum_{(u,v) \in E} \{CT(u,v) - 2A(u,v)\}$$

Largest when commute time differs from 2 due to large number of alternative connecting paths.

Directed Graphs

Von Neumann entropy comes from in-degree and out-degree of vertices connected by edges

$$H = 1 - 1/V - 1/2V^2 \sum_{(i,j) \in E} \frac{d_i^{in} / d_i^{out} + d_j^{in} / d_j^{out}}{d_i^{out} d_j^{in}}$$

Development comes from Laplacian of a directed graph (Chung).

Uses

- Complexity-based clustering (especially protein-protein interaction networks).
- Defining information theoretic (Jensen-Shannon) kernels.
- Controlling complexity of generative models of graphs.

Protein-Protein Interaction Networks

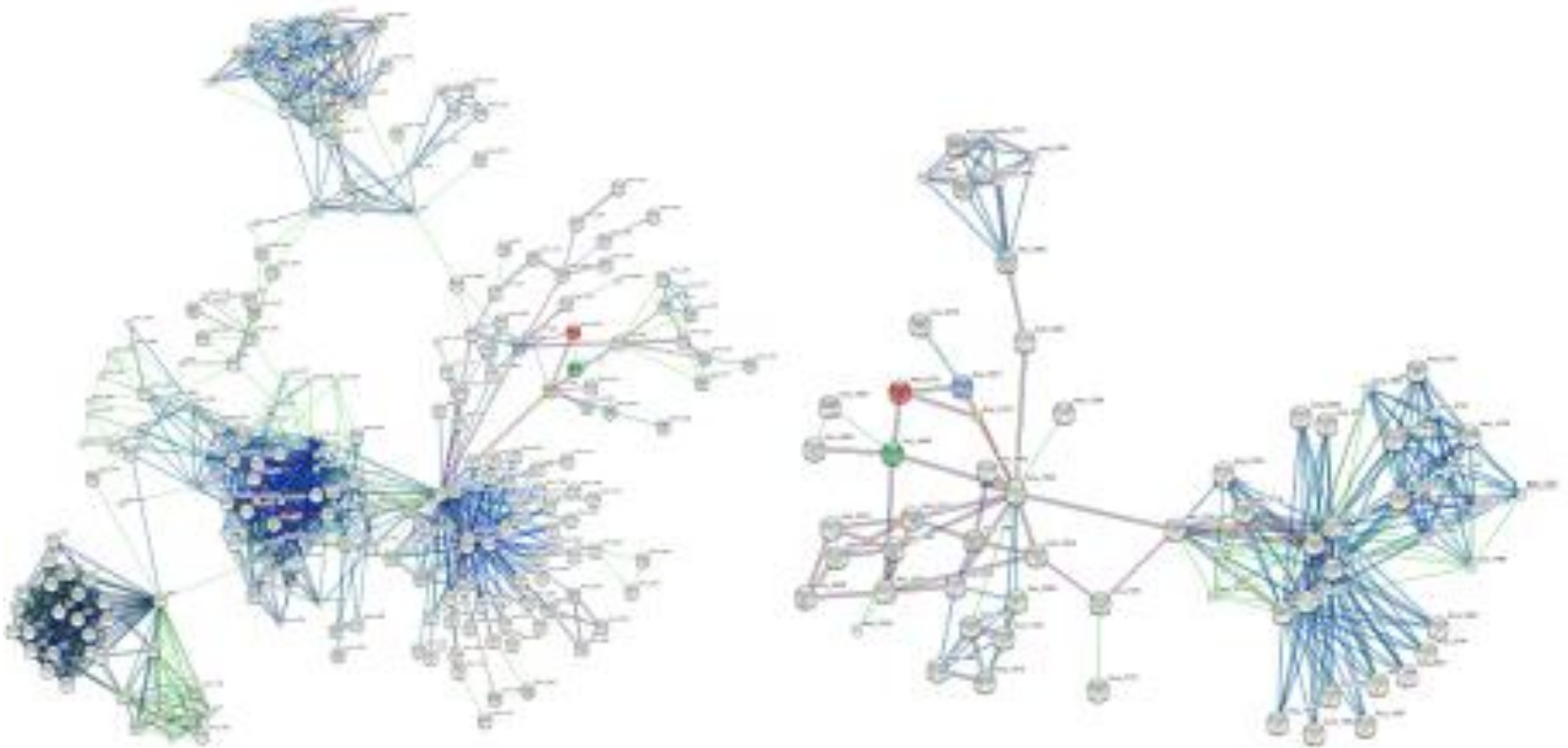


Figure: STRING Protein-Protein Interaction Networks

Experiment

PPIs from STRING database: <http://string.embl.de/>

- ▶ Protein studied: **histidine kinase** is key in the development of signal transduction.
- ▶ Preliminary comparison with Estrada's spectral homogeneity descriptor yields better performance.
- ▶ PPIs: 224 networks from 6 different groups (all the PPIs in the same group corresponds to the same species) with the following evolutive order (from older to more recent): *Aquifex*—4 PPIs, *Thermotoga*—4 PPIs, *Gram-Positive*—52 PPIs, *Cyanobacteria*—73 PPIs *Proteobacteria*—45 PPIs. There is an additional class (*Acidobacteria*—46 PPIs).

Method and Results

- ▶ Histogramming TDs reveals typically long tailed distributions with most of the TDs concentrated at a given point. **Are these points ordered according to the evolutive order?**
- ▶ This question can be answered by studying the cumulative distributions instead of the pdfs. In such case, reaching the top (cumulative=1) soon indicates low TD whereas reaching it later indicates high TD.
- ▶ Then, it can be seen that the evolving complexity of the signal transduction mechanism driven by the histidine kinase is properly quantified by TD for the 5 first phyla studied.
- ▶ However, the *Acidobacterium* sp. chosen seems older than Gram-Positive which seems not to be the case.

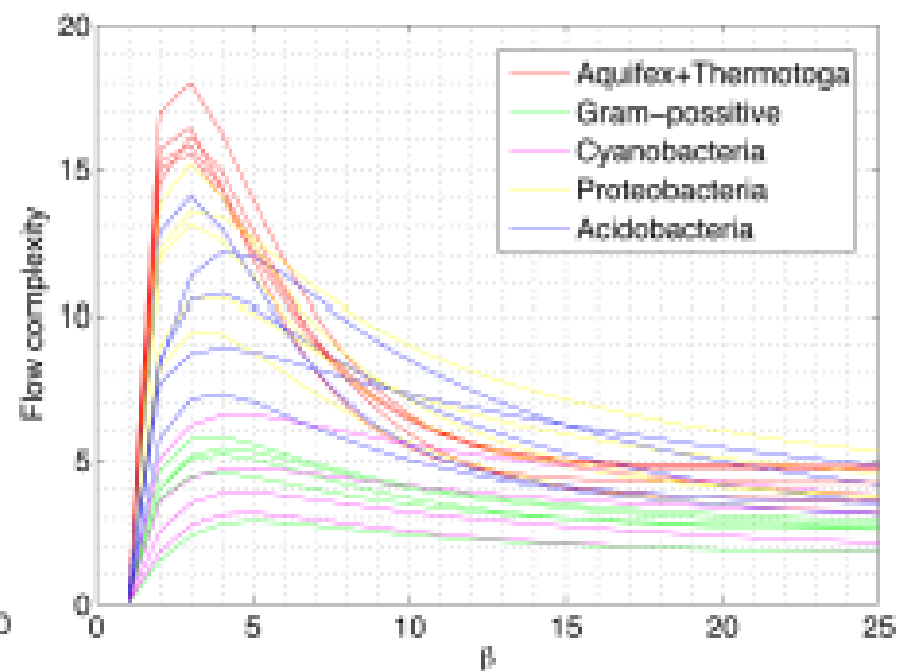
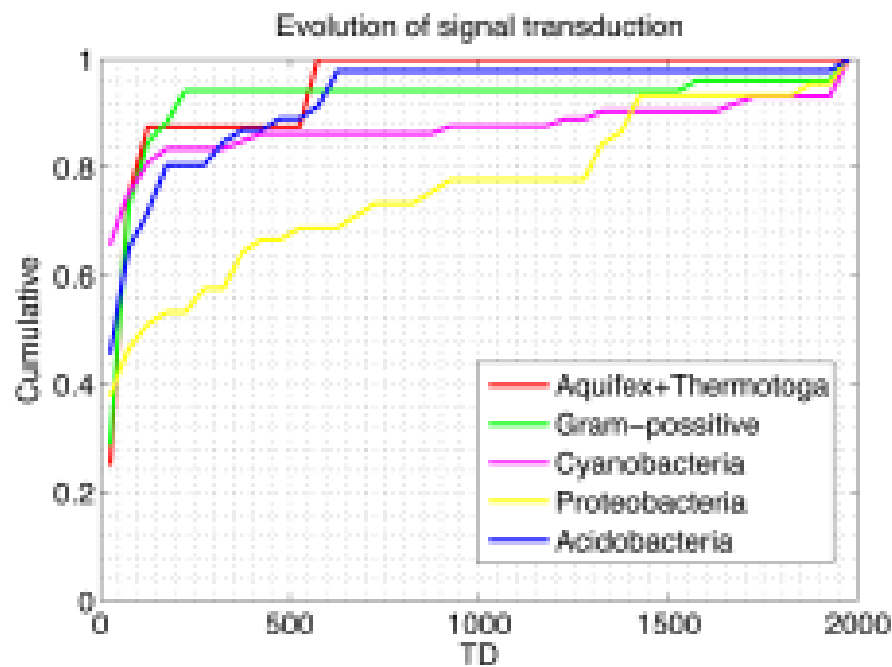


Figure: Evolutionary study (left) and Examples of complexity profiles corresponding to different species (right)

Jensen-Shannon Kernel

- Defined in terms of J-S divergence

$$K_{JS}(G_i, G_j) = \ln 2 - JS(G_i, G_j)$$

$$JS(G_i, G_j) = H(G_i \oplus G_j) - \{H(G_i) + H(G_j)\}$$

- Properties: extensive, positive.

Computation

- Construct direct product graph for each graph pair.
- Compute von-Neumann entropy difference between product graph and two graphs individually.
- Construct kernel matrix over all pairs.



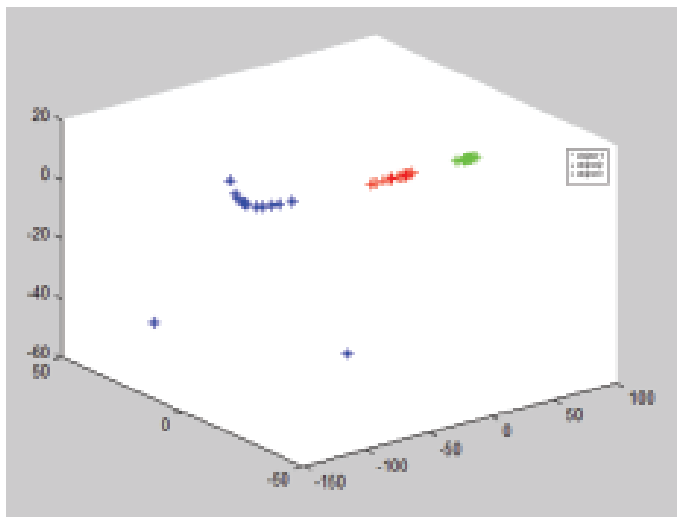
(a) Boxes from ALOI



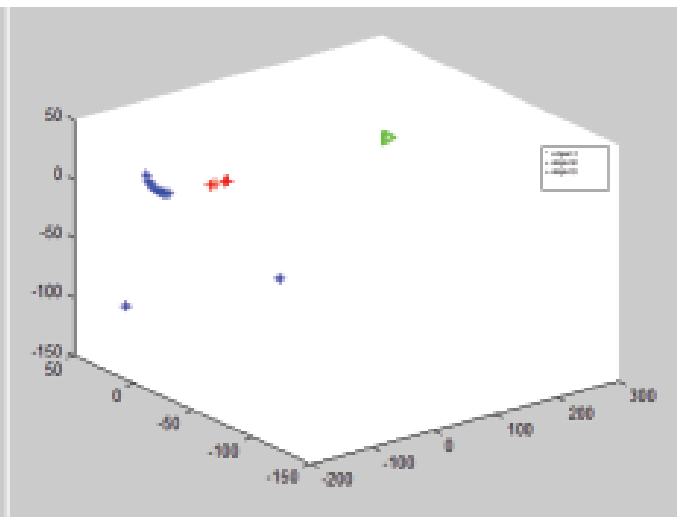
(b) Houses from CMU AND MOVI



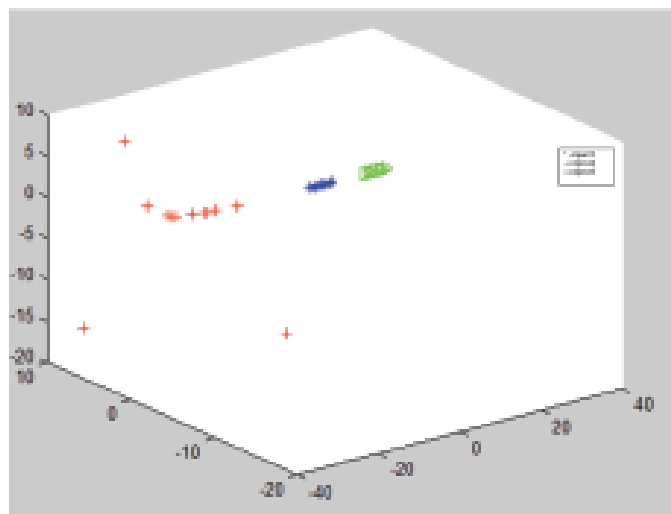
(c) Cups from COIL



(a) Experiments Performance of ALOI



(b) Experiment Performance of CMU and MOVI



(c) Experiment Performance of COIL dataset

Learning

Generative Models

- Structural domain: define probability distribution over prototype structure. Prototype together with parameters of distribution minimise description length (Torsello and Hancock, PAMI 2007) .
- Spectral domain: embed nodes of graphs into vector-space using spectral decomposition. Construct point distribution model over embedded positions of nodes (Bai, Wilson and Hancock, CVIU 2009).

Deep learning

- Deep belief networks (Hinton 2006, Bengio 2007).
- Compositional networks (Amit+Geman 1999, Fergus 2010).
- Markov models (Leonardis 2000)
- Stochastic image grammars (Zhu, Mumford, Yuille)
- Taxonomy/category learning (Todorovic+Ahuja, 2006-2008).

Aim

- Combine spectral and structural methods.
- Use description length criterion.
- Apply to graphs rather than trees.

Prior work

- IJCV 2007 (Torsello, Robles-Kelly, Hancock) –shape classes from edit distance using pairwise clustering.
- PAMI 06 and Pattern Recognition 05 (Wilson, Luo and Hancock) – graph clustering using spectral features and polynomials.
- PAMI 07 (Torsello and Hancock) – generative model for variations in tree structure using description length.
- CVIU09 (Xiao, Wilson and Hancock) – generative model from heat-kernel embedding of graphs.

Structural learning

Using description length

Description length

- Wallace+Freeman: minimum message length.
- Rissanen: minimum description length.

Use log-posterior probability to locate model that is optimal with respect to code-length.

Similarities/differences

- MDL: selection of model is aim; model parameters are simply a means to this end. Parameters usually maximum likelihood. Prior on parameters is flat.
- MML: Recovery of model parameters is central. Parameter prior may be more complex.

Coding scheme

- Usually assumed to follow an exponential distribution.
- Alternatives are universal codes and predictive codes.
- MML has two part codes (model+parameters). In MDL the codes may be one or two-part.

Method

- Model is supergraph (i.e. Graph prototypes) formed by graph union.
- Sample data observation model: Bernoulli distribution over nodes and edges.
- Mode: complexity: Von-Neumann entropy of supergraphs.
- Fitting criterion:
 - MDL-like-make ML estimates of the Bernoulli parameters
 - MML-like: two-part code for data-model fit + supergraph complexity.

Model overview

- Description length criterion

$$L(G, \Gamma) = LL(G | \Gamma) + H(\Gamma)$$

code-length=negative + model code-length
log-likelihood (entropy)

Data-set: set of graphs G

Model: prototype graph+correspondences with it

Updates by expectation maximisation:

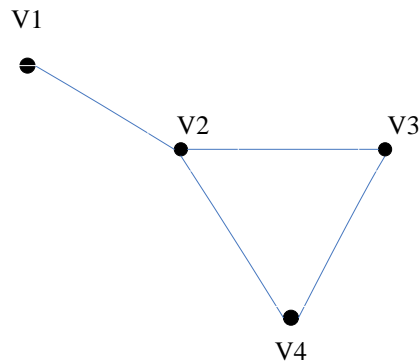
Model graph adjacency matrix (M-step)
+ correspondence indicators (E-step).

Learn supergraph using MDL

- Follow Torsello and Hancock and pose the problem of learning generative model for graphs as that of learning a supergraph representation.
- Required probability distributions is an extension of model developed by Luo and Hancock.
- Use von Neumann entropy to control supergraph's complexity.
- Develop an EM algorithm in which the node correspondences and the supergraph edge probability matrix are treated as missing data.

Probabilistic Framework

Here the structure of the sample graphs and the supergraph are represented by their adjacency matrices



$$A = \begin{matrix} & \begin{matrix} V1 & V2 & V3 & V4 \end{matrix} \\ \begin{matrix} V1 \\ V2 \\ V3 \\ V4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

Observation model

Given a sample graph $G_i = (V_i, E_i)$ and a supergraph $\Gamma = (V_\Gamma, E_\Gamma)$

$$D_{ab}^i = \begin{cases} 1 & \text{if } (a, b) \in E_i \\ 0 & \text{otherwise} \end{cases}, \quad M_{\alpha\beta} = \begin{cases} 1 & \text{if } (\alpha, \beta) \in E_\Gamma \\ 0 & \text{otherwise} \end{cases}$$

along with their assignment matrix,

$$s_{a\alpha}^i = \begin{cases} 1 & \text{if } a \rightarrow \alpha \\ 0 & \text{otherwise} \end{cases}$$

the *a posteriori* probabilities of the sample graphs given the structure of the supergraph and the node correspondences is defined as

$$P(G_i | \Gamma, S^i) = \prod_{a \in V_i} \sum_{\alpha \in V_\Gamma} K_a^i \exp[\mu \sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^i M_{\alpha\beta} s_{b\beta}^i]$$

Data code-length

- For the sample graph-set $\mathcal{G} = \{ G_1, \dots, G_i, \dots, G_N \}$ and the supergraph Γ , the set of assignment is $\mathcal{S} = \{ S^1, \dots, S^i, \dots, S^N \}$. Under the assumption that the graphs in \mathcal{G} are independent samples from the distribution, the likelihood of the sample graphs can be written

$$P(\mathcal{G}|\Gamma, \mathcal{S}) = \prod_{G_i \in \mathcal{G}} P(G_i|\Gamma, S^i) = \prod_{G_i \in \mathcal{G}} \prod_{a \in V_i} \sum_{\alpha \in V_\Gamma} K_a^i \exp[\mu \sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^i M_{\alpha\beta} S_{b\beta}^i]$$

- Code length of observed data

$$LL(\mathcal{G}|\Gamma) = -\frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \log P(G_i|\Gamma, S^i)$$

Overall code-length

- According to Rissanen and Grunwald's minimum description length criterion, we encode and transmit the sample graphs and the supergraph structure. This leads to a two-part message whose total length is given

$$\mathcal{L}(\mathcal{G}, \Gamma) = LL(\mathcal{G}|\Gamma) + H_{VN} =$$
$$-\frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \log \left\{ \sum_{\alpha \in V_\Gamma} K_a^\alpha \exp \left[\mu \sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^\alpha M_{ab} s_{b\beta}^i \right] \right\} + \frac{|V_\Gamma|}{4} - \sum_{(\alpha, \beta) \in E_\Gamma} \frac{1}{4 T_\alpha T_\beta}$$

- We consider both the node correspondence information between graphs S and the structure of the supergraph M as missing data and locate M by minimizing the overall code-length using EM algorithm.

EM – code-length criterion

$$\begin{aligned} \Lambda^{(n+1)}(\mathcal{G}|I, \mathcal{S}^{(n+1)}) &= \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \sum_{\alpha \in V_I} Q_{a\alpha}^{i,(n)} \left\{ \ln K_a^i + \mu \sum_{b \in V_i} \sum_{\beta \in V_I} D_{ab}^i M_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n+1)} \right\} \\ &- \frac{|V_I|}{4} + \sum_{(\alpha, \beta) \in E_I} \frac{1}{4 T_\alpha^{(n)} T_\beta^{(n)}} . \end{aligned} \quad (16)$$

Expectation + Maximization

- M-step :

Recover correspondence matrices: Take partial derivative of the weighted log-likelihood function and soft assign.

$$\frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{b\beta}^{i,(n+1)}} = \frac{1}{|\mathcal{G}|} \sum_{a \in V_i} \sum_{\alpha \in V_G} Q_{a\alpha}^{i,(n)} D_{ab}^i M_{\alpha\beta}^{(n)}$$

$$s_{a\alpha}^{i,(n+1)} \leftarrow \exp\left[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{a\alpha}^{i,(n+1)}}\right] / \sum_{\alpha' \in V_G} \exp\left[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{a\alpha'}^{i,(n+1)}}\right]$$

Modify supergraph structure :

$$M_{\alpha\beta}^{(n+1)} \leftarrow \exp\left[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha\beta}^{(n)}}\right] / \sum_{(\alpha',\beta') \in E_G} \exp\left[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha'\beta'}^{(n)}}\right]$$

$$\frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha\beta}^{(n)}} = \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \sum_{b \in V_i} Q_{a\alpha}^{i,(n)} D_{ab}^i s_{b\beta}^{i,(n+1)} - \frac{1}{(T_\alpha^{(n)})^2} \sum_{(\alpha,\beta') \in E_G} \frac{1}{4 T_{\beta'}^{(n)}}$$

- E-step: Compute the *a posteriori* probability of the nodes in the sample graphs being matching to those of the supergraph.

$$Q_{a\alpha}^{i,(n+1)} = \frac{\exp\left[\sum_{b \in V_i} \sum_{\beta \in V_G} D_{ab}^i M_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n)}\right] \pi_\alpha^{i,(n)}}{\sum_{\alpha' \in V_G} \exp\left[\sum_{b \in V_i} \sum_{\beta \in V_G} D_{ab}^i M_{\alpha'\beta}^{(n)} s_{b\beta}^{i,(n)}\right] \pi_{\alpha'}^{i,(n)}}$$

Experiments

Delaunay graphs from images of different objects.



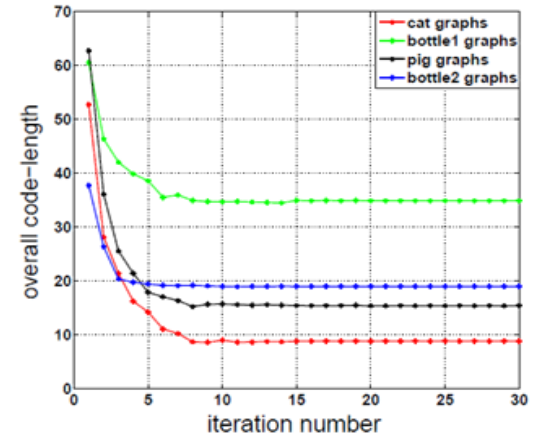
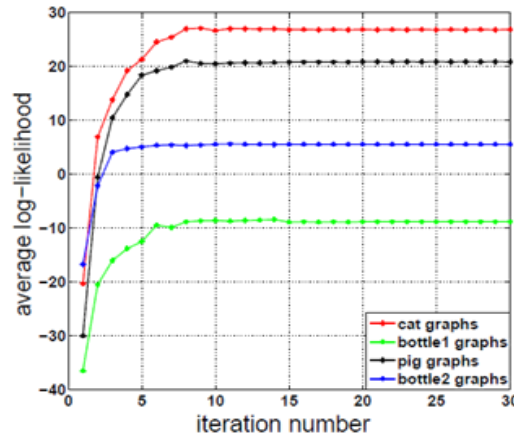
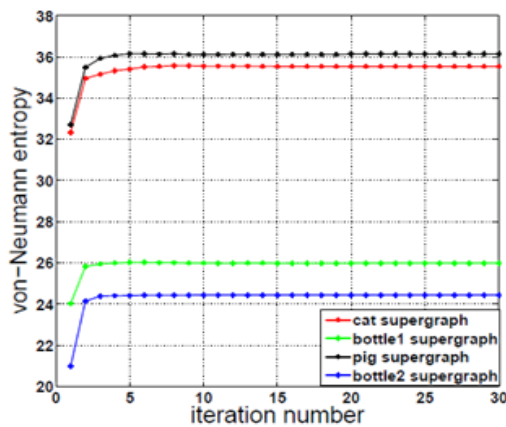
COIL dataset



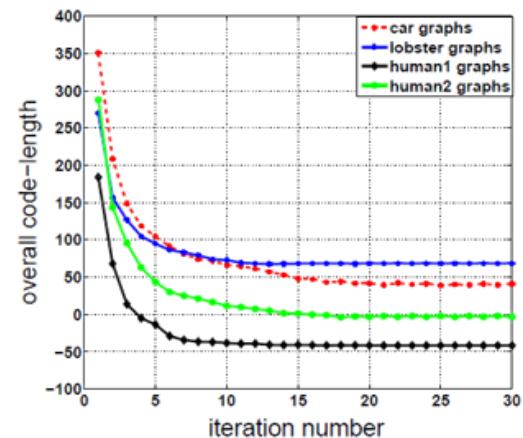
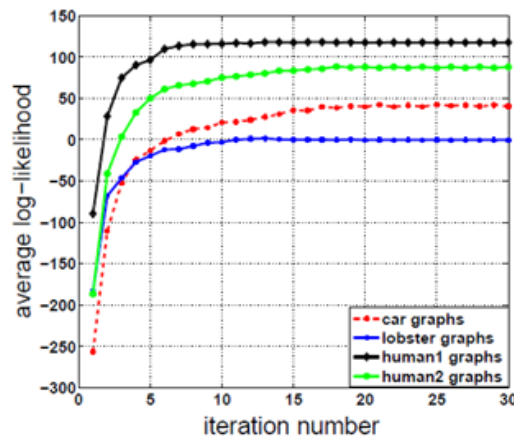
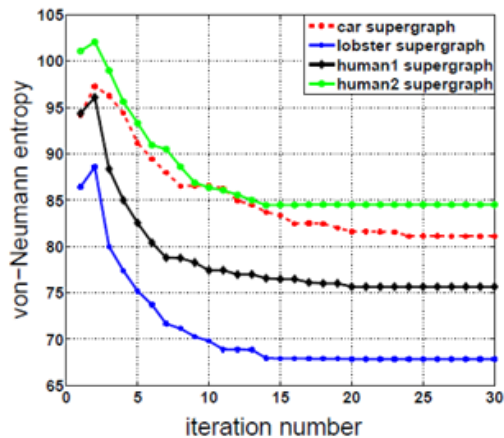
Toys dataset

Experiments---validation

- **COIL dataset: model complexity increase, graph data log-likelihood increase, overall code length decrease during iterations.**



- **Toys dataset: model complexity decrease, graph data log-likelihood increase, overall code length decrease during iterations.**



Experiments---classification task

We compare the performance of our learned supergraph on classification task with two alternative constructions, the median graph and the supergraph learned without using MDL. The table below shows the average classification rates from 10-fold cross validation, which are followed by their standard errors.

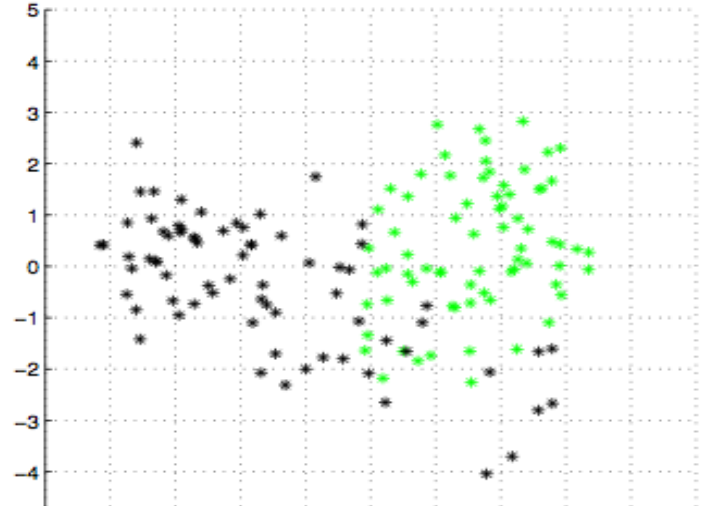
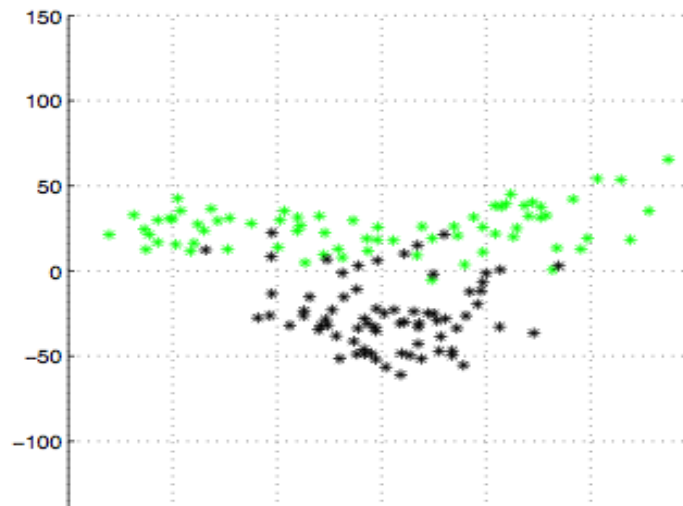
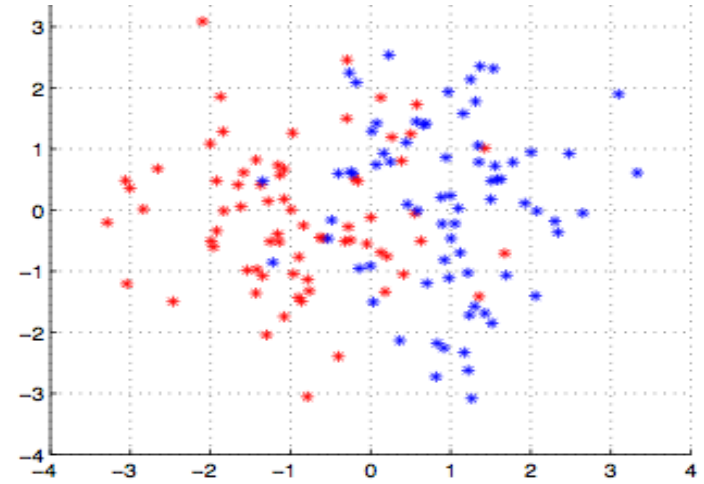
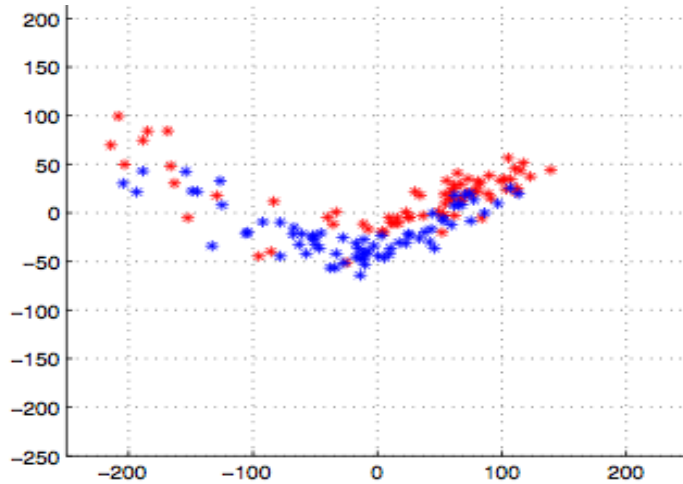
<i>Classification Rate</i>	cat & pig	bottle1 & bottle2	four objects (Toys)
learned supergraph(by MDL)	0.824 \pm 0.033	0.780 \pm 0.023	0.763 \pm 0.026
median graph/concatenated graph	0.669 \pm 0.052	0.651 \pm 0.023	0.575 \pm 0.020
learned supergraph	0.807 \pm 0.056	0.699 \pm 0.029	0.725 \pm 0.022

Experiments---graph embedding

Pairwise graph distance based on the Jensen-Shannon divergence and the von Neumann entropy of graphs

$$JSD(G_i, G_j) = H(G_i \otimes G_j) - \frac{H(G_i) + H(G_j)}{2}$$

Experiments---graph embedding



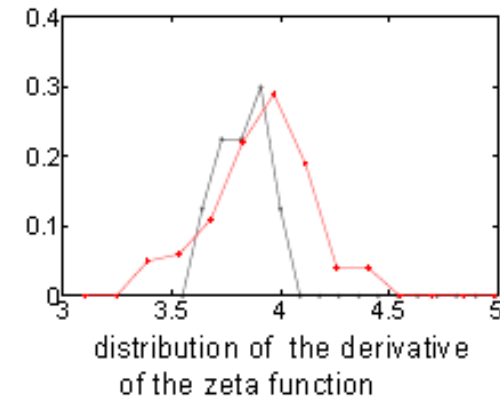
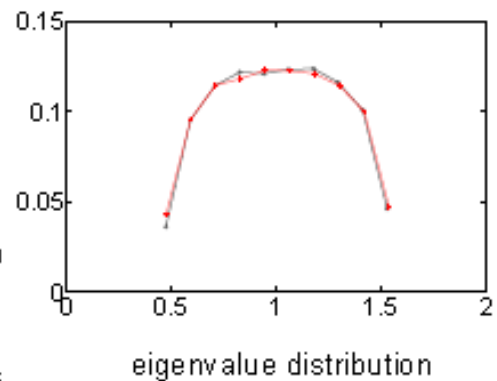
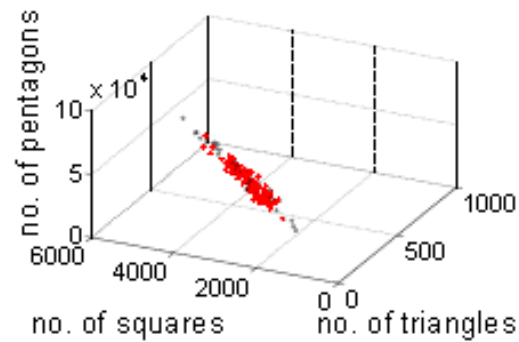
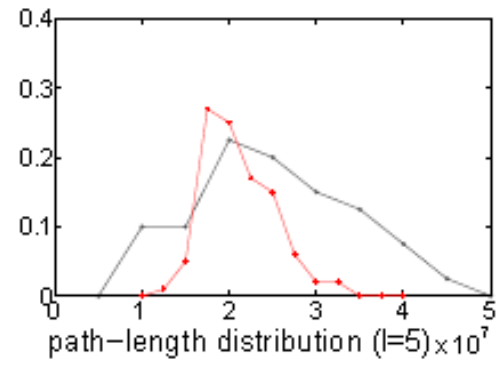
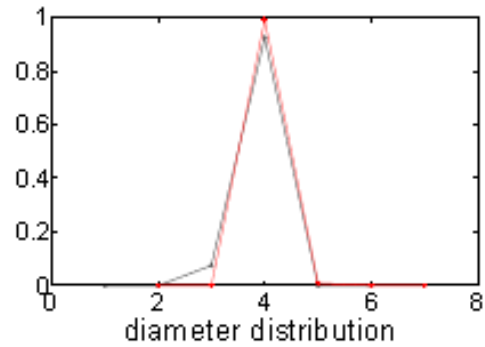
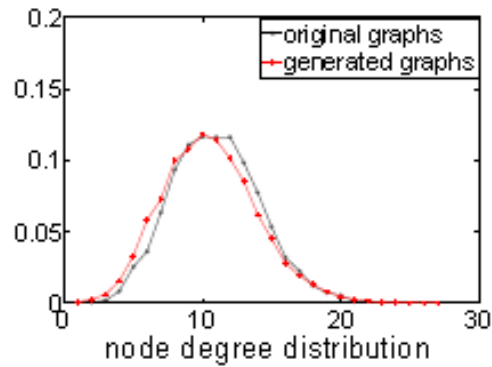
Edit distance

JSD distance

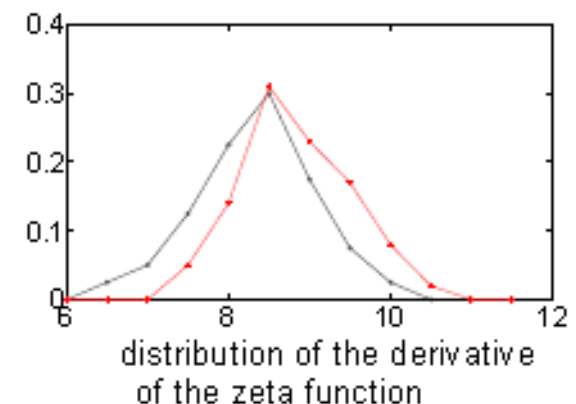
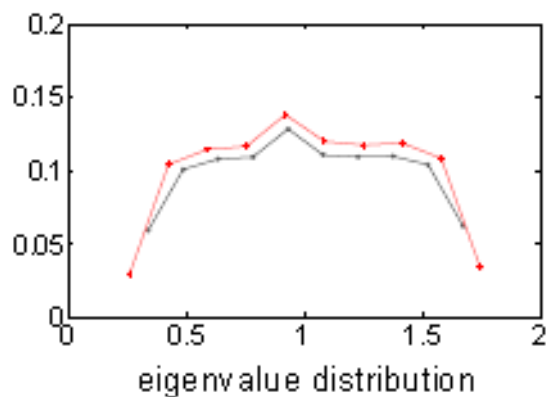
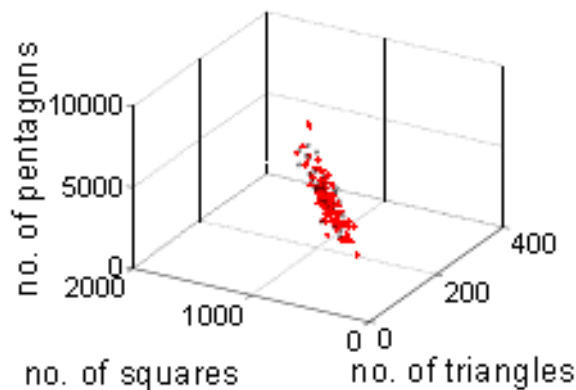
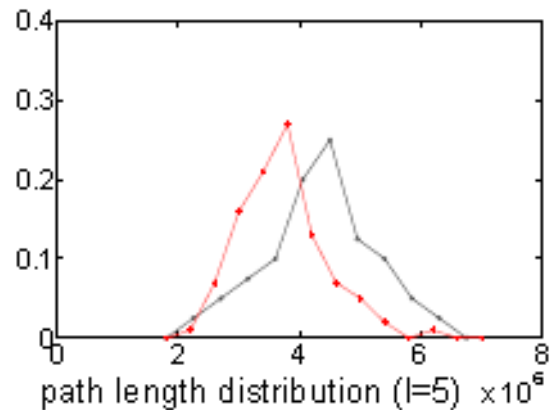
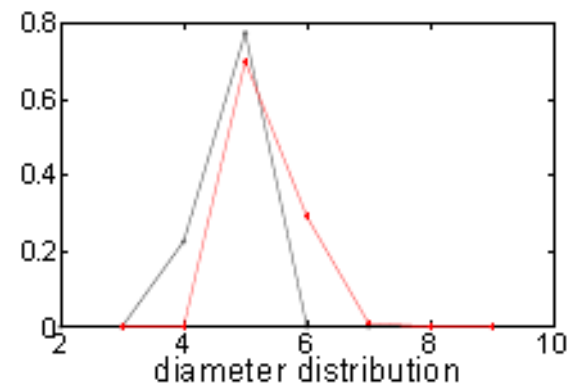
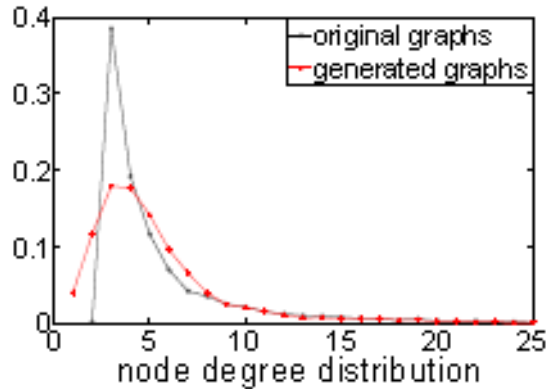
Generative model

- Train on graphs with set of predetermined characteristics.
- Sample using Monte-Carlo.
- Reproduces characteristics of training set, e.g. Spectral gap, node degree distribution, etc.

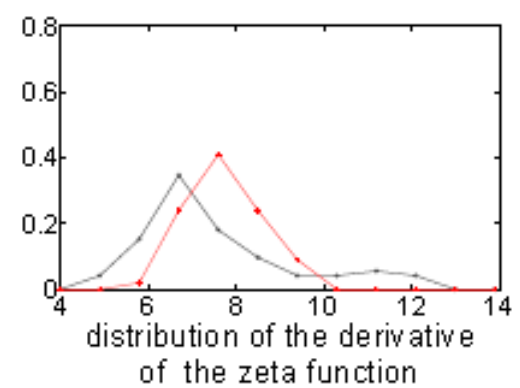
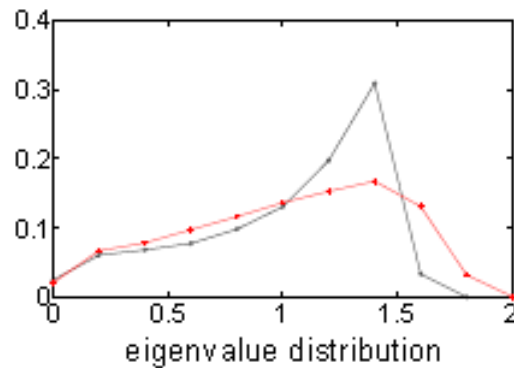
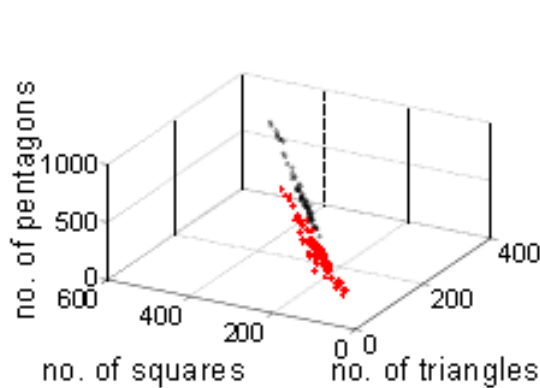
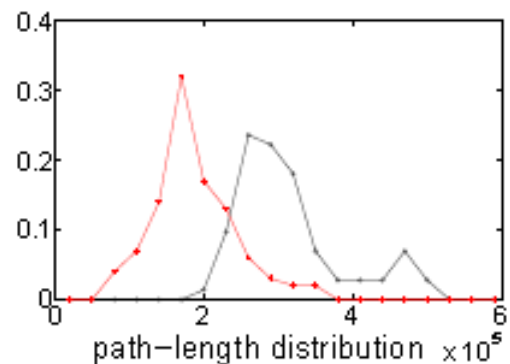
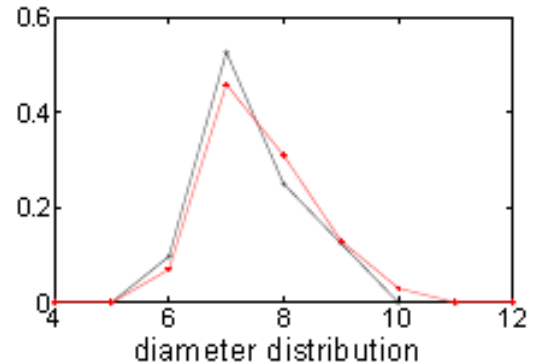
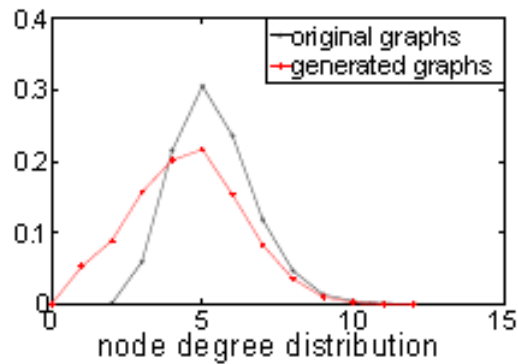
Erdos Renyi



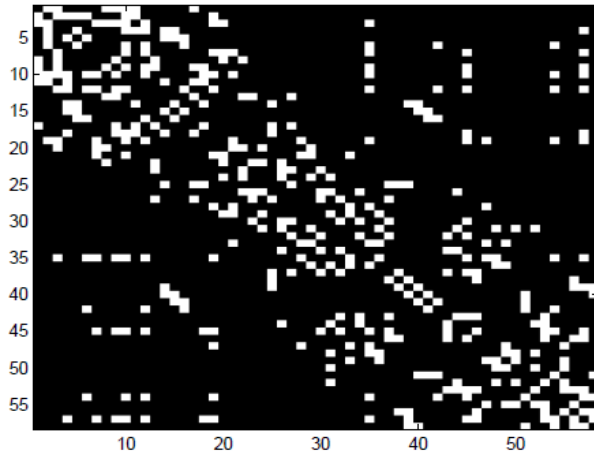
Barabasi Albert (scale free)



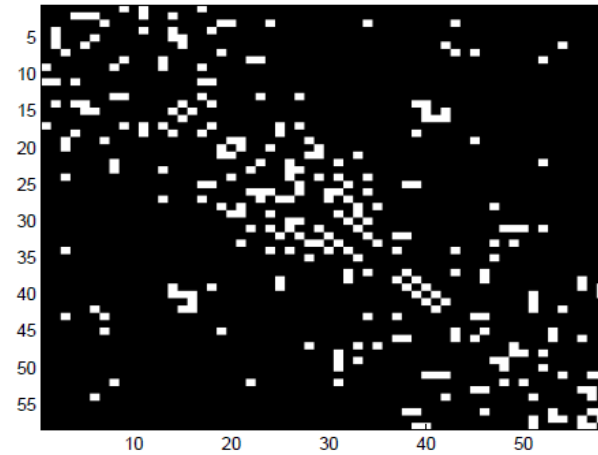
Dealunay Graphs



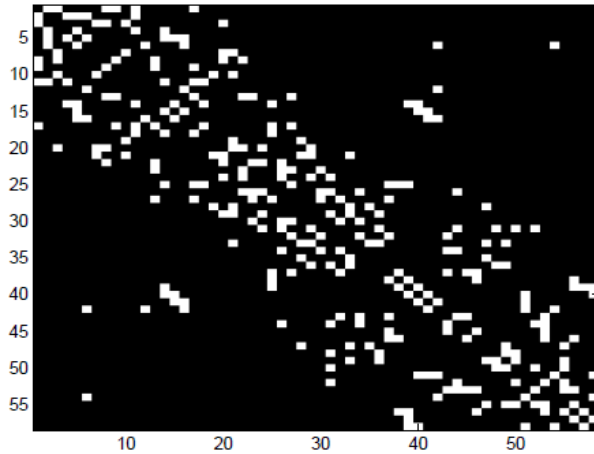
Experiments---generate new samples



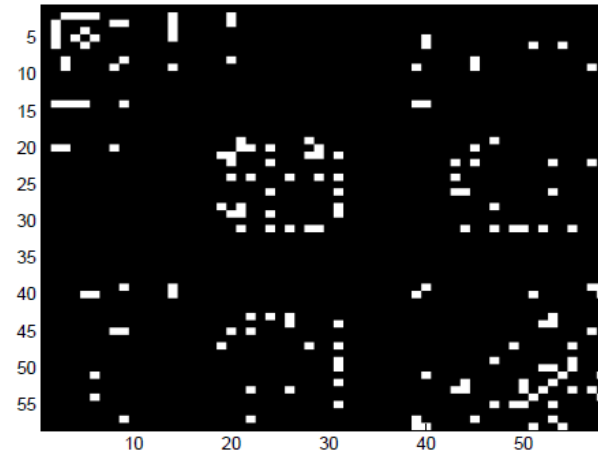
(a) Supergraph



(b) Generated sample graph with high likelihood.



(c) Median graph



(d) Generated sample graph that has low likelihood

Conclusions

- Shown how graph spectra can be used as characterisations of both structure and complexity.
- Presented MDL framework which uses complexity characterisation to learn generative model of graph structure.
- Future: Deeper measures of structure (symmetry) and detailed dynamics of network evolution.