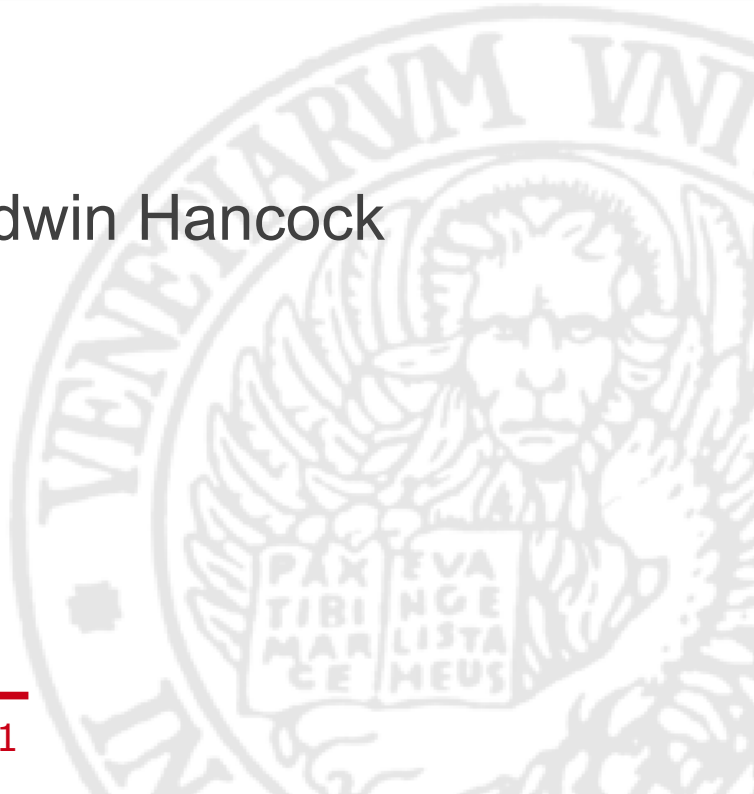
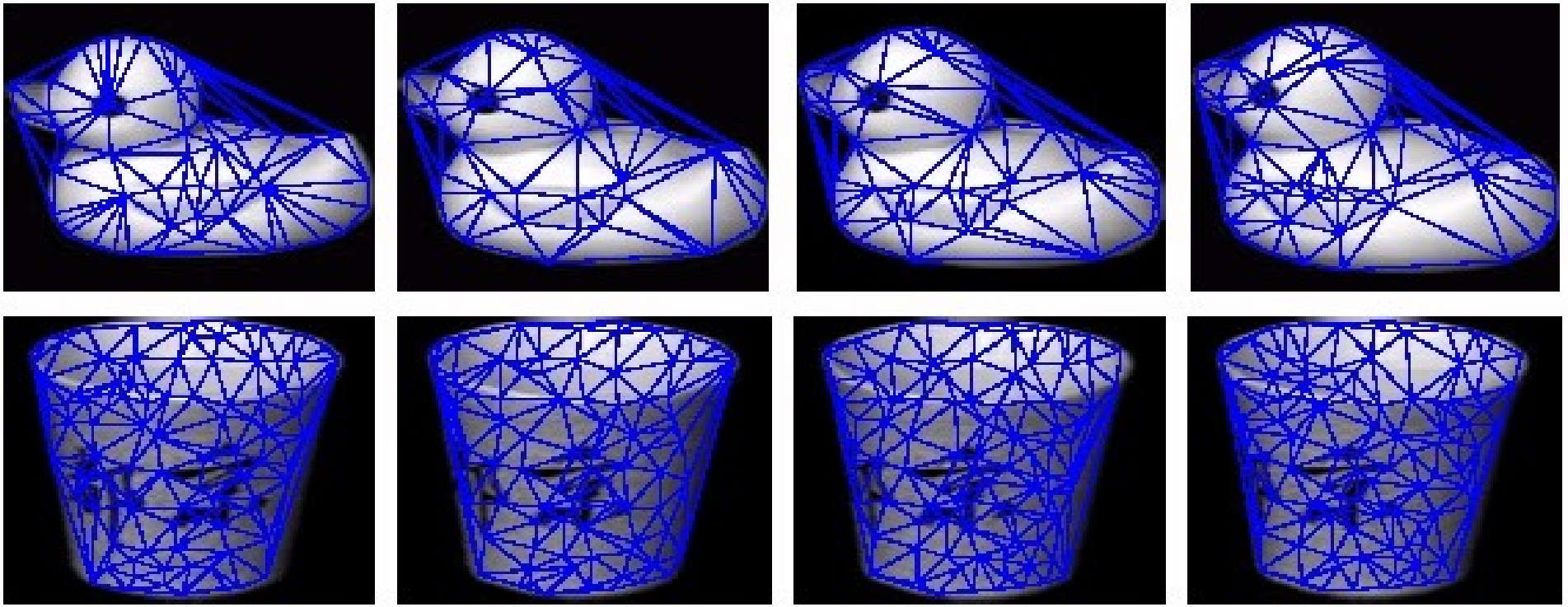
Learning Structure and Symmetry

Andrea Torsello
Joint work with Luca Rossi and Edwin Hancock



Graph-Based Representations

- Capture relational arrangements
- Provide contextual information needed to disambiguate part-identification
- Invariant to transformations (rotation, change in viewpoint, change of scale…)



Learning with Graphs

The algorithms used to segment the primitives are not reliable, as a result there are both additional and **missing nodes** and **variations in edge-structure**.

Hence image matching and recognition cannot be reduced to a simple graph isomorphism or even a subgraph isomorphism problem. Instead inexact graph matching methods are needed.

Little work on classification of sample structure and on learning representations of the extracted classes and group invariants.

Relatively little methodology available as vectorial methods from statistical machine learning are not easily applied on graphs

Difficulties in Graph Learning

Since it is not clear how to convert a graph into a vectorial representation, applying standard pattern recognition techniques is not straightforward

- There is no natural ordering of nodes and edges
  - Correspondences must be used to establish order
- The number of nodes and edges is not fixed
  - Due to noise, occlusion, segmentation errors
- Not easily summarized
  - Since they do not reside in a vector space, mean and covariance are hard to characterize

There have been some successful attempts at embedding graph into vector spaces, but they are not able to characterize the structural variation of the set

Learning with graphs

Prototype learning: Requires model prototypes for classification. Uses nearest neighbor classification (requires distances). Recent interest in learning prototypes (graph median) (Bunke et al.)

Work with (dis) similarities: Can perform pariwise clustering or embed sets of graphs in a vector space using multidimensional scaling on similarities. (Bunke; Buhman; Luo+Torsello+Robles-Kelly).

Embed individual graphs in a low dimensional space: Characterise structural variations in terms of statistical variation in a point-pattern. (Luo,Wilson)

Learn modes of structural variation: Understand how edge (connectivity) structure varies for graphs belonging to the same class. Requires correspondences of raw structure or alignment of an embedded one. (Dickinson,Williams,Torsello).

Construct generative model: Borrow ideas from graphical model to construct model for raw structures or point distribution model to for embedded graphs. (Langley, Friedman, Koller, Torsello, Xiao)

Learning Structure

Characterize the distribution of structure form a set of sample graphs: ***Understand how edge (connectivity) structure varies for graphs belonging to the same class.***

Build generative model: ***Define naïve model and use mixture of simple models to classify (density estimation in graph space)***

Perform supervised/unsupervised classification of graphs

Tree Union (Torsello and Hancock 2006)

Model $\mathcal{T}$ is composed of structure and sampling probability :

- The structural part is a tree with node set N and an order relation $O \subseteq N \times N$
- The sample probability $\theta : N \rightarrow [0,1]$ is a function that associates to each node the probability of being sampled

A sample is a tree t with node-set $N^t \subseteq N$ and hierarchy O^t: restriction to N^t of O.

Sampling is removing node i with probability θ_i.

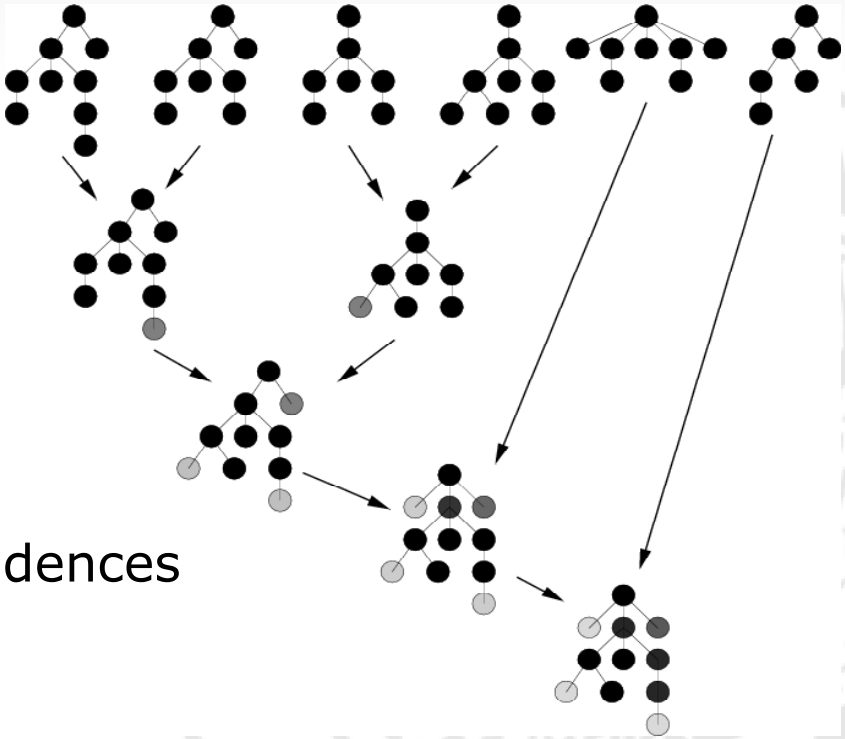
Learning the model

The correspondences are needed to construct the generalized model

The generalized model is needed to estimate the correspondences

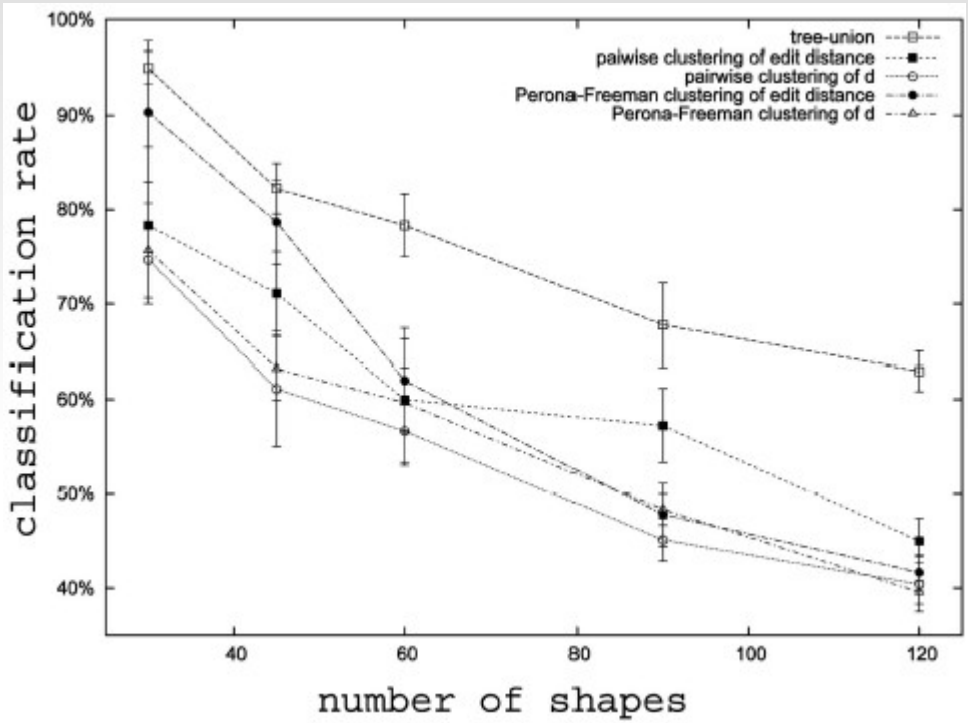
We adopted a description length approach to model estimate:

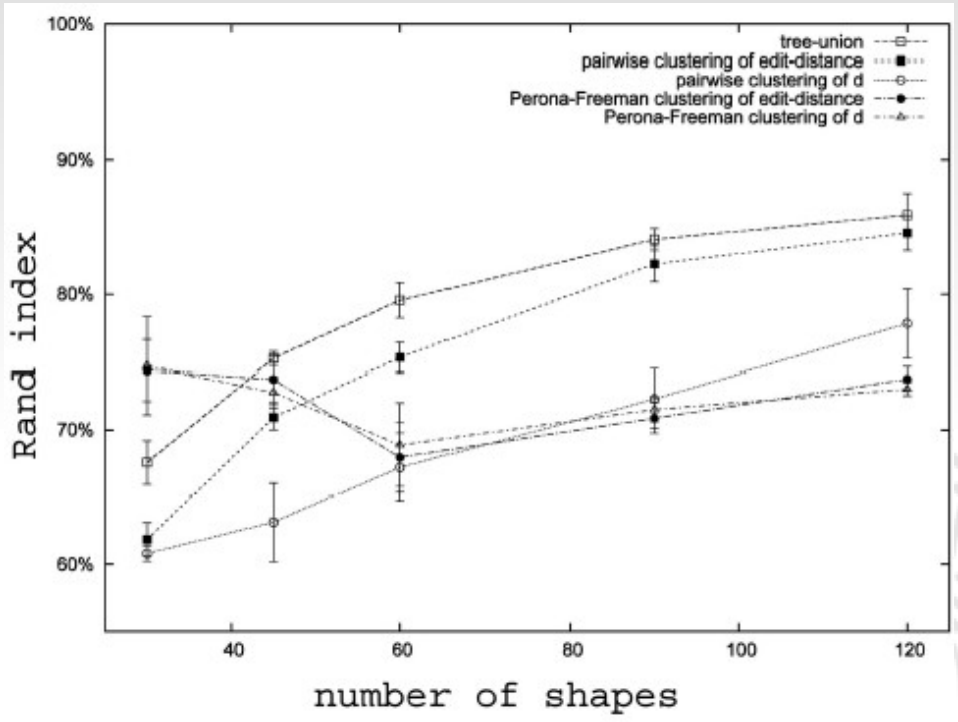
- Start with every tree forming a separate model
- Iteratively merge the pair of trees that reduce the description length the most
- Stop when description length cannot be reduced further



Edit distance used for estimating correspondences

Model-node entropy determine edit costs


classification rate
number of shapes
tree-union
paiwise clustering of edit distance
pairwise clustering of d
Perona-Freeman clustering of edit distance
Perona-Freeman clustering of d
100%
90%
80%
70%
60%
50%
40%
40
60
80
100
120


Rand index
number of shapes
tree-union
pairwise clustering of edit-distance
pairwise clustering of d
Perona-Freeman clustering of edit-distance
Perona-Freeman clustering of d
100%
90%
80%
70%
60%
40
60
80
100
120

Generative Graph Model

We want to generalize the approach to graphs

Decouple structure, class membership, and observation parameters

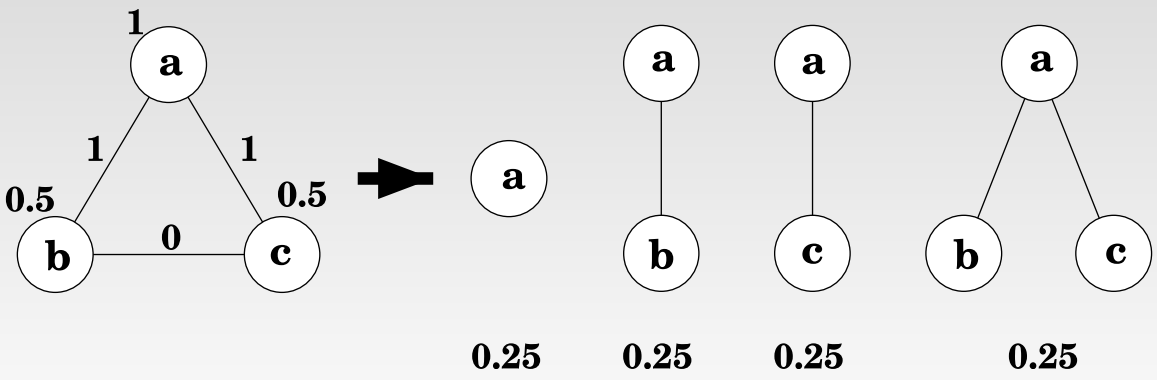
Given a set of undirected graphs **S**, our goal is to learn a generative graph model **G** that can be used to describe the distribution of structural data and characterize the structural variation of the set

Assume that the model is a mixture of naïve models where observations of nodes and edges are independent of the others

The naïve graph model is composed of

- A structural part G=(V,E)
  - Where V are all the nodes that can be generated by the graph and $E \subseteq V \times V$ is the set of possible edges
- A stochastic part that encodes the variability in the observations

Generative Graph Model



θ_i probability of generating node I

T_{ij} probability of generating node (i,j), conditioned on the generation of both i and j

W^n_i and W^e_{ij} generative models for the nodes and edges attributes

Note that θ_i and W^n_i need not to be independent (nor do T_{ij} and W^e_{ij})

Correspondences

After we sample a graph *g* from G, we lose track of the correspondences between the sample's nodes and the nodes of the model

A random permutation is applied to the nodes of the sample

Hence the observation probability of a graph depends on this unknown set of correspondences.

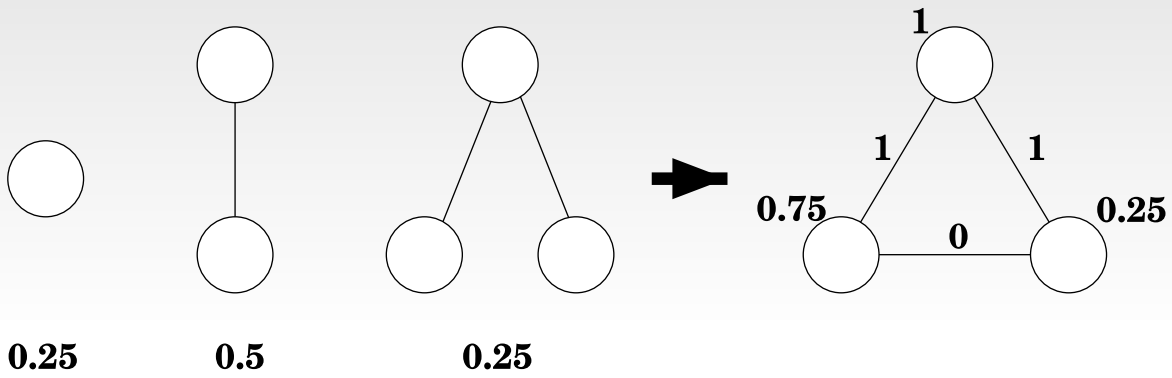
In particular, given the set of correspondences σ between nodes in $\mathcal{G}$ and nodes in *g*, the probability of observing *g* from G is

Typically the correspondences are estimated using graph matching techniques. This is equivalent to say that

$$P(g|\mathcal{G}) = \max_{\sigma \in \Sigma_n} P(g|\mathcal{G}, \sigma)$$

Symmetry bias

Maximum likelihood estimation of correspondences introduces **bias** in the model estimation Whenever you have **model** symmetries



Common wisdom: ignore graphs

- Derives from result by Erdös Rényi (1963) that states that as *n* grows almost all graphs of size *n* have no non-trivial automorphism

Note that In this result asymmetry is a characteristic of the observation

The model is actually maximally symmetric!

Overcoming the Bias

- An alternative approach is to take the expectation over all the possible correspondences

$$P(\hat{g}|\mathcal{G}) = \sum_{\sigma \in \Sigma_n^m} P(g|\mathcal{G},\sigma)P(\sigma) = \frac{1}{|\Sigma_g|} \sum_{\sigma \in \Sigma_n^m} P(g|\mathcal{G},\sigma)$$

  - Note the $|\Sigma_g|$ term taking into account the symmetries of g

- Can be rewritten as $P(g|\mathcal{G},\sigma) = \prod_{i=1}^{n} \prod_{j=i}^{n} \Theta_{ij}^{\sigma(i)\sigma(j)}$

  - Generalizes the permanent to average quadratic assignment

- Averaging over all possible correspondences is not possible due to the super-exponential growth of the space

- Resort to an estimation approach: importance sampling approach to compute a fast-converging estimate of $P(g|\mathcal{G})$

Correspondence Sampler

We can sample a correspondence as follows

- We start from an initial guess of the correspondences matrix M
  - m_{ih} gives the probability that model node i corresponds to observation node h

$$\sum_h m_{ih} \leq 0 \qquad \sum_i m_{ih} \leq 0$$

- We sample the correspondence for model node i picking a node j with probability m_{ij}
- Then, we condition M to the current match by taking into account the structural information between the sampled node and all the others

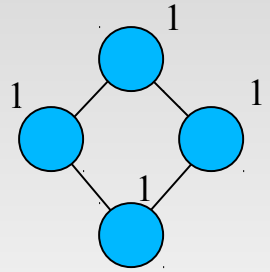
$$M_{1,\ldots,i}^{h_1,\ldots,h_i} = \pi\left(M_{1,\ldots,i-1}^{h_1,\ldots,h_{i-1}} \odot \Theta_{i-}^{h_i-}\right)$$

  - Project to sub-bistocastic matrix minimizing KL divergence

The process is iterated until a complete set of correspondences σ has been sampled

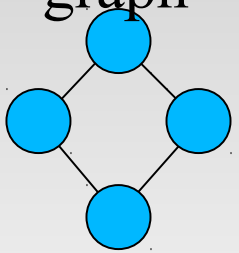
Example

model



$$\Theta = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

graph



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

8 correspondences each with probability 1/8

$$M = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

Sample 1->1 (p=0.25)

$$M_1^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix}$$

Sample 2->4 (p=0.5)

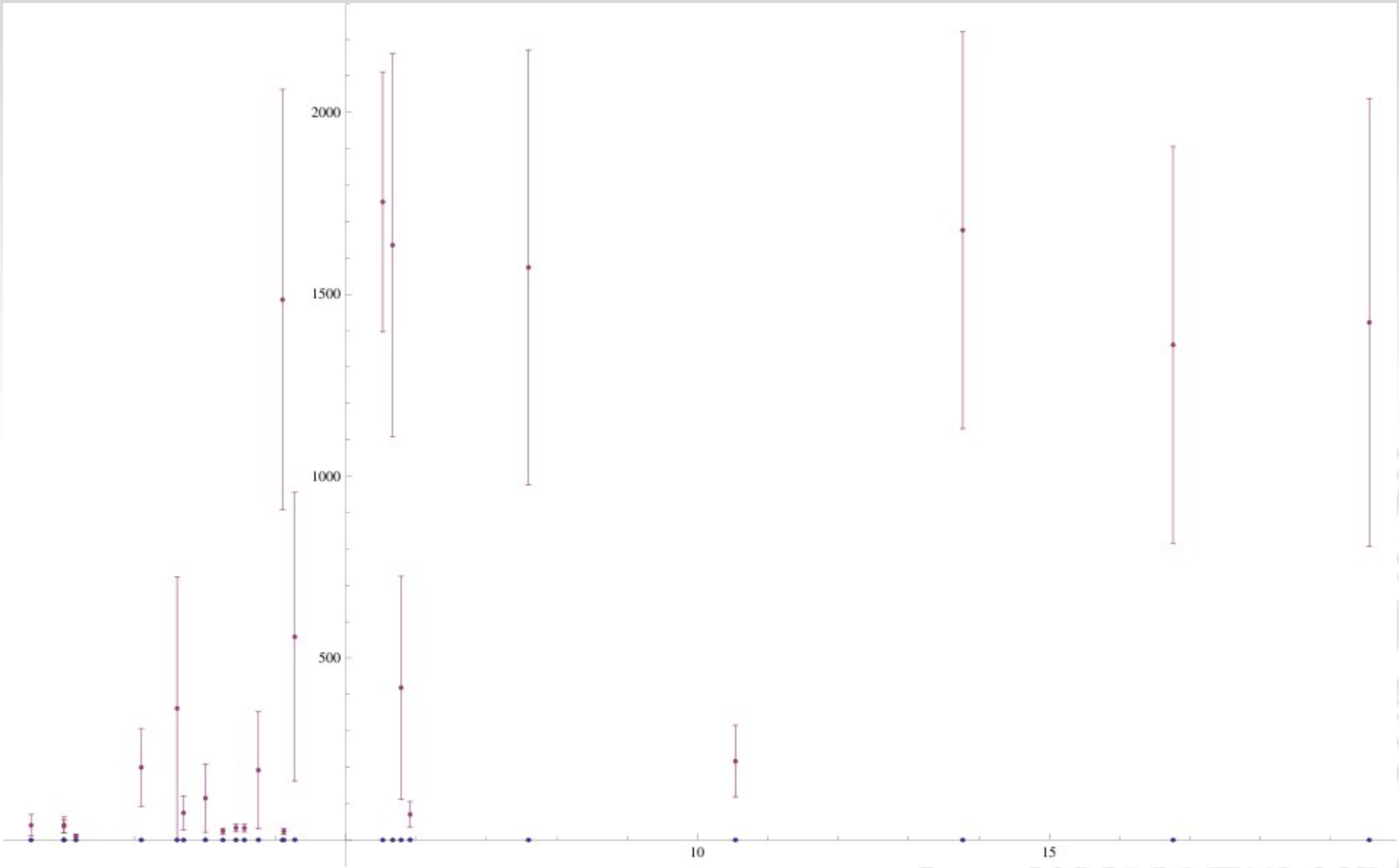
$$M_{1,2}^{1,4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Sampler Vs. Model Variance

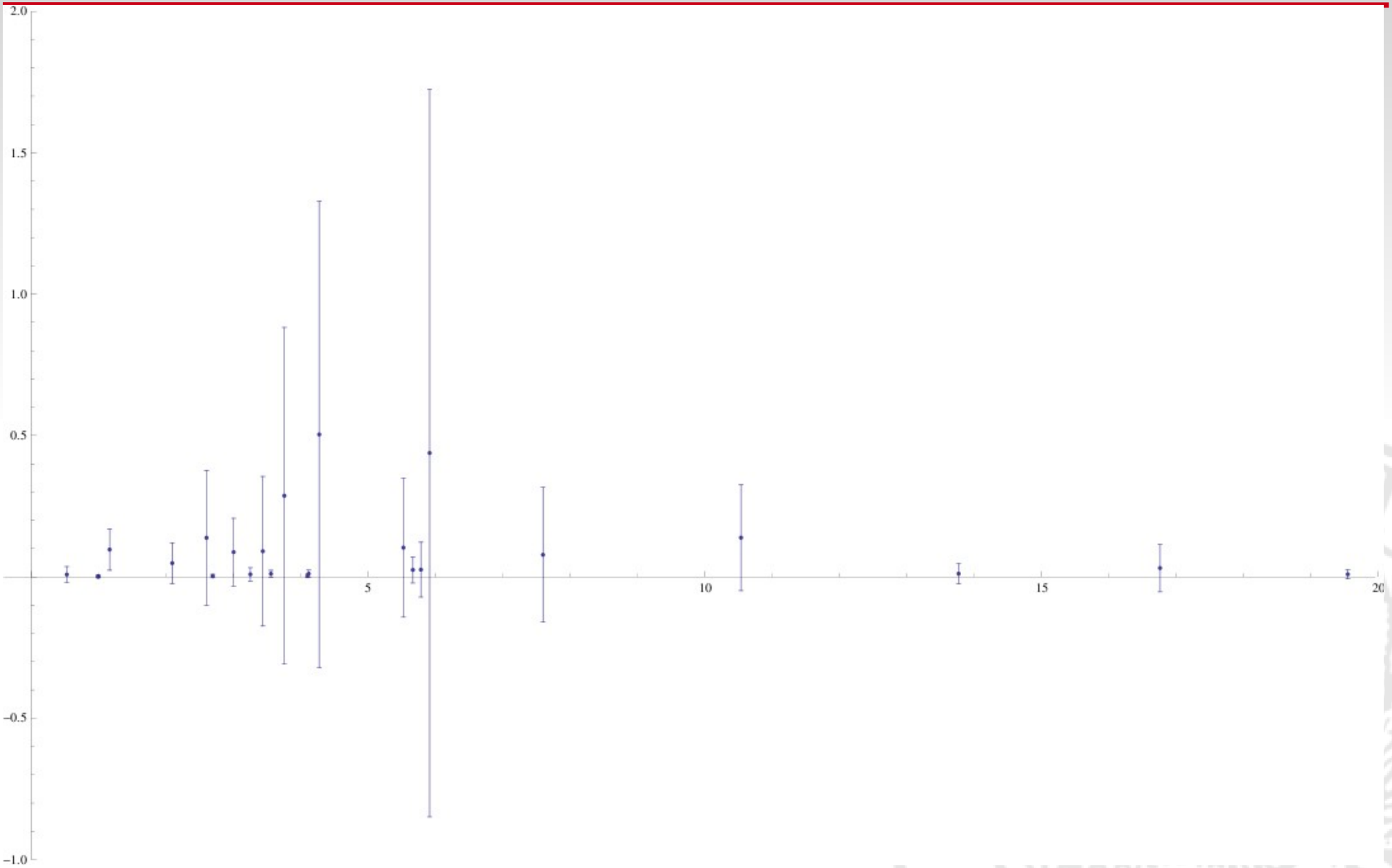
How close is the sampler distribution to the posterior of the correspondences?

- For very high entropy models both the distribution will be approximately uniform
- For deterministic models the sampler becomes equivalent to the labeling procedure used by Babai, Erdos and Selkow to show that graph isomorphism is expected polynomial time.
  - Convergence depend on degree distribution (heterogenity)
    - Use sampler to estimate $|\Sigma_g|$
- We expect a peak at medium entropy levels, but in general approximates well posteriors when node independence assumption holds.

Sampler Variance



Relative variance (σ^2/μ^2) of Montecarlo (red) and importance-sampling (blue) as a function of model entropy


2.0
1.5
1.0
0.5
−0.5
−1.0
5
10
15
20

Learning the Model

Starting from an initially oversized model, we prune the number and size of the mixture components until we get an optimal model

We adopt a MML approach to guide model pruning

- Minimized the joint cost of a two part message
  - describing a probabilistic model for the data
  - describing the data given the model

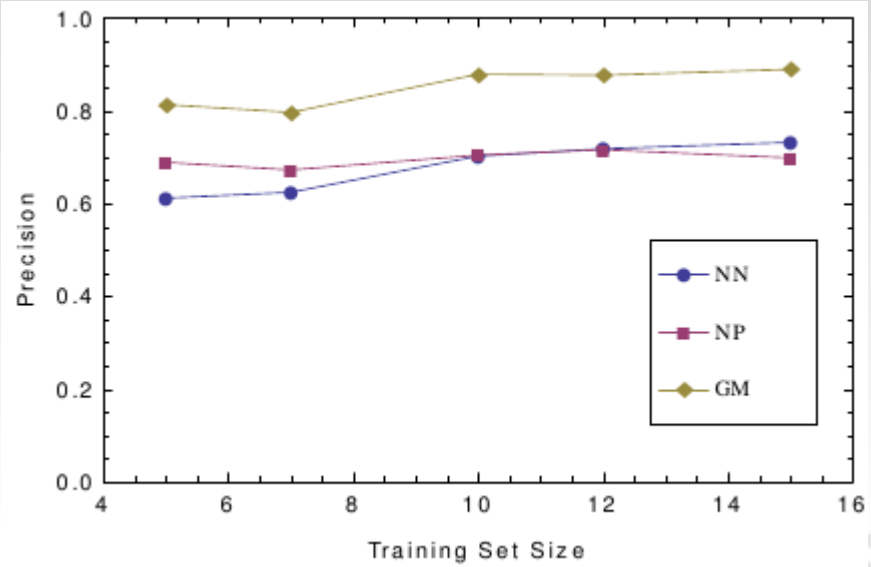
We choose the model that corresponds to the shortest two-part message

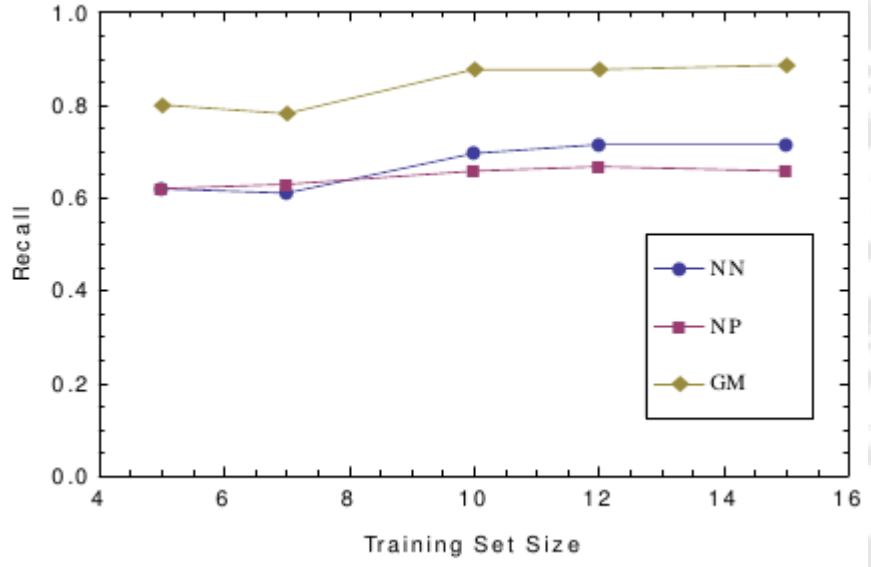
$$L_{\mathcal{M}} = \frac{1}{2}\log(\pi D) - 1 - \frac{D}{2}\log(2\pi) - \log D! + \log|\Sigma_{\mathcal{M}}|$$

Model symmetries

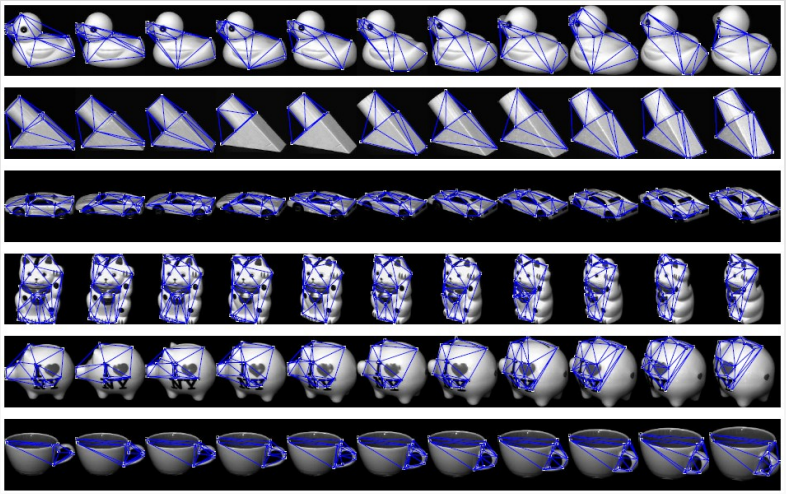
$$L_D = \frac{D}{2}\log(|S|) - \sum_{g \in S}\log(P(\hat{g}|\mathcal{G}))$$

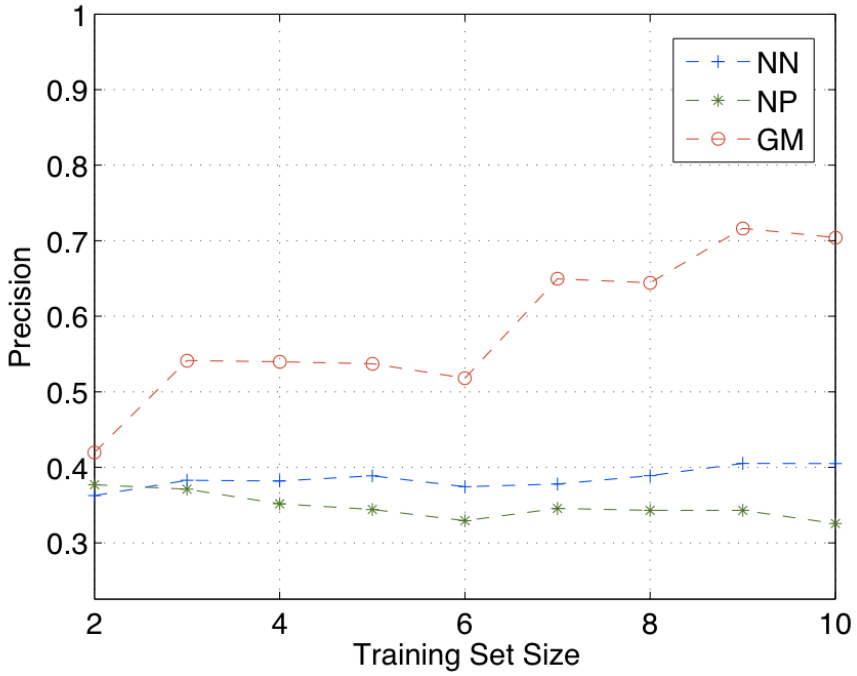
Shock Graphs: Precision and Recall

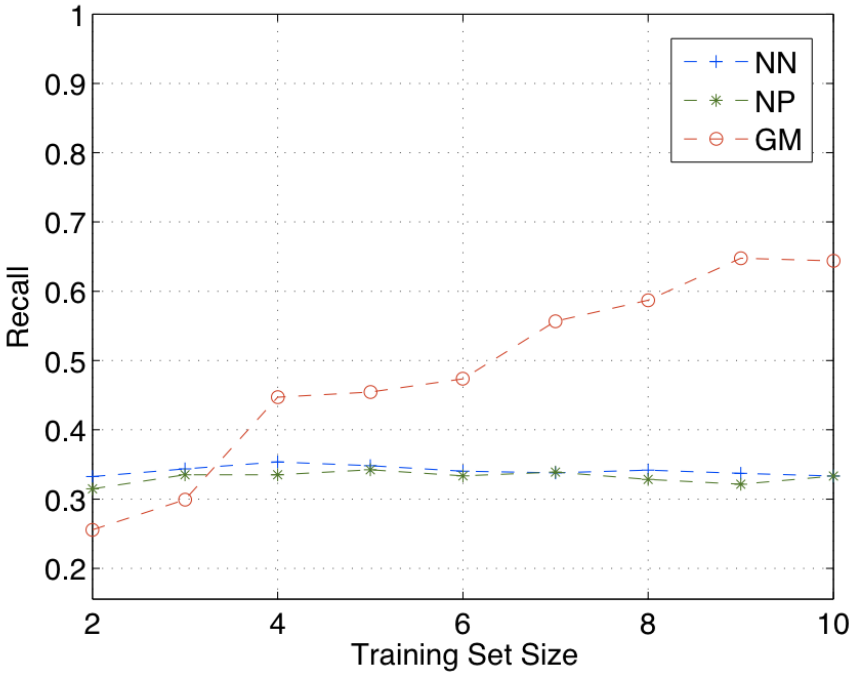




COIL-20: Precision and Recall



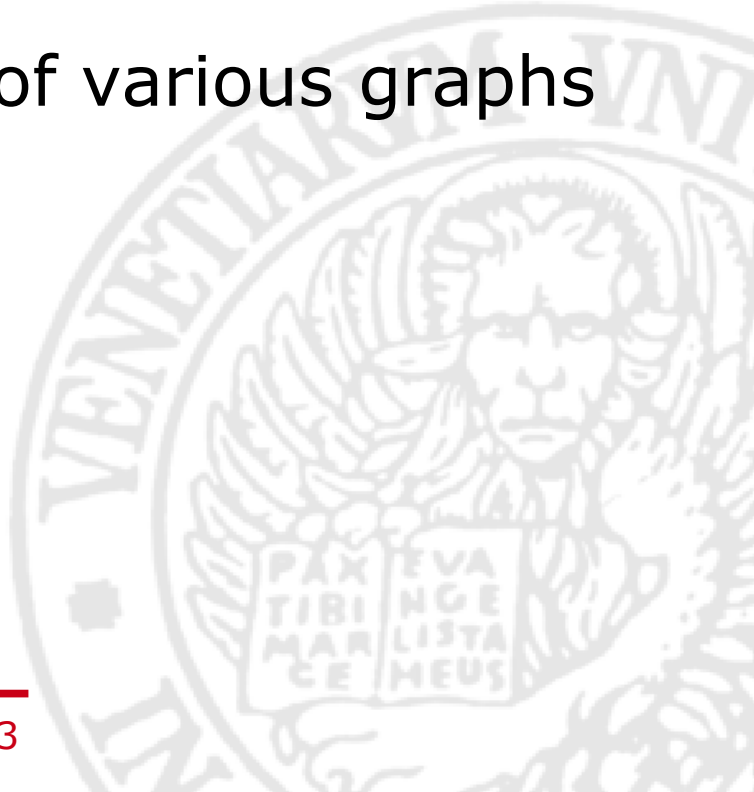




Characterizing Symmetries

Work in progress!

- (partial) Symmetries are central to the characterization of model complexity
- Try to characterize symmetries of various graphs



Quantum walks

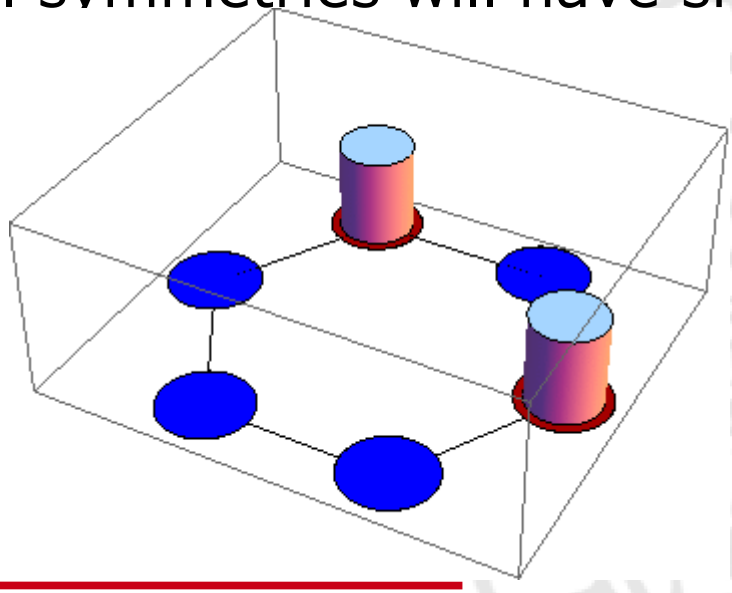
- In his work David Emms used the interference patterns in (continuous time) quantum walks to match graphs
  - Constructed an auxiliary structure by adding nodes corresponding to matches
  - When graphs are isomorphic destructive interference kept 0 amplitudes at the correct matches
- We use a similar approach to characterize axial symmetries in graphs

Quantum Walk

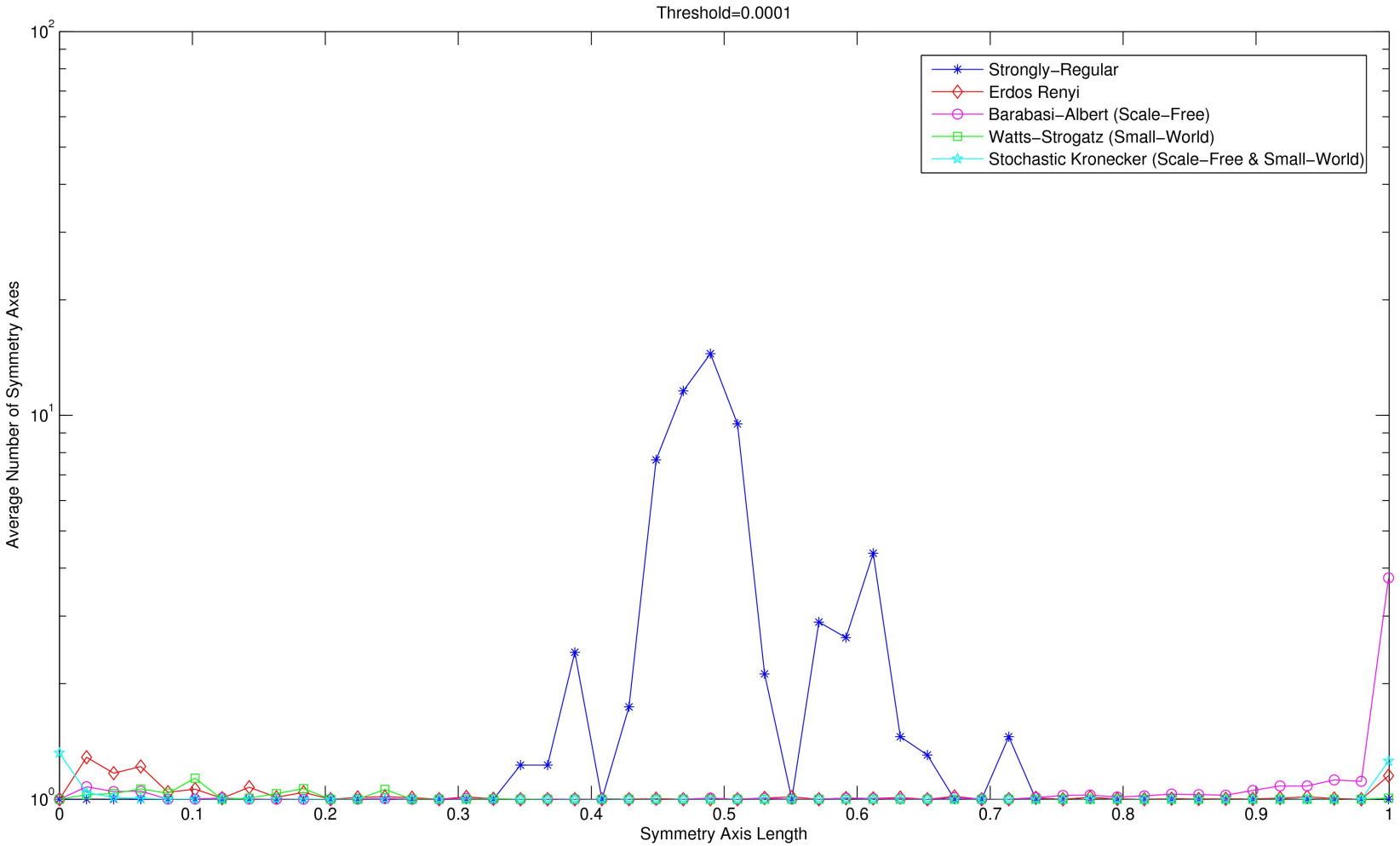
- Continuous-time quantum walk

$$U(t)=\exp(-i\mathcal{L}t)$$

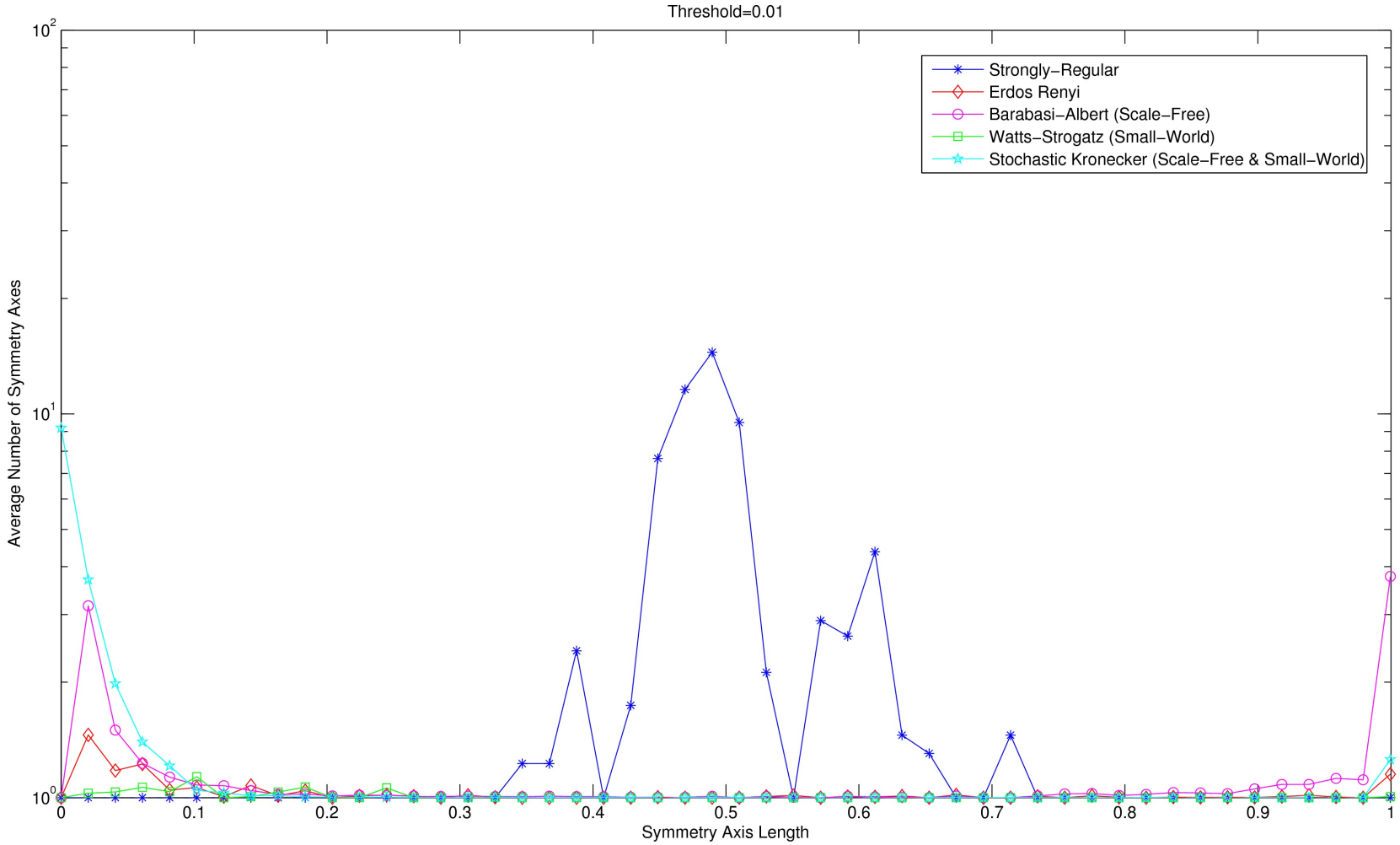
- Start the walk with two nodes with opposite amplitudes and the rest 0
- The nodes on an axial symmetry with respect to those nodes will have 0 amplitude at all times
  - (Hopefully)partial symmetries will have smaller amplitudes



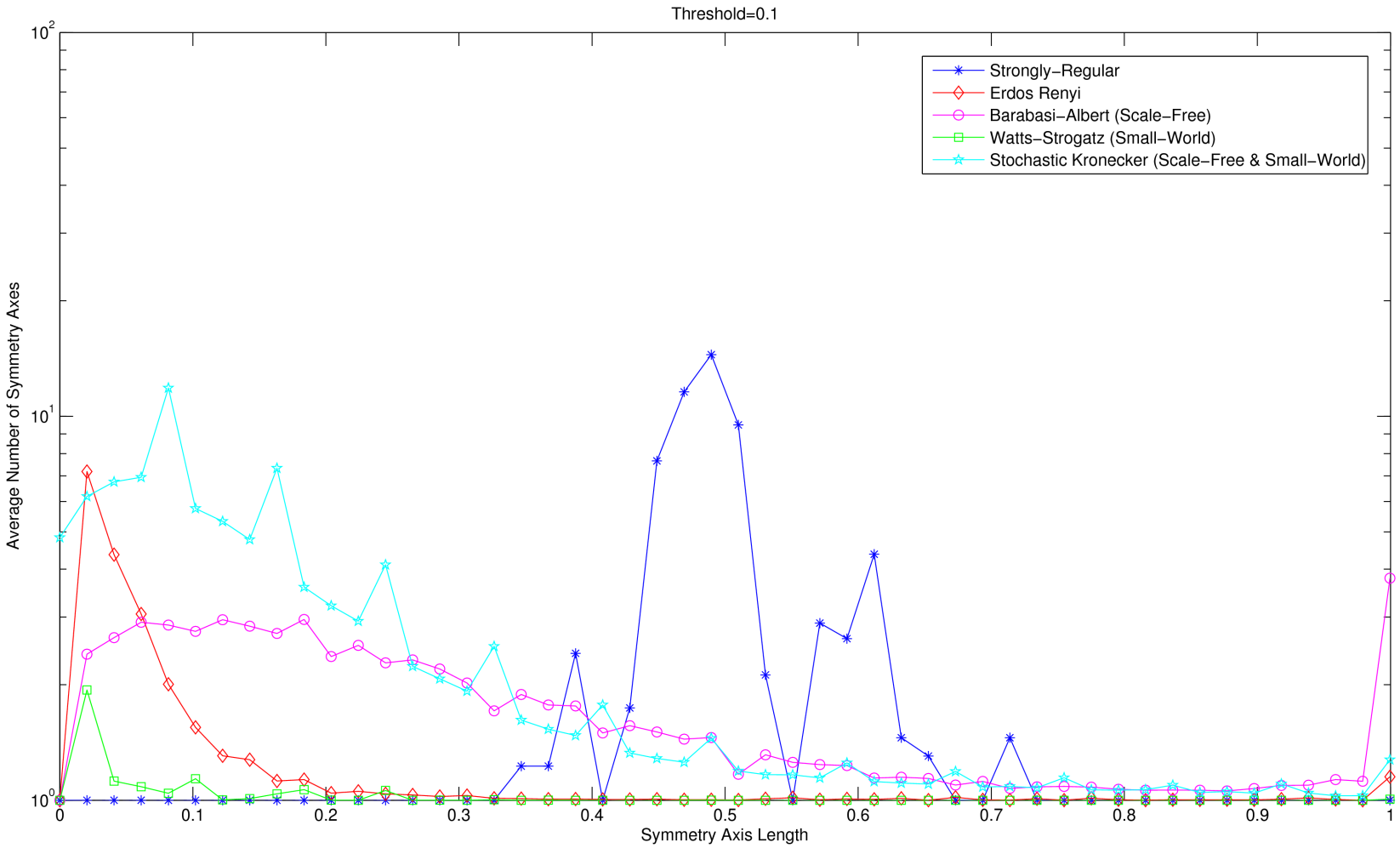
Axial Symmetries – threshold 0.001



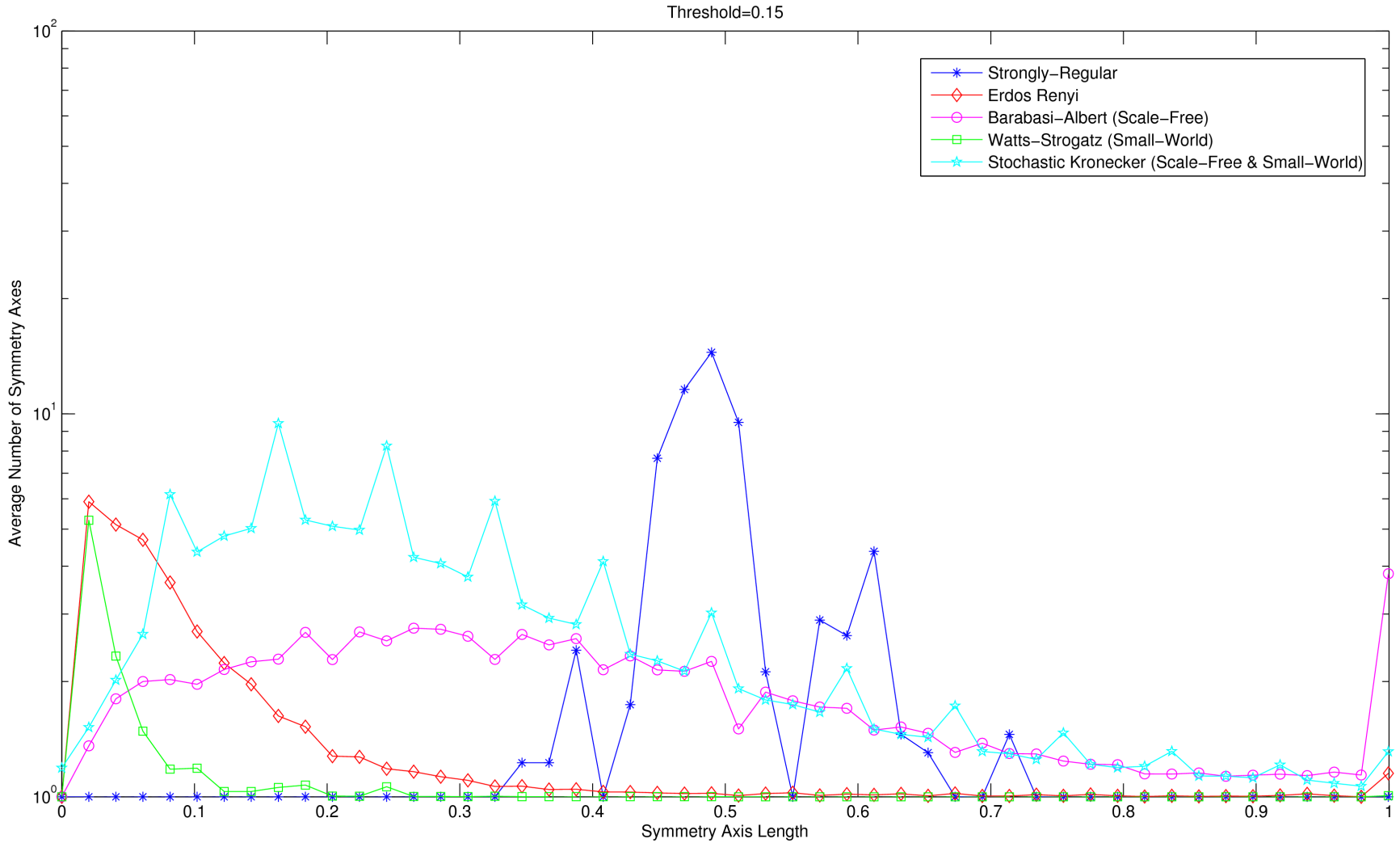
Axial Symmetries – threshold 0.01



Axial Symmetries – threshold 0.1



Axial Symmetries – threshold 0.15



Axial Symmetries – threshold 0.2