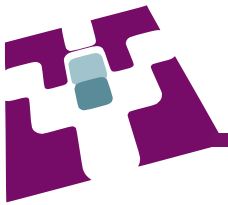
Two New Graph Kernels for Chemoinformatics

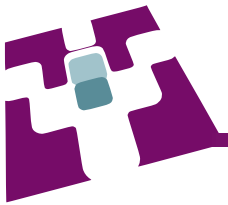
Benoit Gaüzère, Luc Brun and Didier Villemin

GREYC, CRNS UMR 6072 LCMT, CNRS UMR 6507
ENSICAEN
benoit.gauzere@ensicaen.fr **luc.brun@ensicaen.fr** **didier.villemin@ensicaen.fr**

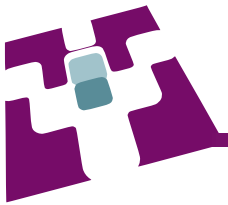
May 19, 2011



Overview



Outline



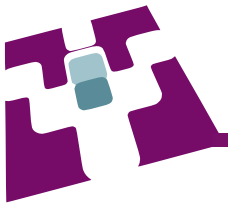
Quantitative Structure Activity Relationship

Similarity Principle [Johnson and Maggiora, 1990]

"Similar compounds have similar properties"

QSAR Applications

- Activity Prediction
- Property Prediction (QSPR)
- Virtual Screening
- Drug Design



Quantitative Structure Activity Relationship

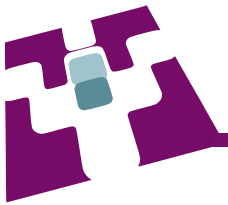
Similarity Principle [Johnson and Maggiora, 1990]

"Similar compounds have similar properties"

QSAR Applications

- Activity Prediction
- Property Prediction (QSPR)
- Virtual Screening
- Drug Design

$\Rightarrow$ Similarity Measure Between Molecules



Fingerprint Methods

Descriptor Set Comparison

- \+ Many Descriptors available
- – Remains Heuristics
- – Information Loss

Similarity Between Molecules

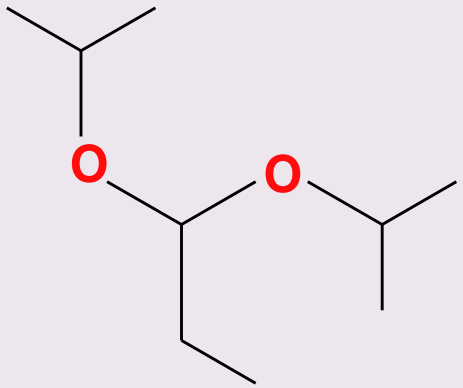
Fingerprint Methods

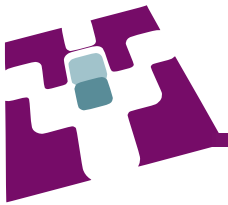
Descriptor Set Comparison

- \+ Many Descriptors available
- – Remains Heuristics
- – Information Loss

Graph Kernels

- Kernels on Molecular Graphs:
  - Vertices ⇒ Atoms
  - Edges ⇒ Bonds



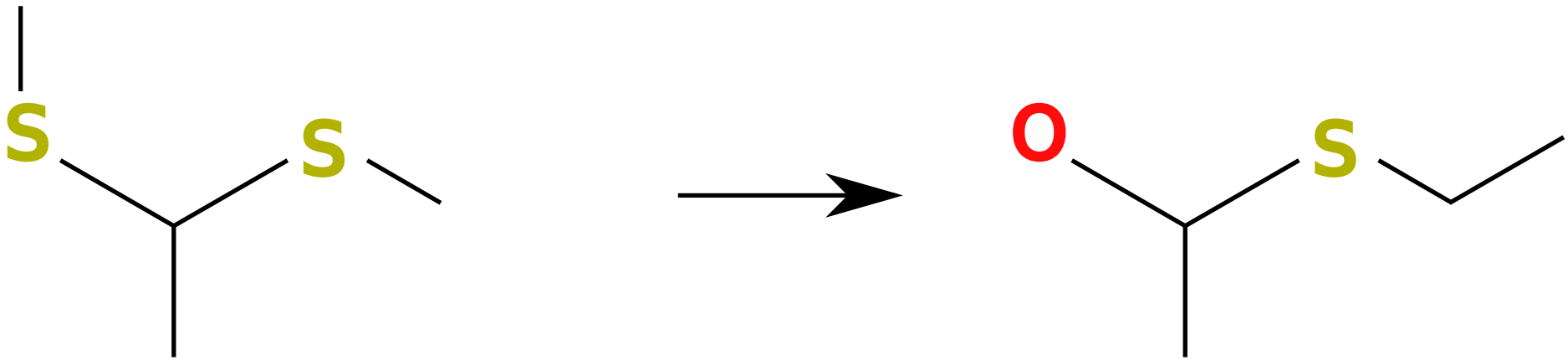


Outline

Kernel from Edit Distance

Edit Distance

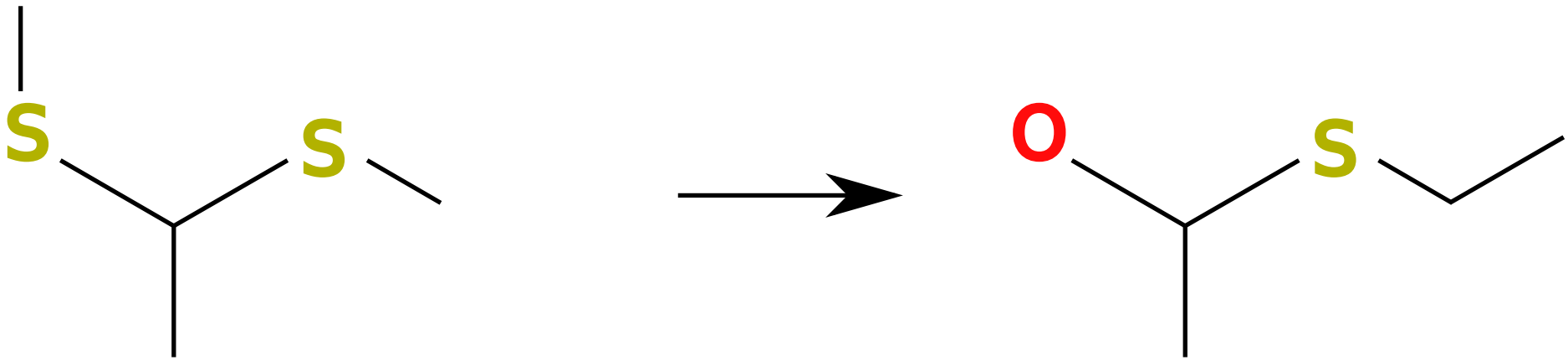
- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$



Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

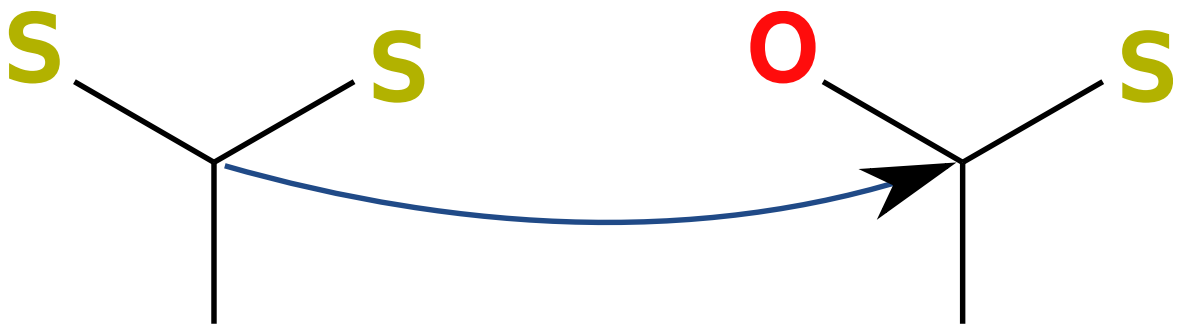


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

$$C(u \rightarrow u') = C(l(u) \rightarrow l(u'))+$$

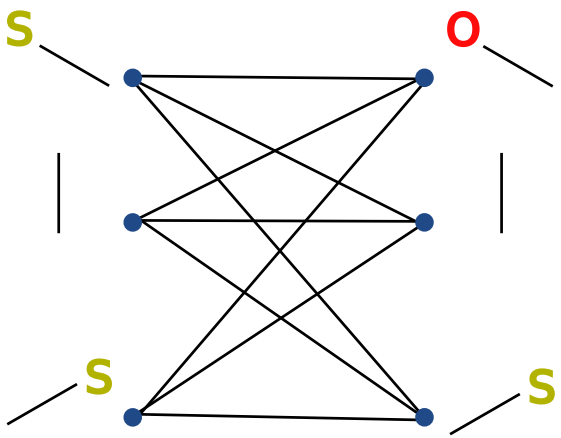


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

$$C(u \rightarrow u') = C(l(u) \rightarrow l(u'))+$$

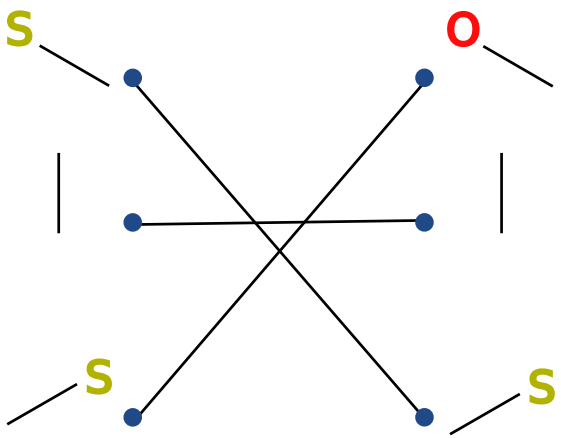


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

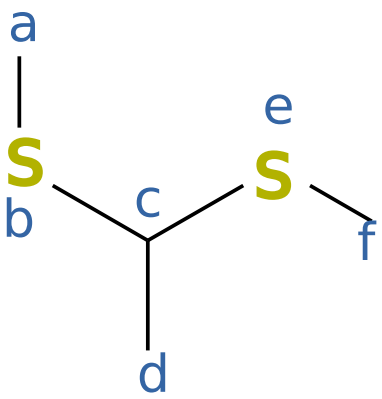
$$C(u \to u') = C(l(u) \to l(u')) + \arg\min \sum C(l(e) \to l(e'))$$

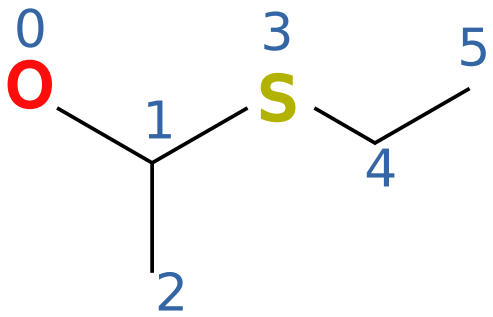


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

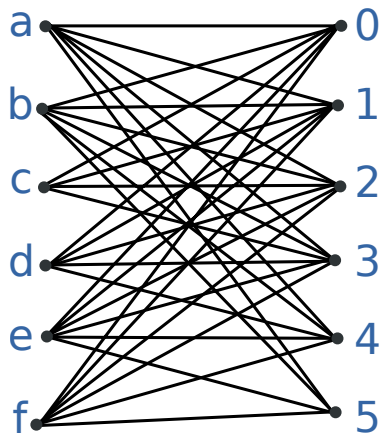




Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

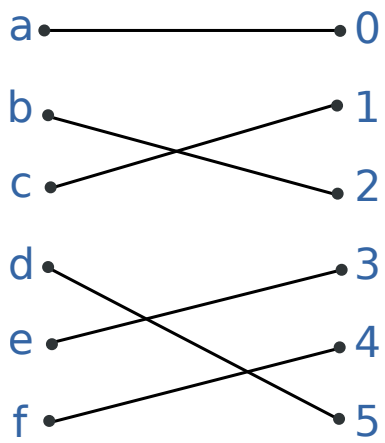


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]

$$\text{Sub Optimal Edit Distance} = \arg\min \sum C(u \rightarrow u')$$

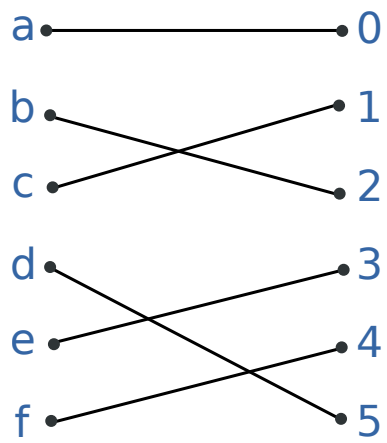


Kernel from Edit Distance

Edit Distance

- Sequence of Edit Operations
- Edit Distance $= \arg\min \sum Cost(operation)$
- Sub-optimal Distance Computable in polynomial time [Riesen and Bunke, 2009]
- Trivial Kernels may not be Definite Positive

$$\text{Sub Optimal Edit Distance} = \arg\min \sum C(u \rightarrow u')$$



Minimization Problem

Gram Matrix K

$$K_{i,j} = \langle G_i \cdot G_j \rangle = k(G_i, G_j)$$

Minimization Problem [Steinke and Schölkopf, 2008]

$$f^* = \arg\min_{f \in \mathbb{R}^n} C\,Loss(f, y, K) + f^t K^{-1} f$$

- $Loss(f, y, K)$: Fit to Data Term
- $f^t K^{-1} f$: Regularization Term

Regularization operator $\tilde{\ell}$

Laplacian of a Graph

Laplacian of a Graph:

$$\ell = \Delta - W$$

- $\Delta_{i,i} = \sum_{j=1}^{n} W_{i,j}$
- $W_{ij} = e^{-\frac{d(G_i, G_j)}{\sigma}}$

Regularization operator $\tilde{\ell}$

Laplacian of a Graph

Laplacian of a Graph:

$$\ell = \Delta - W$$

- $\Delta_{i,i} = \sum_{j=1}^{n} W_{i,j}$
- $W_{ij} = e^{-\frac{d(G_i, G_j)}{\sigma}}$

Regularized Laplacian [Smola and Kondor, 2003]

$$\tilde{\ell} = I + \lambda \ell$$

Regularization operator $\tilde{\ell}$

Laplacian of a Graph

Laplacian of a Graph:

$$\ell = \Delta - W$$

- $\Delta_{i,i} = \sum_{j=1}^{n} W_{i,j}$
- $W_{ij} = e^{-\frac{d(G_i, G_j)}{\sigma}}$

Regularized Laplacian [Smola and Kondor, 2003]

$$\tilde{\ell} = I + \lambda \ell$$

$$f^t \tilde{\ell} f = \|f\|^2 + \lambda \sum_{i,j=1}^{n} W_{ij}(f_i - f_j)^2$$

Graph Laplacian Kernel

$S = \{G_1, \ldots, G_n\}$,

$$K_S = \tilde{\ell}_S^{-1} \Rightarrow \mathcal{O}(N^3)$$

Graph Laplacian Kernel

$S = \{G_1, \ldots, G_n\}$,

$$K_S = \tilde{\ell}_S^{-1} \Rightarrow \mathcal{O}(N^3)$$

$$K_{S \cup G} = (\tilde{\ell}_{S \cup G})^{-1} \Rightarrow \mathcal{O}((N+1)^3)$$

Graph Laplacian Kernel

Graph Laplacian Kernel

$S = \{G_1, \ldots, G_n\}$,

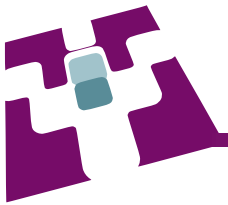
$$K_S = \tilde{\ell}_S^{-1} \Rightarrow \mathcal{O}(N^3)$$

$$K_{S \cup G} = (\tilde{\ell}_{S \cup G})^{-1} \Rightarrow \mathcal{O}((N+1)^3)$$

Efficient Update

$$\left(\tilde{\ell}_{S \cup G}\right)^{-1} = \begin{pmatrix} \tilde{\ell}_S - \delta_S & B \\ B^t & 1 - \sum_i B_i \end{pmatrix}^{-1}$$

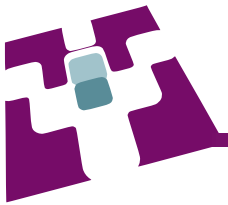
- Blockwise Inversion $\Rightarrow \tilde{\ell}_S - \delta_S$ to Invert
- Approximation of $(\tilde{\ell}_S - \delta_S)^{-1}$ in $\mathcal{O}(CN^2)$



Graph Laplacian Kernel

Conclusion

- Based on Edit Distance
- Definite Positive Kernel
- Too Costly for Large Databases
- Dependant Rely on the Accuracy of Sub Optimal Edit Distance



Outline

Bag of Patterns Kernel

Principle

- Extract Patterns
- Bag of Patterns Comparison

$$K(x,y) = \sum_{x_i \in B(x)} \sum_{y_i \in B(y)} k(x_i, y_i)$$

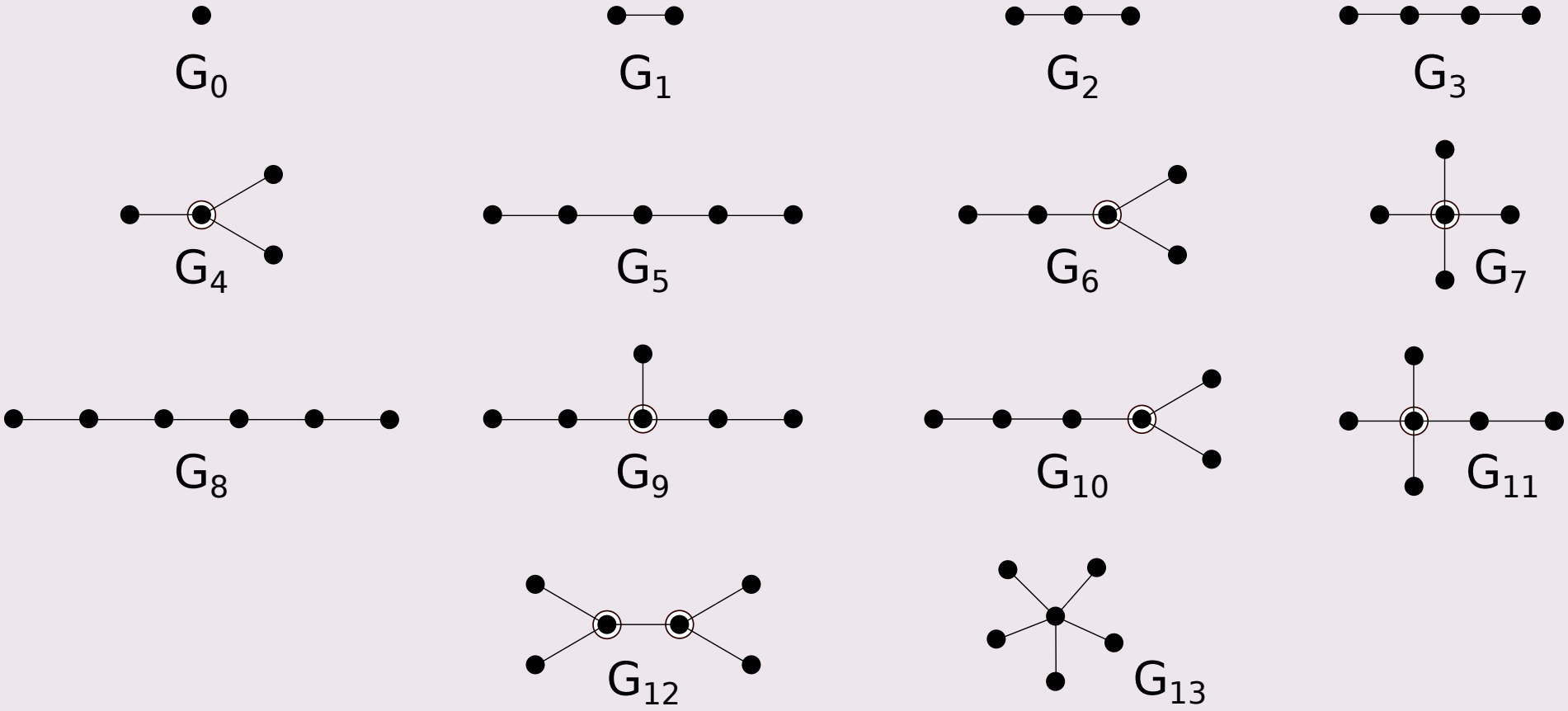
Patterns:

- Random Walks
- Trails, Paths
- Tree-Patterns

Treelet Kernel

Patterns Enumerated

All Acyclic Unlabeled Graphs with a Size < 7:



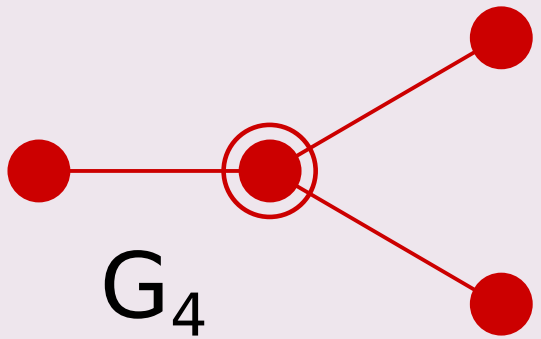
Treelet Enumeration Method

Path Enumeration

Recursive Search from Each Node

Non Linear Treelets Enumeration

Detection of *n-star* $\Rightarrow$ Enumeration of derived treelets



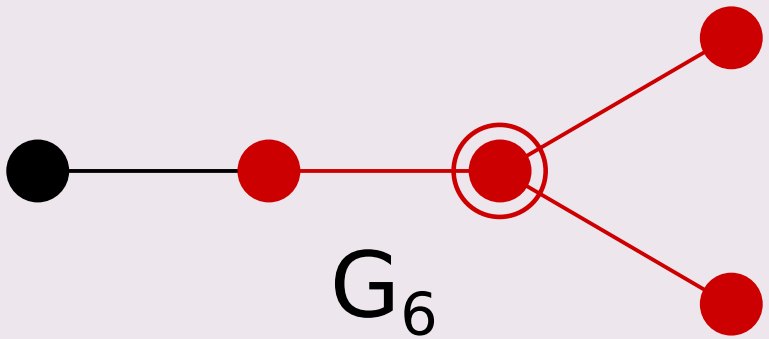
Treelet Enumeration Method

Path Enumeration

Recursive Search from Each Node

Non Linear Treelets Enumeration

Detection of *n-star* $\Rightarrow$ Enumeration of derived treelets



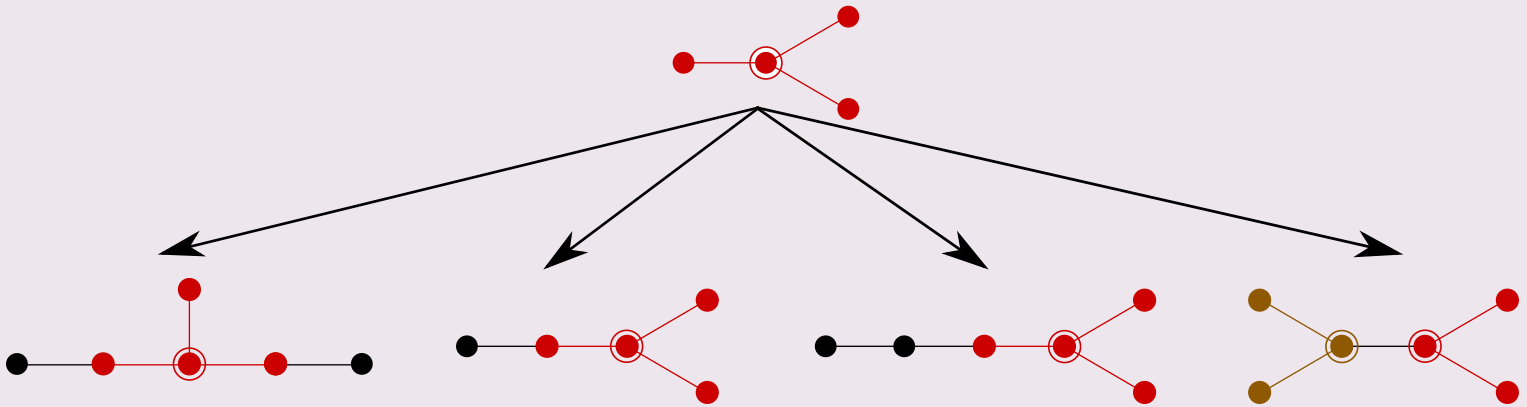
Treelet Enumeration Method

Path Enumeration

Recursive Search from Each Node

Non Linear Treelets Enumeration

Detection of *n-star* $\Rightarrow$ Enumeration of derived treelets



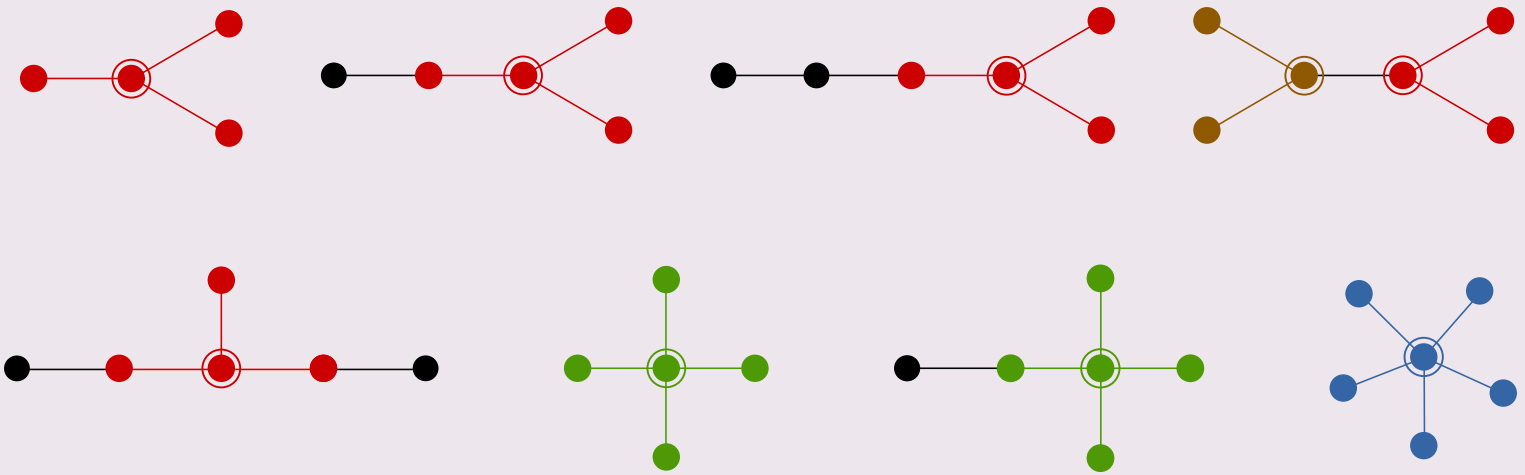
Treelet Enumeration Method

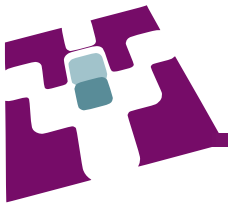
Path Enumeration

Recursive Search from Each Node

Non Linear Treelets Enumeration

Detection of *n-star* $\Rightarrow$ Enumeration of derived treelets





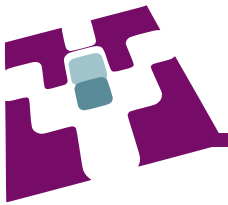
Treelet Kernel Definition

Spectrum of a graph

$$\text{Spectrum}_i(G) = |(G_i \subseteq G)|$$

Treelet Kernel Definition

$$k_{\text{Treelet}}(G, G') = \sum_{k=0}^{N} e^{-\frac{(Spectrum_k(G) - Spectrum_k(G'))^2}{\sigma}}$$



Outline

Experiments on Alkane Dataset

Alkane Dataset

- Acyclic Molecules
- Only Carbons and Hydrogens
- $\Rightarrow$ Acyclic Unlabeled Graphs
- 150 Molecules
- Boiling Point Prediction Problem
- Kernel Ridge Regression

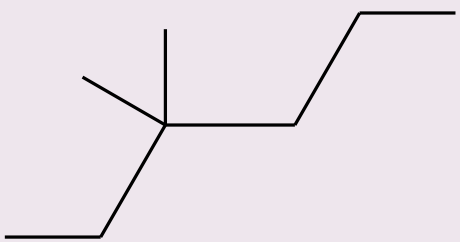
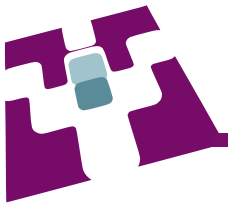


Figure: Alkane Molecule



Results on Alkane Dataset

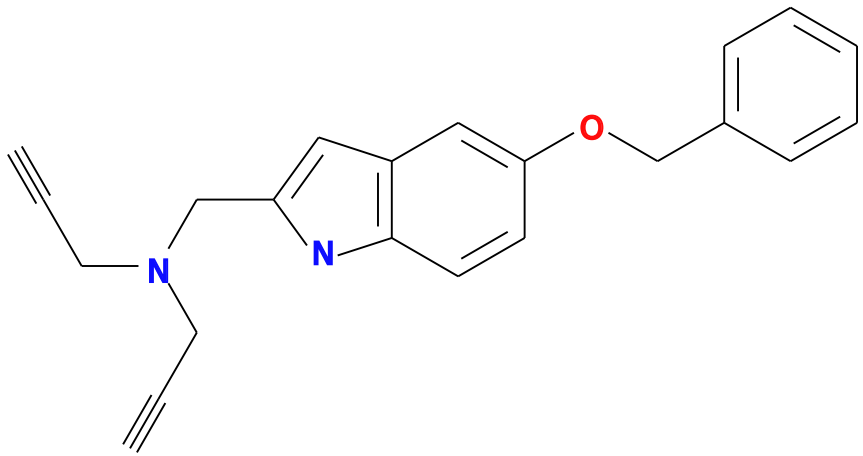
Table: Boiling Point Prediction on Alkane Dataset

Method	Average Error (°C)	MSE (°C)
Neural Network	3.11453	3.69993
KMean	4.65536	6.20788
Random Walk Kernel	10.6084	16.2799
Graph Laplacian Kernel	10.7948	16.4484
Treelet Kernel	1.40663	1.91695

MAO Dataset

MAO Dataset

- Cyclic Molecules with Heteroatoms
- 68 Molecules
- Monoamine Oxidase Inhibitor Problem
- $\Rightarrow$ Classification Problem
- 2-Class SVM



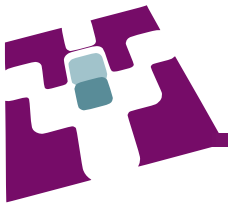
MAO Dataset

Table: Classification Accuracy on MAO Dataset

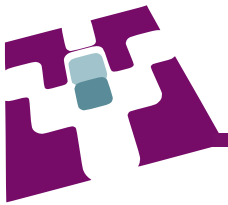
Method	Classification Accuracy
KMean	80% (55/68)
Random Walk Kernel	82% (56/68)
KWMean	88% (60/68)
Gaussian Kernel on Edit Distance	90% (61/68)
Standard Graph Laplacian Kernel	90% (61/68)
Fast Graph Laplacian Kernel	90% (61/68)

Inversion Computing Time:

- Standard Update: 0.498 ms
- Fast Update: 0.273 ms



Outline



Conclusion

Graph Laplacian Kernel

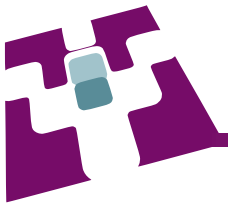
+ Definite Positive Kernel Based on Edit Distance
+ Reduction of Update Complexity
– Poor Results on Unlabeled Graphs
– Heuristics Required for Cost Operations

Treelet Kernel

+ Exhaustive Count of Unlabeled Acyclic Subgraphs
+ Non Linear Patterns Comparison
– Limited to Unlabeled Acyclic Graphs

Future Work

- Labeled Treelets
- Cycles



End

Thank You for your Attention.


Johnson, M. A. and Maggiora, G. M., editors (1990).
Concepts and Applications of Molecular Similarity.
Wiley.


Riesen, K. and Bunke, H. (2009).
Approximate graph edit distance computation by means of bipartite graph matching.
Image and Vision Computing, 27(7):950–959.


Smola, A. and Kondor, R. (2003).
Kernels and regularization on graphs.
In *Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, page 144. Springer Verlag.


Steinke, F. and Schölkopf, B. (2008).
Kernels, regularization and differential equations.
Pattern Recogn., 41:3271–3286.