Graph Embedding in Vector Spaces

GbR'2011 Mini-tutorial

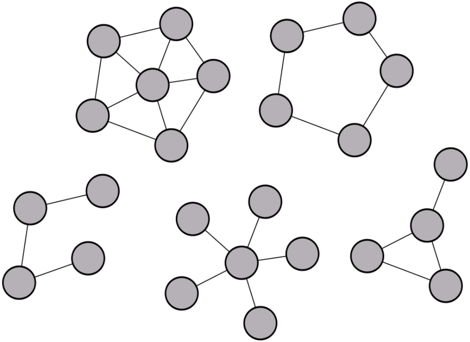
Jaume Gibert, Ernest Valveny
Computer Vision Center,
Universitat Autònoma de Barcelona,
Barcelona, Spain

Horst Bunke
Institute of Computer Science and Applied Mathematics,
University of Bern,
Bern, Switzerland

What do we understand by Graph Embedding?

Motivational problem

Given a set of graphs to be categorized, how do we process them?



What do we understand by Graph Embedding?

- A possible and intuitive solution:
  Consider a similarity measure between the input graphs and apply kNN.

What do we understand by Graph Embedding?

- A possible and intuitive solution:
  Consider a similarity measure between the input graphs and apply kNN.

$\longrightarrow$ There is not really any other option in the graph domain.

What do we understand by Graph Embedding?

- A possible and intuitive solution:
  Consider a similarity measure between the input graphs and apply kNN.

$\longrightarrow$ There is not really any other option in the graph domain.

- Adapt Neural Networks and, more generally Graphical Models, to graph input patterns

$\longrightarrow$ *A General Framework for Adaptive Processing of Data Structures*
Paolo Frasconi, Marco Gori, Alessandro Sperduti (Neural Networks 1998).

What do we understand by Graph Embedding?

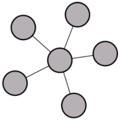
- A possible and intuitive solution:
  Consider a similarity measure between the input graphs and apply kNN.

⟶ There is not really any other option in the graph domain.

- Adapt Neural Networks and, more generally Graphical Models, to graph input patterns

⟶ *A General Framework for Adaptive Processing of Data Structures*
Paolo Frasconi, Marco Gori, Alessandro Sperduti (Neural Networks 1998).

- The solution we are here interested in:
  Assign a feature vector to every graph.

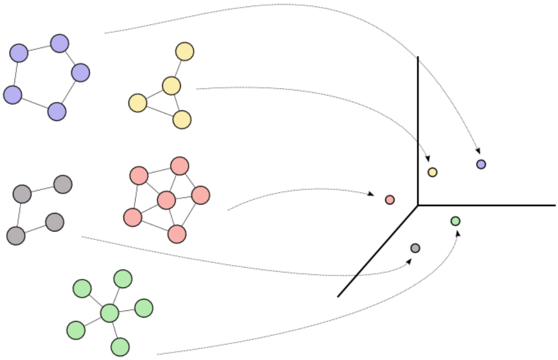


$v = (f_1, f_2, \ldots, f_n) \in \mathbb{R}^n$

What do we understand by Graph Embedding?

Formally, a graph embedding is a mapping from the set of graphs to a vectorial space

$$\begin{aligned} \phi : G &\longrightarrow \mathbb{R}^n \\ g &\longmapsto \phi(g) = (f_1, f_2, \ldots, f_n) \end{aligned}$$



Possible misunderstandings

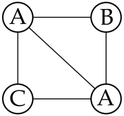
- We do not want to draw a graph in the 2D plane.

- Graph kernels are an implicit way of defining a graph embedding (more on that tomorrow).

Crucial Issue

Which features do we extract from graphs?

Simple features

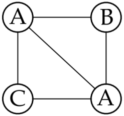


Crucial Issue

Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges

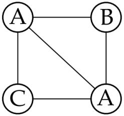


Crucial Issue

Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges
- Number of nodes with label A, or label B,...

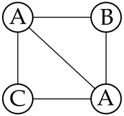


Crucial Issue

Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges
- Number of nodes with label A, or label B,...
- Number of edges between label A and label C,...

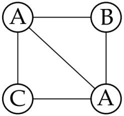


Crucial Issue

Which features do we extract from graphs?

Simple features

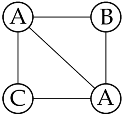
- Number of nodes, number of edges
- Number of nodes with label A, or label B,...
- Number of edges between label A and label C,...
- Average degree of the nodes



Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges
- Number of nodes with label A, or label B,...
- Number of edges between label A and label C,...
- Average degree of the nodes
- Number of cycles of a certain length
- ...

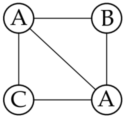


Crucial Issue

Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges
- Number of nodes with label A, or label B,...
- Number of edges between label A and label C,...
- Average degree of the nodes
- Number of cycles of a certain length
- ...



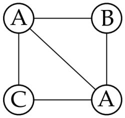
$\longrightarrow$ Are these features discriminative enough? Is there another way to get more features?

Crucial Issue

Which features do we extract from graphs?

Simple features

- Number of nodes, number of edges
- Number of nodes with label A, or label B,...
- Number of edges between label A and label C,...
- Average degree of the nodes
- Number of cycles of a certain length
- ...



$\longrightarrow$ Are these features discriminative enough? Is there another way to get more features?

Let us review the literature.

Literature Review

- Substructure finding methods
- Spectral methods
- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data (Inokuchi et al., PKDD 2000)

- Spectral methods

- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)

- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory
     (Wilson, Hancock and Luo, TPAMI 2005)

- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory
     (Wilson, Hancock and Luo, TPAMI 2005)
  3. Graph Characterization via Ihara Coefficients
     (Ren, Wilson and Hancock, Neural Networks 2011)

- Dissimilarity Representation

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory
     (Wilson, Hancock and Luo, TPAMI 2005)
  3. Graph Characterization via Ihara Coefficients
     (Ren, Wilson and Hancock, Neural Networks 2011)

- Dissimilarity Representation

  1. MDS on the dissimilarity matrix of a set of graphs

Literature Review

- Substructure finding methods

  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)

- Spectral methods

  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory
     (Wilson, Hancock and Luo, TPAMI 2005)
  3. Graph Characterization via Ihara Coefficients
     (Ren, Wilson and Hancock, Neural Networks 2011)

- Dissimilarity Representation

  1. MDS on the dissimilarity matrix of a set of graphs
  2. Graph Embedding using Constant Shift Embedding
     (Jouili and Tabbone, ICPR 2010)

Literature Review

- Substructure finding methods
  1. An Apriori-based algorithm for mining frequent substructures from graph data
     (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations
     (Sidère et al., SSPR 2008)
- Spectral methods
  1. Spectral Embedding of Graphs
     (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory
     (Wilson, Hancock and Luo, TPAMI 2005)
  3. Graph Characterization via Ihara Coefficients
     (Ren, Wilson and Hancock, Neural Networks 2011)
- Dissimilarity Representation
  1. MDS on the dissimilarity matrix of a set of graphs
  2. Graph Embedding using Constant Shift Embedding
     (Jouili and Tabbone, ICPR 2010)
  3. Graph Classification based on Vector Space Embedding
     (Riesen and Bunke, IJPRAI 2009)

Literature Review

- Substructure finding methods
  1. An Apriori-based algorithm for mining frequent substructures from graph data (Inokuchi et al., PKDD 2000)
  2. A Vectorial Representation for the Indexation of Structural Informations (Sidère et al., SSPR 2008)
- Spectral methods
  1. Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)
  2. Pattern Vectors from Algebraic Graph Theory (Wilson, Hancock and Luo, TPAMI 2005)
  3. Graph Characterization via Ihara Coefficients (Ren, Wilson and Hancock, Neural Networks 2011)
- Dissimilarity Representation
  1. MDS on the dissimilarity matrix of a set of graphs
  2. Graph Embedding using Constant Shift Embedding (Jouili and Tabbone, ICPR 2010)
  3. Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Spectral Embedding of Graphs

(Luo, Wilson and Hancock, Pattern Recognition 2003)

Basic notation

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

- The eigenvalues λ_k of A_k are the solutions of $|A_k - \lambda_k I| = 0$

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

- The eigenvalues λ_k of A_k are the solutions of $|A_k - \lambda_k I| = 0$
- The eigenvectors ϕ_k^w are the solutions of $A_k \phi_k^w = \lambda_k^w \phi_k^w$, where w is the eigenmode index.

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

- The eigenvalues λ_k of A_k are the solutions of $|A_k - \lambda_k I| = 0$
- The eigenvectors ϕ_k^w are the solutions of $A_k \phi_k^w = \lambda_k^w \phi_k^w$, where w is the eigenmode index.
- The modal matrix is defined by $\phi_k = (\phi_k^1 \mid \phi_k^2 \mid \ldots \mid \phi_k^{|V_k|})$

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

- The eigenvalues λ_k of A_k are the solutions of $|A_k - \lambda_k I| = 0$
- The eigenvectors ϕ_k^w are the solutions of $A_k \phi_k^w = \lambda_k^w \phi_k^w$, where w is the eigenmode index.
- The modal matrix is defined by $\phi_k = (\phi_k^1 \,|\, \phi_k^2 \,|\, \ldots \,|\, \phi_k^{|V_k|})$
- The spectral decomposition of the adjacency matrix is

$$A_k = \sum_{i=1}^{|V_k|} \lambda_k^w \phi_k^w (\phi_k^w)^T$$

Basic notation

- Set of graphs $G_1, \ldots, G_N$, where $G_k = (V_k, E_k)$
- Adjacency matrix of G_k is defined by

$$A_k(i,j) = \begin{cases} \mathcal{W}(i,j), & \text{if } (i,j) \in E_k \\ 0, & \text{otherwise.} \end{cases}$$

- The eigenvalues λ_k of A_k are the solutions of $|A_k - \lambda_k I| = 0$
- The eigenvectors ϕ_k^w are the solutions of $A_k \phi_k^w = \lambda_k^w \phi_k^w$, where w is the eigenmode index.
- The modal matrix is defined by $\phi_k = (\phi_k^1 \,|\, \phi_k^2 \,|\, \ldots \,|\, \phi_k^{|V_k|})$
- The spectral decomposition of the adjacency matrix is

$$A_k = \sum_{i=1}^{|V_k|} \lambda_k^w \phi_k^w (\phi_k^w)^T$$

- The truncated modal matrix is defined by

$$\phi_k = (\phi_k^1 \,|\, \phi_k^2 \,|\, \ldots \,|\, \phi_k^n)$$

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

The work presented in this paper is mainly concerned on extracting features from the eigen-modes of the truncated modal matrix. In particular,

Spectral Embedding of Graphs
(Luo, Wilson and Hancock, Pattern Recognition 2003)

The work presented in this paper is mainly concerned on extracting features from the eigen-modes of the truncated modal matrix. In particular,

- Unary features:

  $\longrightarrow$ One feature for each eigen-mode.

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

The work presented in this paper is mainly concerned on extracting features from the eigen-modes of the truncated modal matrix. In particular,

- Unary features:

  $\longrightarrow$ One feature for each eigen-mode.

- Binary features:

  $\longrightarrow$ One feature for each pair of eigen-modes.

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

- Unary features (among others)

Spectral Embedding of Graphs
(Luo, Wilson and Hancock, Pattern Recognition 2003)

- Unary features (among others)
  - Leading eigenvalues

    Consider all eigenvalues as features for the vectorial representation of G_k:

$$B_k = (\lambda_k^1, \lambda_k^2, \ldots, \lambda_k^n)^T$$

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

- Unary features (among others)
  - Leading eigenvalues

    Consider all eigenvalues as features for the vectorial representation of G_k:

    $$B_k = (\lambda_k^1, \lambda_k^2, \ldots, \lambda_k^n)^T$$

  - Eigen-mode volume

    Let $D_k(i)$ be the degree of the node i in the graph G_k. The volume of the eigenmode w is defined as

    $$Vol_k(w) = \sum_{i \in V_k} \phi_k(i, w) D_k(i).$$

    As a feature vector for G_k, we define

    $$B_k = (Vol_k(1), Vol_k(2), \ldots, Vol_k(n))^T$$

Spectral Embedding of Graphs
(Luo, Wilson and Hancock, Pattern Recognition 2003)

- Binary features

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

- Binary features
  - Inter-mode adjacency matrix

    Project the adjacency matrix onto the basis of eigenvectors

    $$U_k = \phi_k^T A_k \phi_k$$

    The vectorial representation of the graph G_k is defined by $B_k = (U_k(1,1), U_k(1,2), \ldots, U_k(n,n))$, where

    $$U_k(u,v) = \sum_{i \in V_k} \sum_{j \in V_k} \phi_k(i,u)\phi_k(j,v)A_k(i,j).$$

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

- Binary features
  - Inter-mode adjacency matrix

    Project the adjacency matrix onto the basis of eigenvectors

    $$U_k = \phi_k^T A_k \phi_k$$

    The vectorial representation of the graph G_k is defined by $B_k = (U_k(1,1), U_k(1,2), \ldots, U_k(n,n))$, where

    $$U_k(u,v) = \sum_{i \in V_k} \sum_{j \in V_k} \phi_k(i,u)\phi_k(j,v)A_k(i,j).$$

  - Inter-mode distance

    The leading node (most important) in the eigenmode u is defined by

    $$i_u^k = \underset{i \in V_k}{\operatorname{argmax}}\, \phi_k(i,u)$$

    The vectorial representation of the graph G_k is defined by $B_k = (d_{1,1}, d_{1,2}, \ldots, d_{n,n})$, where

    $$d_{u,v} = \underset{p}{\operatorname{argmin}} (A_k)^p (i_u^k, i_v^k).$$

Spectral Embedding of Graphs (Luo, Wilson and Hancock, Pattern Recognition 2003)

The proposed feature vectors are further reduced by PCA, ICA, MDS in order to perform graph visualization and clustering:

Fig. 8. MDS space clustering using the eigenvalues.

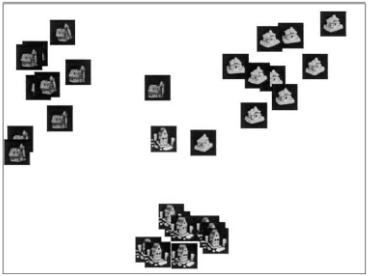


Fig. 10. ICA space clustering using the inter-mode adjacency matrix.

Spectral methods - Other approaches

Pattern Vectors from Algebraic Graph Theory (Wilson et al., TPAMI 2005)

Graph Characterization via Ihara Coefficients (Ren et al., Neural Networks 2011)

Spectral methods - Other approaches

Pattern Vectors from Algebraic Graph Theory (Wilson et al., TPAMI 2005)

- Spectral study of the Laplacian matrix
- Node permutation invariant features
- Sample elementary symmetric polynomials on the eigen-modes

Graph Characterization via Ihara Coefficients (Ren et al., Neural Networks 2011)

Pattern Vectors from Algebraic Graph Theory (Wilson et al., TPAMI 2005)

- Spectral study of the Laplacian matrix
- Node permutation invariant features
- Sample elementary symmetric polynomials on the eigen-modes

Graph Characterization via Ihara Coefficients (Ren et al., Neural Networks 2011)

- Extract Ihara Coefficients from the Oriented line graph
- Important topological information

Spectral methods - Summary

Good points

- Solid theoretical insight into the meaning of the extracted features
- Rich and discriminative features (really good clustering examples)

Drawbacks

- Spectral analysis is sensitive to structural errors
- Restriction on the nature of the graphs

Graph Classification based on Vector Space Embedding

(Riesen and Bunke, IJPRAI 2009)

Dissimilarity based embedding

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

The extracted features are based on distances to a set of prototype graphs:

Given a graph G and set of prototypes $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, the dissimilarity based embedding is defined by

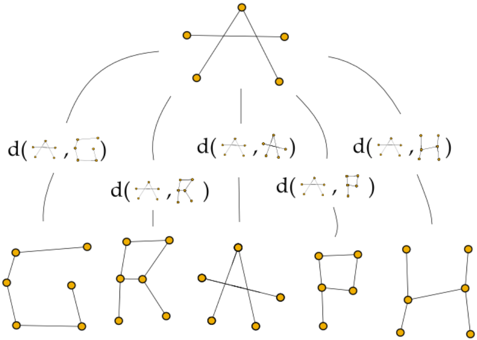
$$\varphi^{\mathcal{P}}(G) = (d(G, p_1), d(G, p_2), \ldots, d(G, p_n))$$

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

The extracted features are based on distances to a set of prototype graphs:

Given a graph G and set of prototypes $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, the dissimilarity based embedding is defined by

$$\varphi^{\mathcal{P}}(G) = (d(G, p_1), d(G, p_2), \ldots, d(G, p_n))$$



Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Issues to take care of:

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Issues to take care of:

- Distance between graphs

  $\longrightarrow$ Graph Edit Distance

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Issues to take care of:

- Distance between graphs

  $\longrightarrow$ Graph Edit Distance

- Selection of prototypes

  $\longrightarrow$ Random
  $\longrightarrow$ Spanning prototypes
  $\longrightarrow$ k-Centres
  $\longrightarrow$ And many others...

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Issues to take care of:

- Distance between graphs

  $\longrightarrow$ Graph Edit Distance

- Selection of prototypes

  $\longrightarrow$ Random
  $\longrightarrow$ Spanning prototypes
  $\longrightarrow$ k-Centres
  $\longrightarrow$ And many others...

- Number of prototypes

  $\longrightarrow$ Cross-validated

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Issues to take care of:

- Distance between graphs

  $\longrightarrow$ Graph Edit Distance

- Selection of prototypes

  $\longrightarrow$ Random
  $\longrightarrow$ Spanning prototypes
  $\longrightarrow$ *k*-Centres
  $\longrightarrow$ And many others...

- Number of prototypes

  $\longrightarrow$ Cross-validated

$\longrightarrow$ The work is concerned with the classification of graphs using the dissimilarity features.

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Good behaviour of the vectors in the embedding space:

$$\begin{aligned}
\| \varphi^P(G_1) - \varphi^P(G_2) \|^2 &= \langle \varphi^P(G_1), \varphi^P(G_1) \rangle + \langle \varphi^P(G_2), \varphi^P(G_2) \rangle - 2 \langle \varphi^P(G_1), \varphi^P(G_2) \rangle \\
&= \sum_{i=1}^{n} d(G_1, p_i)^2 + \sum_{i=1}^{n} d(G_2, p_i)^2 - 2 \sum_{i=1}^{n} d(G_1, p_i) d(G_2, p_i) \\
&= \sum_{i=1}^{n} (d(G_1, p_i) - d(G_2, p_i))^2 \\
&\leq n \cdot d(G_1, G_2)^2
\end{aligned}$$

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Good behaviour of the vectors in the embedding space:

$$\begin{aligned}
\| \varphi^P(G_1) - \varphi^P(G_2) \|^2 &= \langle \varphi^P(G_1), \varphi^P(G_1) \rangle + \langle \varphi^P(G_2), \varphi^P(G_2) \rangle - 2\langle \varphi^P(G_1), \varphi^P(G_2) \rangle \\
&= \sum_{i=1}^{n} d(G_1, p_i)^2 + \sum_{i=1}^{n} d(G_2, p_i)^2 - 2 \sum_{i=1}^{n} d(G_1, p_i) d(G_2, p_i) \\
&= \sum_{i=1}^{n} (d(G_1, p_i) - d(G_2, p_i))^2 \\
&\leq n \cdot d(G_1, G_2)^2
\end{aligned}$$

$\longrightarrow$ The Euclidean distance between feature vectors of graphs is equal to the sum of the squared differences between the edit distances of the graphs to the prototypes.

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Good behaviour of the vectors in the embedding space:

$$\begin{aligned}
\| \varphi^P(G_1) - \varphi^P(G_2) \|^2 &= \langle \varphi^P(G_1), \varphi^P(G_1) \rangle + \langle \varphi^P(G_2), \varphi^P(G_2) \rangle - 2\langle \varphi^P(G_1), \varphi^P(G_2) \rangle \\
&= \sum_{i=1}^{n} d(G_1, p_i)^2 + \sum_{i=1}^{n} d(G_2, p_i)^2 - 2\sum_{i=1}^{n} d(G_1, p_i) d(G_2, p_i) \\
&= \sum_{i=1}^{n} (d(G_1, p_i) - d(G_2, p_i))^2 \\
&\leq n \cdot d(G_1, G_2)^2
\end{aligned}$$

$\longrightarrow$ The Euclidean distance between feature vectors of graphs is equal to the sum of the squared differences between the edit distances of the graphs to the prototypes.

$\longrightarrow$ The Euclidean distance between feature vectors of graphs is upper-bounded by the edit distance of the graphs

Graph Classification based on Vector Space Embedding (Riesen and Bunke, IJPRAI 2009)

Good points

- Any kind of graphs can be plugged into this methodology (because of GED)
- Good behaviour of vectors in the embedding space, which leads to good classification rates

Drawbacks

- The distance measure (edit distance) is computationally challenging
- Validation of parameters has to be performed

Applications - Conclusions

- Assign a feature vector to every graph by
  - Spectral methods
  - Dissimilarity representation
  - And others...

Applications - Conclusions

- Assign a feature vector to every graph by
  - Spectral methods
  - Dissimilarity representation
  - And others...

- By providing a vector to every graph we are capable to visualize graphs and to apply statistical learning machines to graph-based input patterns (SVMs, Neural Networks,...)

Applications - Conclusions

- Assign a feature vector to every graph by
  - Spectral methods
  - Dissimilarity representation
  - And others...

- By providing a vector to every graph we are capable to visualize graphs and to apply statistical learning machines to graph-based input patterns (SVMs, Neural Networks,...)

- ICPR'2010 Graph Embedding Contest for a general framework

Applications - Conclusions

- Assign a feature vector to every graph by
  - Spectral methods
  - Dissimilarity representation
  - And others...
- By providing a vector to every graph we are capable to visualize graphs and to apply statistical learning machines to graph-based input patterns (SVMs, Neural Networks,...)
- ICPR'2010 Graph Embedding Contest for a general framework

$\longrightarrow$ We bridge the gap between the structural and the statistical pattern recognition fields.

Thanks for the attention!

Time for discussions?

In the next talk, we will present another graph embedding methodology.
Do not miss it!