Dimensionality Reduction for Graph of Words Embedding

GbR'2011 Oral talk

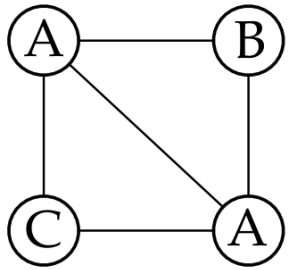
Jaume Gibert, Ernest Valveny
Computer Vision Center,
Universitat Autònoma de Barcelona,
Barcelona, Spain

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

Are these features really non-informative?

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,...
- ...



$\longrightarrow$ These features might not be informative nor discriminative.

Goals of the paper

- Define a new embedding methodology exploiting features based on statistics of node labels

Goals of the paper

- Define a new embedding methodology exploiting features based on statistics of node labels
- Adapt these features to graphs with continuous labels on nodes

→ Graph of Words Embedding

Goals of the paper

- Define a new embedding methodology exploiting features based on statistics of node labels
- Adapt these features to graphs with continuous labels on nodes

  → Graph of Words Embedding

- Deal with an inherent issue of the methodology: high dimensionality and sparsity

  → Dimensionality reduction

Motivation

- Given a graph $g = (V, E, \mu)$, a vectorial representation could be

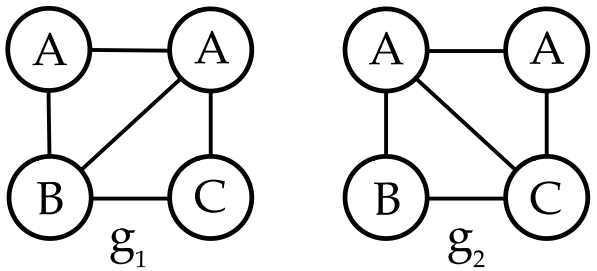
$$\mathbf{x}_g = (\#(l_1, g), \#(l_2, g), \ldots, \#(l_n, g)),$$

Motivation

- Given a graph $g = (V, E, \mu)$, a vectorial representation could be

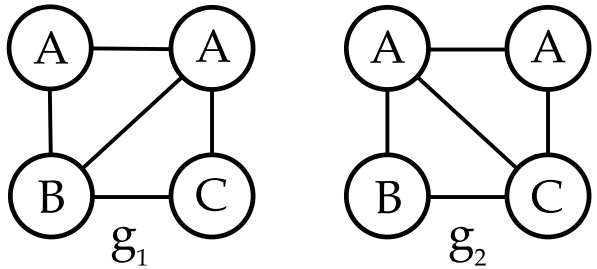
$$\mathbf{x}_g = (\#(l_1, g), \#(l_2, g), \ldots, \#(l_n, g)),$$

- For instance,



$$\begin{aligned}\mathbf{x}_{g_1} = \mathbf{x}_{g_2} &= (\#(A, g), \#(B, g), \#(C, g)) \\ &= (2, 1, 1)\end{aligned}$$

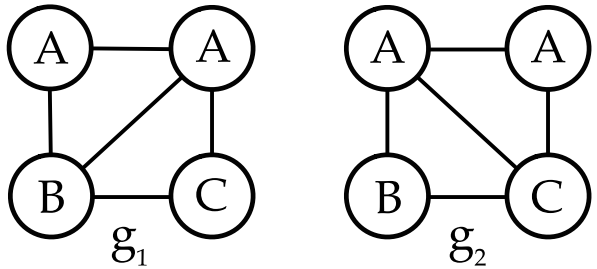
Motivation



- More features taking topology into account:

$$\#(l_i \leftrightarrow l_j, g),$$

Motivation



- More features taking topology into account:

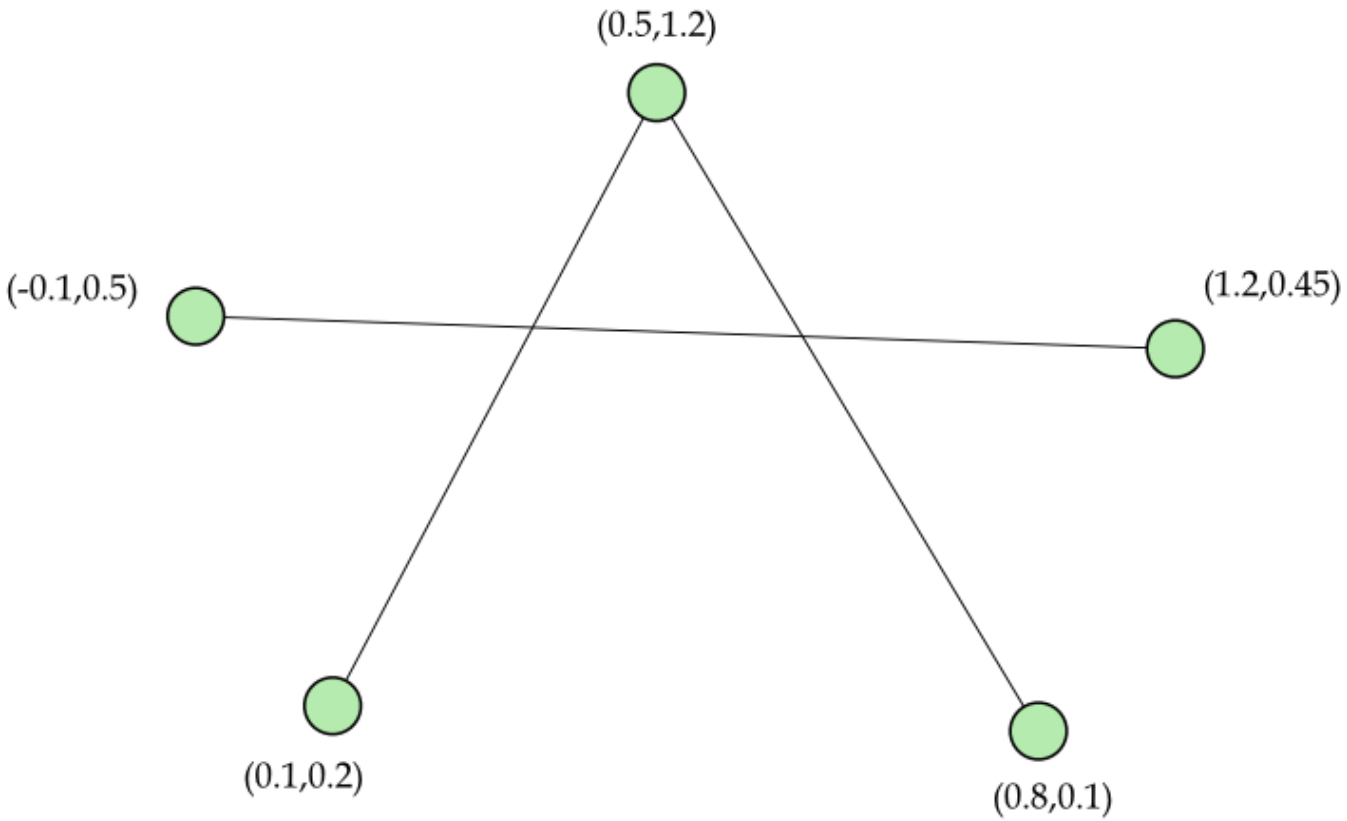
$$\#(l_i \leftrightarrow l_j, g),$$

$$\begin{aligned}\mathbf{x}_g = (&\#(A,g), \#(B,g), \#(C,g),\\ &\#(A \leftrightarrow A, g), \#(A \leftrightarrow B, g), \#(A \leftrightarrow C, g),\\ &\#(B \leftrightarrow B, g), \#(B \leftrightarrow C, g), \#(C \leftrightarrow C, g)),\end{aligned}$$

$$\begin{aligned}\mathbf{x}_{g_1} &= (\overbrace{2,1,1}^{\text{nodes}}, \overbrace{1,2,1,0,1,0}^{\text{edges}}),\\ \mathbf{x}_{g_2} &= (2,1,1,1,1,2,0,1,0).\end{aligned}$$

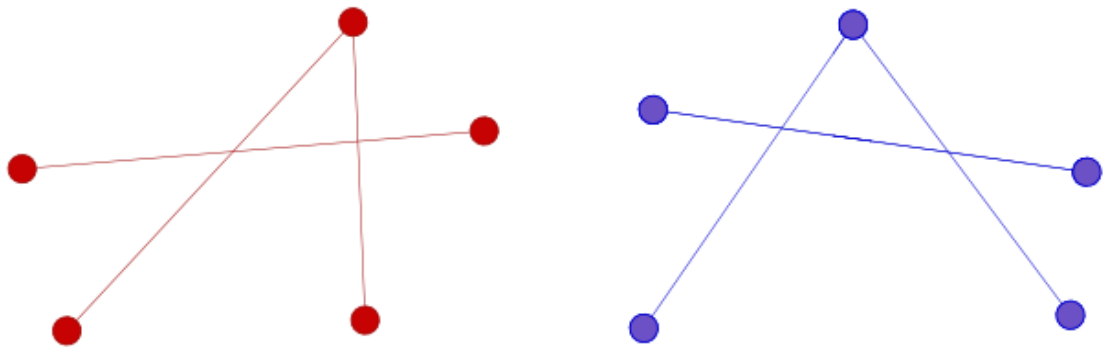
Problem

What if we have graphs with continuous labels?



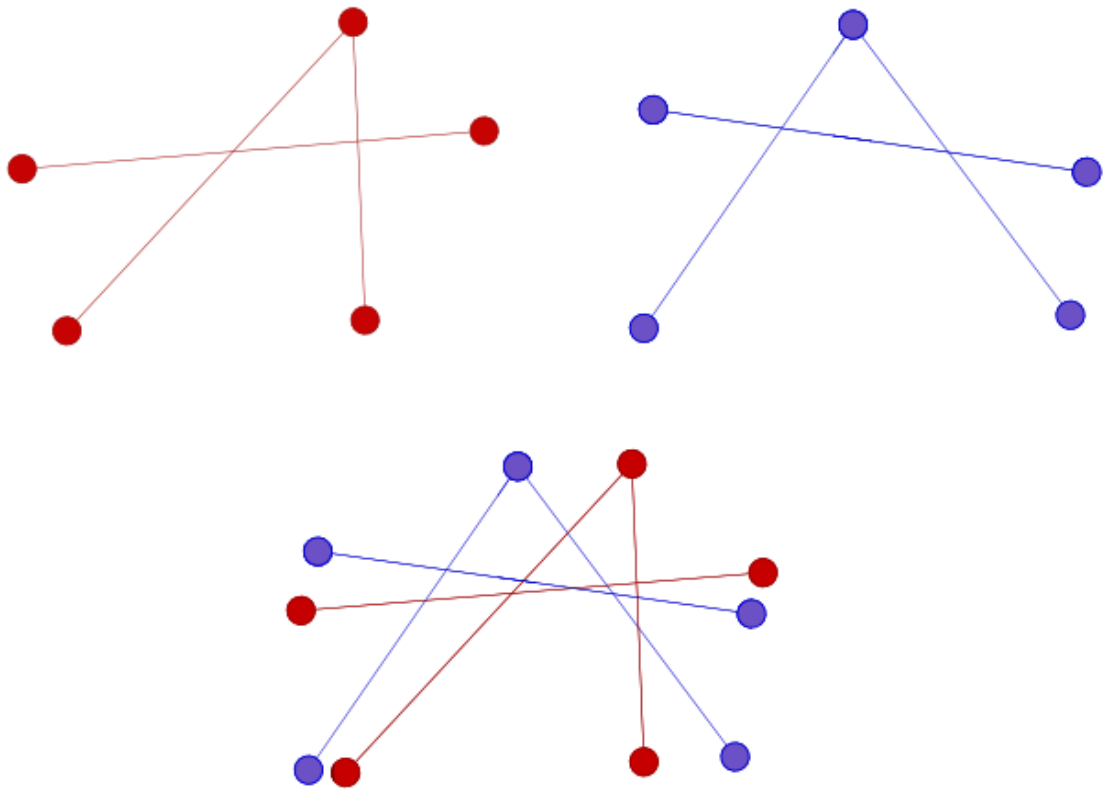
Idea

→ The more similar two labels are, the more they should count for the same label:



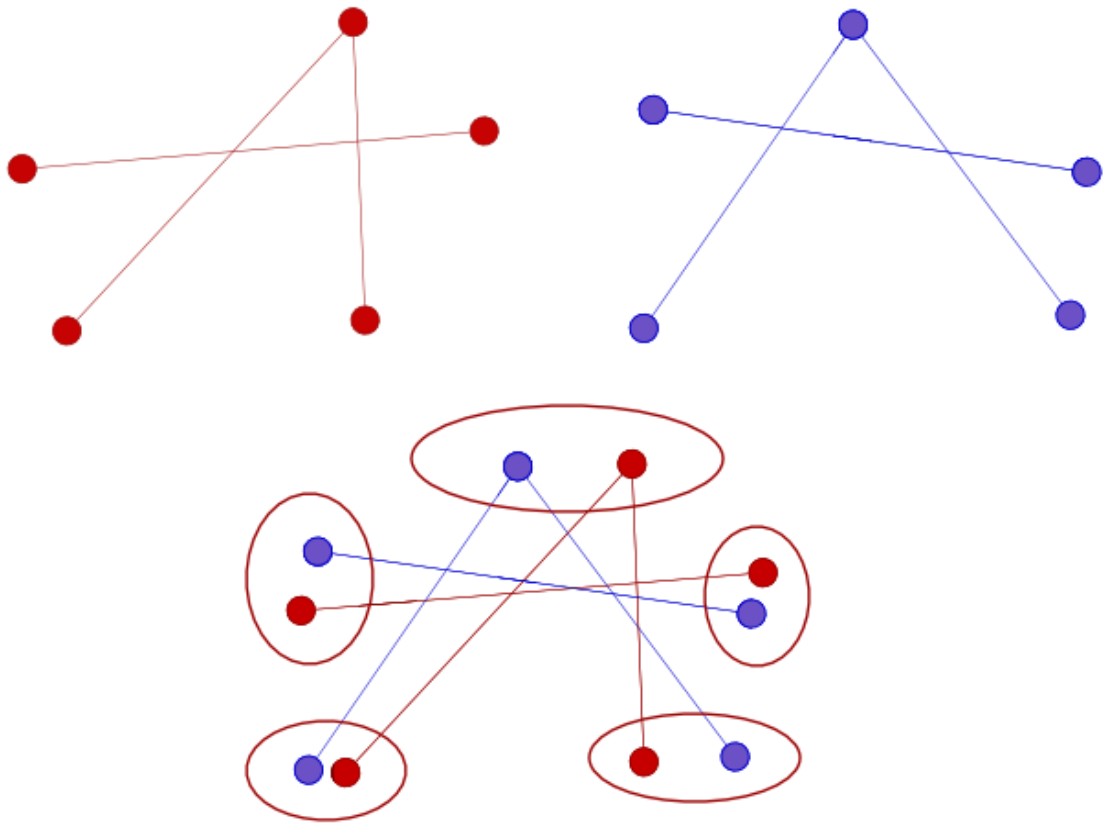
Idea

→ The more similar two labels are, the more they should count for the same label:



Idea

→ The more similar two labels are, the more they should count for the same label:



How to

We need a set of special points:

How to

We need a set of special points:

- From the whole set of node labels, select a set $\mathcal{W}$ of representatives / words[1]

[1] Graph of Words, in analogy to Bag-of-Words

How to

We need a set of special points:

- From the whole set of node labels, select a set $\mathcal{W}$ of representatives / words[1]
- Assign each node to the closest word

$$\lambda_h : V \longrightarrow \mathcal{W}$$
$$v \longmapsto \lambda_h(v) = \underset{w_i \in \mathcal{W}}{\operatorname{argmin}} \parallel \mu(v) - w_i \parallel_2$$

[1] Graph of Words, in analogy to Bag-of-Words

How to

We need a set of special points:

- From the whole set of node labels, select a set $\mathcal{W}$ of representatives / words[1]
- Assign each node to the closest word

$$\lambda_h : V \longrightarrow \mathcal{W}$$
$$v \longmapsto \lambda_h(v) = \underset{w_i \in \mathcal{W}}{\operatorname{argmin}} \parallel \mu(v) - w_i \parallel_2$$

- Each edge counts as a relation between the words assigned to its nodes

[1] Graph of Words, in analogy to Bag-of-Words

How to

We need a set of special points:

- From the whole set of node labels, select a set $\mathcal{W}$ of representatives / words[1]
- Assign each node to the closest word

$$\begin{aligned} \lambda_h : V &\longrightarrow \mathcal{W} \\ v &\longmapsto \lambda_h(v) = \underset{w_i \in \mathcal{W}}{\operatorname{argmin}} \parallel \mu(v) - w_i \parallel_2 \end{aligned}$$

- Each edge counts as a relation between the words assigned to its nodes
- Thus we have the features

$$U_i = \#(w_i, g) = |\{v \in V \,|\, w_i = \lambda_h(v)\}|$$

and

$$\begin{aligned} B_{ij} &= \#(w_i \leftrightarrow w_j, g) \\ &= |\{(u, v) \in E \,|\, w_i = \lambda_h(u) \wedge w_j = \lambda_h(v)\}|. \end{aligned}$$

[1] Graph of Words, in analogy to Bag-of-Words

Consequences

We have some issues to take care of:

Consequences

We have some issues to take care of:

- Some representatives might not be informative

Consequences

We have some issues to take care of:

- Some representatives might not be informative
- Some relations between words might not even appear in the whole set (sparsity)

Consequences

We have some issues to take care of:

- Some representatives might not be informative
- Some relations between words might not even appear in the whole set (sparsity)
- We may have redundancy in the features

Consequences

We have some issues to take care of:

- Some representatives might not be informative
- Some relations between words might not even appear in the whole set (sparsity)
- We may have redundancy in the features
- The size of the feature vector is quadratic in the number of words

Consequences

We have some issues to take care of:

- Some representatives might not be informative
- Some relations between words might not even appear in the whole set (sparsity)
- We may have redundancy in the features
- The size of the feature vector is quadratic in the number of words

$\longrightarrow$ Dimensionality Reduction

DR methods

In order to solve the dimensionality problem we have applied two different dimensionality reduction techniques

- Kernel Principal Component Analysis (kPCA)
- Independent Component Analysis(ICA)

DR methods

In order to solve the dimensionality problem we have applied two different dimensionality reduction techniques

- Kernel Principal Component Analysis (kPCA)
  - Non-linear extension of PCA
  - Use the kernel matrix to discover linear behavior in the kernel space
  - Back-project to find non-linear components of data
- Independent Component Analysis(ICA)

DR methods

In order to solve the dimensionality problem we have applied two different dimensionality reduction techniques

- Kernel Principal Component Analysis (kPCA)
  - Non-linear extension of PCA
  - Use the kernel matrix to discover linear behavior in the kernel space
  - Back-project to find non-linear components of data
- Independent Component Analysis(ICA)
  - Finds linearly uncorrelated components
  - Statistically independent components
  - Maximize the non-Gaussianity of data

Aim of the experimentation

- Discover the effect of the reduction by classification rates on
  - The raw vectors
  - The kPCA reduced vectors
  - The ICA reduced vectors
- We use a kNN classifier and validate the meta-parameters using a validation set

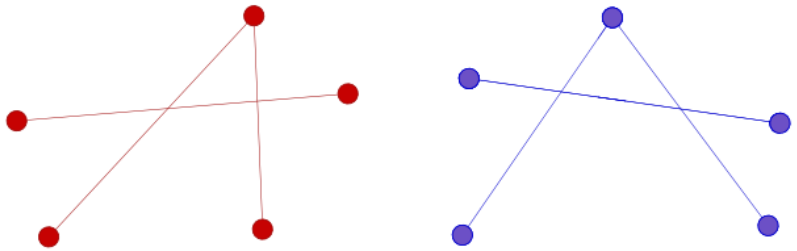
Databases

We use datasets from the IAM graphs database repository:

Databases

We use datasets from the IAM graphs database repository:

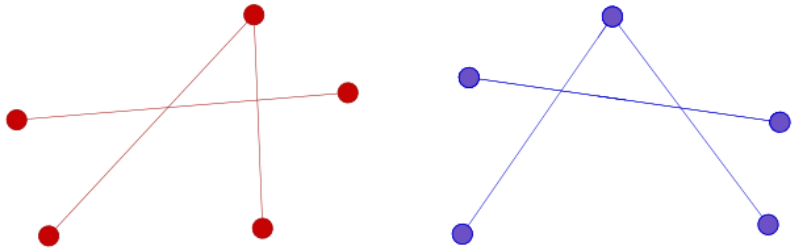
- Letters (low distortion)



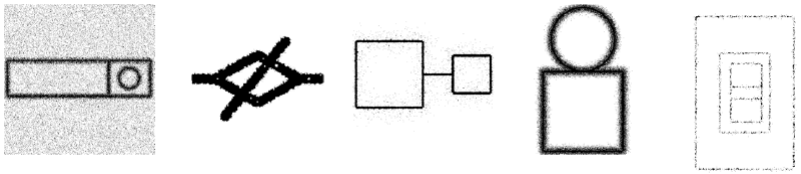
Databases

We use datasets from the IAM graphs database repository:

- Letters (low distortion)



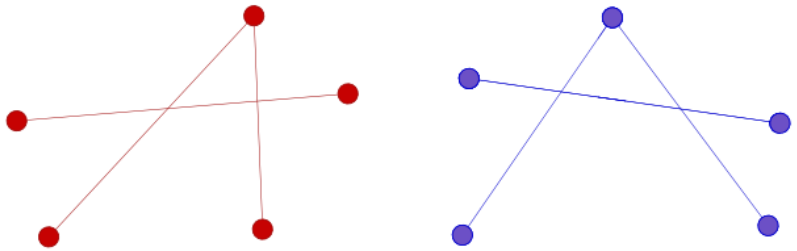
- GREC



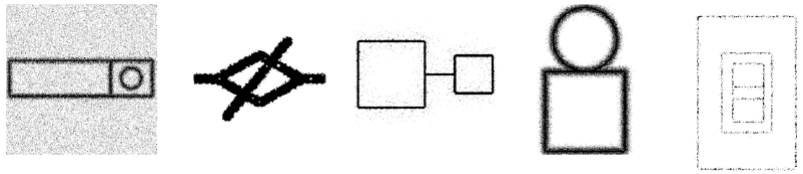
Databases

We use datasets from the IAM graphs database repository:

- Letters (low distortion)



- GREC



- Fingerprints

Results on validation

Selection of representatives: kMeans

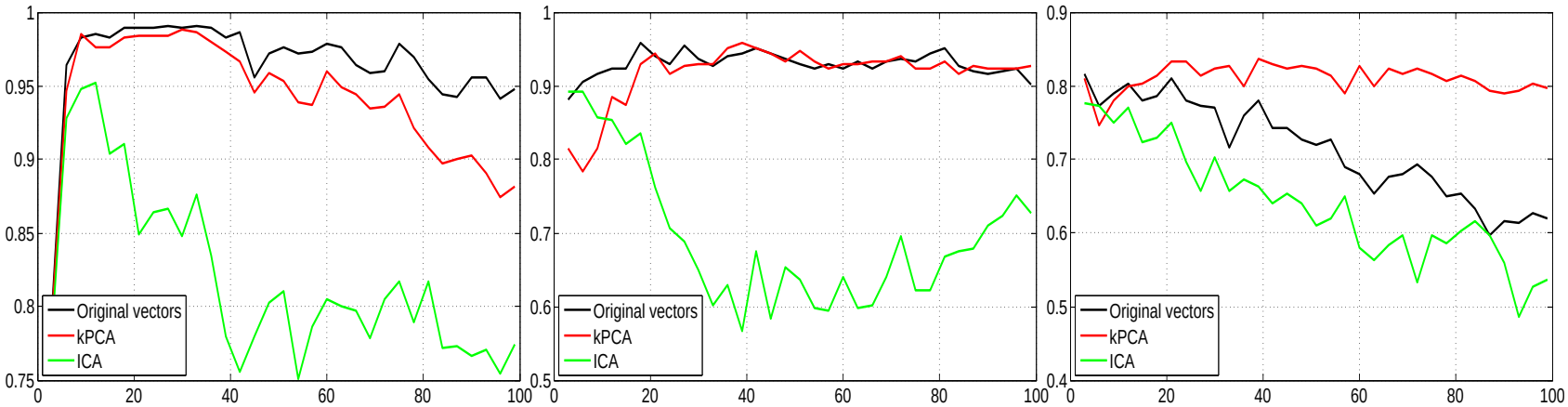


Figure: Accuracy rate on the validation set (vertical axis) for every vocabulary size (horizontal axis). These figures depict the comparison between the classification of the original vectors (without reduction) with the two dimensionality reduction techniques, kPCA and ICA.

Results on test

	Letter		GREC		Fingerprint	
	AR	Dims	AR	Dims	AR	Dims
Original vectors	**98.8**	702	**97.5**	3402	77.7	189
kPCA	97.6	43	97.1	67	**80.6**	108
ICA	82.8	251	58.9	218	63.3	184

- Results with kPCA are as good as with raw vectors while reducing dimensionality
- ICA does not provide good results due to the correlation between features

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives
- Address the dimensionality problem it exhibits

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives
- Address the dimensionality problem it exhibits
- Good results when using kPCA

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives
- Address the dimensionality problem it exhibits
- Good results when using kPCA

- More kernel functions on the kPCA framework to improve results

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives
- Address the dimensionality problem it exhibits
- Good results when using kPCA

- More kernel functions on the kPCA framework to improve results
- Other selection methods for the set of representatives

Conclusions and future work

- We have presented an embedding methodology for continuously labelled graphs
- It exploits statistics of node labels by similarity to a set of representatives
- Address the dimensionality problem it exhibits
- Good results when using kPCA

- More kernel functions on the kPCA framework to improve results
- Other selection methods for the set of representatives
- Combination of different sets of representatives

Thanks for the attention!

Questions?