A Non-Parametric Spectral Model for Graph Classification

Andrea Gasparetto, Giorgia Minello and Andrea Torsello
Dipartimento di Scienze Ambientali, Informatica e Statistica
Universitá Ca' Foscari Venezia
Via Torino 155, 30172 Mestre (VE), Italy
{*andrea.gasparetto, torsello*}*@unive.it, giorgia.min@gmail.com*

Keywords: Classification, Statistical Learning Framework, Structural Representation, Graph Model

Abstract: Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Despite this, the methodology available for learning structural representations from sets of training examples is relatively limited. In this paper we take a simple yet effective spectral approach to graph learning. In particular, we define a novel model of structural representation based on the spectral decomposition of graph Laplacian of a set of graphs, but which make away with the need of one-to-one node-correspondences at the base of several previous approaches, and handles directly a set of other invariants of the representation which are often neglected. An experimental evaluation shows that the approach significantly improves over the state of the art.

1 INTRODUCTION

Graph-based representations have been applied with considerable success to several tasks as convenient means of representing structural patterns. Examples include the arrangement of shape primitives or feature points in images, molecules, and social networks [Estrada and Jepson, 2009]. Their success lies in their ability to concisely capture the relational arrangement of primitives, in a manner which can be invariant to irrelevant transformation such as changes in object viewpoint. Despite their many advantages and attractive features, the methodology available for learning structural representations from sets of training examples is relatively limited, and the process of capturing the modes of structural variation for sets of graphs has proved to be elusive.

Structural representations are widely adopted in the context of Bayesian networks, or general relational models [Friedman and Koller, 2003], where structural learning processes are used to infer the stochastic dependency between these variables. However, these approaches rely on the availability of correspondence information for the nodes of the different structures used in learning. In many cases the identity of the nodes and their correspondences across samples of training data are not known, rather, the correspondences must be recovered from structure.

In the last few years, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs. In this context, spectral approaches have provided simple and effective procedures. For example, Luo and Hancock [Luo et al., 2006] use graph spectral features to embed graphs in a (low) fixed-dimensional space where standard vectorial analysis can be applied. While embedding approaches like this one preserve the structural information present, they do not provide a means of characterizing the modes of structural variation encountered and are limited by the stability of the graph's spectrum under structural perturbation. Bonev et al. [Bonev et al., 2007], and Bunke et al. [Bunke et al., 2003] summarize the data by creating super-graph representation from the available samples, while White and Wilson [White and Wilson, 2007] use a probabilistic model over the spectral decomposition of the graphs to produce a generative model of their structure. While these techniques provide a structural model of the samples, the way in which the super-graph is learned or estimated is largely heuristic in nature and is not rooted in a statistical learning framework. Torsello and Hancock [Torsello and Hancock, 2006] define a superstructure called tree-union that captures the relations and observation probabilities of all nodes of all the trees in the training set. The structure is obtained by merging the corresponding nodes and is critically dependent on the order in which trees are merged.

Todorovic and Ahuja [Todorovic and Ahuja, 2006] applied the approach to object recognition based on a hierarchical segmentation of image patches and lifted the order dependence by repeating the merger procedure several times and picking the best model according to an entropic measure. While these approaches do capture the structural variation present in the data, the model structure and model parameter are tightly coupled, which forces the learning process to be approximated through a series of merges, and all the observed nodes must be explicitly represented in the model, which then must specify in the same way proper structural variations and random noise.

In more recent work [Torsello, 2008, Torsello and Rossi, 2011] Torsello and co-workers proposed a generalization for graphs which allowed to decouple structure and model parameters and used a stochastic process to marginalize the set of correspondences. The process however still requires a (stochastic) one-to-one relationship between model and observed nodes and could only deal with size differences in the graphs by explicitly adding a isotropic noise model for the nodes.

In this paper we aim at defining a novel model of structural representation based on a spectral description of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. In particular, we follow White and Wilson [White and Wilson, 2007] in defining separate models for eigenvalues and eigenvectors, but cast the eigenvector model in terms of observation over an implicit density function over the spectral embedding space, and we learn the model through non-parametric density estimation. The eigenvalue model, on the other hand, is assumed to be log-normal, due to consideration similar to [Aubry et al., 2011].

2 SPECTRAL GENERATIVE MODEL

Let $G = (V,E)$ be a graph, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges, and let $A = (a_{ij})$ be its adjacency matrix. The degree d of a node is the number of edges incident to the node and it can be represented through the degree matrix $D = (d_{ij})$ which is a diagonal matrix with $d_{ii} = \sum_j a_{ij}$. Starting from these two matrix representations of a graph, it is possible to compute the *Laplacian* matrix, which is defined as the difference between the degree matrix D and the adjacency matrix A:

$$L = D - A$$

The Laplacian is a symmetric positive-definite matrix. Its lower eigenvalue is equal to 0 with multiplicity equal to the number of connected components in G. Further, the Laplacian is associated with random walks over the graph and it has been extensively used to provide spectral representations of structures [**?**]. The spectral representation of the graph can be obtained from the Laplacian through singular value decomposition. Given a Laplacian L, its decomposition is $L = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the matrix whose diagonal contains the ordered eigenvalues, while $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix whose columns are the ordered eigenvectors. This decomposition is unique up to a permutation of the nodes of the graph, a change of sign of the eigenvectors, or a change of basis over the eignespaces associated with a single eigenvalue, i.e., the following properties hold:

$$\begin{aligned} L &\simeq PLP^T = P\Phi\Lambda(P\Phi)^T && (1)\\ L &= \Phi\Lambda\Phi^T = \Phi S\Lambda S\Phi^T && (2)\\ L &= \Phi\Lambda\Phi^T = \Phi B_\lambda\Lambda B_\lambda\Phi^T && (3) \end{aligned}$$

where $\simeq$ indicates isomorphism of the underlying graphs, P is a permutation matrix, S is a diagonal matrix with diagonal entries equal to ± 1, and B_λ is a block-diagonal matrix with the block diagonal corresponding to the eigenvalues equal to λ in Λ and is orthogonal while all the remaining diagonal blocks are equal to the identity matrices.

Our goal is to devise a model for the graph spectra that can capture the main modes of variation present in a set of sample graphs, and that takes into account the invariances of the spectral representation. Following [White and Wilson, 2007] we make two separate and independent models for the eigenvalues and eigenvectors of the Laplacian:

$$P(G|\Theta) = P(\Lambda^G|\Theta^\Lambda)P(\Phi^G|\Theta^\Phi) \qquad (4)$$

where Θ is the graph-class model divided into its eigenvalue-model component Θ^Λ and eigenvector-model component Θ^Φ.

For the eigenvalue model we follow [Aubry et al., 2011] and opt to model the observation distribution of a single eigenvalue as a log-normal distribution. In [Aubry et al., 2011] it was shown that this model derived directly from rather straightforward stability considerations derived from matrix perturbation theory. As a result, we model the set of eigenvalues as a series of independent log-normal distribution, one per eigenvalue used, resulting in:

$$P(\Lambda^G|\Theta^\Lambda) = (2\pi)^{\frac{d}{2}} \prod_{i=1}^{d} \frac{1}{\lambda_i\sigma_i} \exp\left(\frac{-(\ln\lambda_i - \mu_i)^2}{2\sigma_i^2}\right) \qquad (5)$$

where λ_i and μ_i are model parameters to be learned from data and d is the number of eigenvalues/eigenvectors used in the model.

On the other hand, the eigenvector component is modelled as an unknown distribution $\mathcal{F}$ on the d-dimensional spectral embedding space $\Omega_d \subseteq \mathbb{R}^d$. The d-dimensional spectral embedding of a graph is obtained from the eigenvector matrix Φ^G by taking its first d columns, corresponding to the eigenvectors associated with the d smallest eigenvalues, excluding the trivial constant eigenvector corresponding to a 0 eigenvalue. With the reduced $n \times d$ eigenvector matrix $\hat{\Phi}$ at hand, we take its rows to be points in the d dimensional spectral embedding space Ω_d.

Note that there is a length invariance in the eigenvectors, which are usually assumed to be of unit Euclidean norm. This, however, results in a size compression of the spectral embedding points as the graph size grows. To correct this issue we scale the embedding vectors by multiplying them by the graph size n.

With this model we cast the learning phase into a non-parametric density estimates of the distribution of the spectral embedding points $\phi_1^G, \ldots, \phi_n^G$. Under these assumptions, the eigenvector model parameter Θ^Φ is constituted of a collection of N d-dimensional vectors $\theta_1^\Phi, \ldots, \theta_N^\Phi$ corresponding to samples from the unknown density function. In the learning phase these are obtained aligning and merging spectral embedding points from the sample graphs belonging to each class.

This per-vertex sample approach takes care of the permutational invariance, but we still need to explicitly deal with the other invariances, i.e., the sign of eigenvectors and choice of an eigenbasis. We solve those invariances by optimizing over the respective transformation groups. Furthermore, we lift the block constraint over the eigenbasis selection, relaxing it to an optimization over the orthogonal group $\mathbb{O}(d)$. This results in the following definition of the eigenvalue probability:

$$P(\Phi^G|\Theta^\Phi) = \max_{\mathcal{R}\in\mathbb{O}(d)} \max_{S\in\{\pm 1\}^d} \frac{1}{Nh^d} \prod_{i=1}^{d} \sum_{j=1}^{N} \exp\left(-\frac{\|\mathcal{R}S\phi_i^G - \theta_j^\Phi\|^2}{2h^{2d}}\right) \quad (6)$$

which is the product of Parzen-Rosenblatt kernel density estimators. ϕ_i^G is the vector obtained taking the first d elements of the i-th row of the eigenvector matrix Φ^G and θ_j^Φ is the j-th component of the eigenvector model Θ^Φ. Here we assume that the model is simply an array of samples from the graph class.

In this work we use Silverman's rule-of-thumb [Silverman, 1986] for the multivariate case to estimate the bandwidth parameter h.

$$h = \left(N\frac{d+2}{4}\right)^{-\frac{1}{d+4}} \sigma \quad (7)$$

where σ is computed as the squared root of the trace of the covariance matrix Σ of the eigenvector model divided by the number of nodes of the model

$$\sigma = \sqrt{\frac{1}{n} Tr(\Sigma)} \quad (8)$$

2.1 Model Learning

The learning process aims to estimate the parameters for the eigenvector and eigenvalue models. Given a set of graphs $G = \{G_1, G_2, \ldots, G_m\}$, belonging to the same class $\mathscr{C}$, we firstly compute their spectral decomposition, obtaining the set $\{(\Phi_1^\mathscr{C}, \Lambda_1^\mathscr{C}), (\Phi_2^\mathscr{C}, \Lambda_2^\mathscr{C}), \ldots, (\Phi_m^\mathscr{C}, \Lambda_m^\mathscr{C})\}$. In particular, the $\Phi_i^\mathscr{C}$s are composed by column vectors which are the first d non-trivial eigenvectors of the Laplacian matrix of the corresponding graph, while the $\Lambda_i^\mathscr{C}$s contain the first d non-zero eigenvalues. Hence, d represents our embedding dimension. The eigenvector model of the class $\mathscr{C}$, denoted as $\Phi^\mathscr{C}$, is defined as

$$\Phi^\mathscr{C} = \begin{bmatrix} \phi_1^1 & \phi_2^1 & \ldots & \phi_d^1 \\ \phi_1^2 & \phi_2^2 & \ldots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^m & \phi_2^m & \ldots & \phi_d^m \end{bmatrix}$$

where ϕ_j^i denotes the j-th non-trivial eigenvector (still a column vector) of the i-th graph of the set G. In other word, we perform a vertical concatenation of all the eigenvectors matrices of the graphs that belong to class $\mathscr{C}$. Thus, the dimension of the eigenvector model of the class is $(\sum_{i=1}^m ||G_i||) \times d$.

2.1.1 Estimating the Eigenvector Sign-Flips

The eigenvector matrix produced by the eigendecomposition is unique up to a sign factor. Since our method characterize every node of a graph with a feature vector, a sign disambiguation is mandatory. There are several techniques that allow to detect and solve this ambiguity, like using the correlation between two functions (i.e. probability density functions). If the correlation grows after a flip, then the eigenvector sign should be flipped. Unfortunately, with increasing size, this method becomes computationally heavy.

For such reason, we have to employ an heuristic-based method in order to solve the sign-ambiguity problem. Since it is an heuristic approach, it does not guarantee the discovery of all the correct signs. Given two graphs G_A and G_B, which belong to the same class $\mathscr{C}$, let Φ_j^A and Φ_j^B be the j-th eigenvectors of the spectral representation of the graphs. We assume eigenvectors to be random variables having unknown probability density function. We assume that all the j-th eigenvectors of graphs in the same class share a very similar *pdf* among them, up to the sign. A flipped sign does not influence the shape of a pdf, but the peak of the function results shifted. Once a reference graph is selected (for example, A), the sign ambiguity is solved by checking the sign of the peaks of each eigenvector of the reference graph and the others. An eigenvector is flipped when the signs of the peaks are different.

$$\phi_j^B = \begin{cases} \phi_j^B(-1) & \text{if } x_j^{A*} < 0 \text{ and } x_j^{B*} \geq 0\,, \\ \phi_j^B(-1) & \text{if } x_j^{A*} \geq 0 \text{ and } x_j^{B*} < 0\,, \\ \phi_j^B & \text{otherwise.} \end{cases} \tag{9}$$

The pdfs of each eigenvectors are estimated using kernel density estimation. The density estimates are evaluated at 100 points covering the range of the eigenvectors. Those evaluations are then used to find the peaks x_j^{A*} and x_j^{B*} of the distributions.

Hence, to solve the sign-ambiguity issue, before the construction of $\Phi^{\mathscr{C}}$, we flip each graph according to a reference graph G_f (chosen randomly within G) using (9).

The next step involves the rotation of each eigenvectors matrix according to the same reference graph G_f.

2.1.2 Estimating the Eigenvector Orthogonal Transformation

The sign disambiguation process produces a rough rotation which helps to align the eigenvectors of a graph with respect to the eigenvectors of a reference graph. In order to minimize the variance between the eigenvector matrices of a reference graph (one for each class) and the eigenvector matrices of the other graphs, another rotation step is applied. In particular, we are looking for the rotation which minimize the distance between the nodes of two graphs. More formally, we want to maximize the following:

$$\arg\max_{\mathcal{R}\in\mathbb{O}(d)} \prod_i P(\mathcal{R}x) \tag{10}$$

where

$$P(x) \propto \sum_j e^{-\frac{1}{2}\frac{\|x-x_j\|^2}{h^2}} \tag{11}$$

The above formulation of the optimization problem is then applied to our definition of probability density applying the constraints to a Parzen-Rosenblatt kernel density estimator, obtaining

$$\arg\max_{\mathcal{R}} \prod_i \sum_j e^{-\frac{1}{2}\frac{\|\mathcal{R}x_i-y_j\|^2}{h^2}} \tag{12}$$

We subdivide our rotation matrix in two rotation matrices, namely $\mathcal{R}$ (the initial rotation) and S (an additive rotation). The log-likelihood obtained after the introduction of the new rotation matrix to equation 12 can be written as

$$\mathcal{L} = \sum_i \log\left(\sum_j e^{-\frac{1}{2}\frac{\|S\mathcal{R}x_i-y_j\|^2}{h^2}}\right) \tag{13}$$

Let α_{ij} be defined as

$$\alpha_{i,j} = e^{-\frac{1}{2}\frac{\|\mathcal{R}\phi_i\phi_j^{\mathscr{C}}\|^2}{h^2}} \tag{14}$$

In order to solve 10, we compute the gradient with respect to the additive rotation matrix S introduced in 13.

$$\frac{\partial\mathcal{L}}{\partial S_{hk}} = \sum_i \frac{\sum_j \alpha_{ij}\left(-\frac{1}{2}\frac{\frac{\partial}{\partial S_{hk}}\|S\mathcal{R}x_i-y_j\|^2}{h^2}\right)}{\sum_j \alpha_{ij}} \tag{15}$$

where

$$\frac{\partial}{\partial S_{hk}}\|S\mathcal{R}x_i - y_j\|^2 = -2(y_i)_h(\mathcal{R}x_i)_k \tag{16}$$

Since they are scalar

$$\partial_S = -2y_j(\mathcal{R}x_i)^T = -2y_j x_i^T \mathcal{R}^T \tag{17}$$

We can now rewrite 13 as

$$\frac{\partial\mathcal{L}}{\partial S} = \left(\sum_i \frac{\sum_j \alpha_{ij}\frac{1}{h^2}y_j x_i^T}{\sum_j \alpha_{ij}}\right)\mathcal{R}^T \tag{18}$$

For the sake of readability, let A be defined as

$$A = \sum_i \frac{\sum_j \alpha_{ij}\frac{1}{h^2}y_j x_i^T}{\sum_j \alpha_{ij}} \tag{19}$$

Since S is an orthogonal rotation matrix, it belongs to the Lie group $\mathbb{O}(d)$. The tangent space at the identity element of the Lie group is its Lie algebra, which is the skew-symmetric matrices space. The

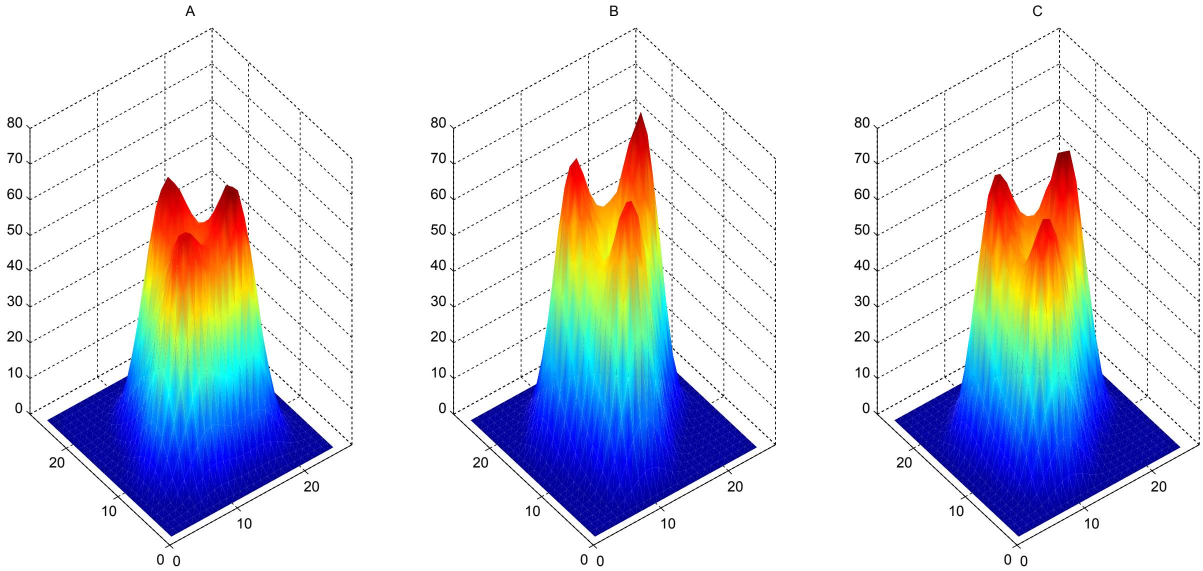

Figure 1: Example of the computation of the rotation matrix. A) KDE applied to the eigenvectors matrix of the Laplacian of a graph, B) KDE of a synthetically rotated eigenvectors matrix of the same graph, C) show the KDE of the eigenvectors matrix after the application of the rotation matrix computed using the described method.

skew-symmetric component of a matrix M is given by $\frac{M-M^T}{2}$.

In order to project the gradient to the null space (to find the maximum), we have to make $A\mathcal{R}^T$ symmetric. The rotation matrix $\mathcal{R}$ which symmetrize the previously computed gradient is obtained through the singular value decomposition (SVD) of A, $svd(A) = ULV^T$. In particular, we can compute R as

$$\mathcal{R} = UV^T \tag{20}$$

which symmetrize the gradient. Indeed

$$A\mathcal{R}^T = (ULV^T)(VU^T) = ULU^T \tag{21}$$

which is symmetric. Refer to figure 1 for a graphical example of the described process.

To compute the rotation we used the following algorithm:

1. The initial value of $\mathcal{R}$ is the identity matrix
2. Compute α_{ij} (14) for each $i = 1,\dots,n$ (where n is the number of nodes of a graph) and $j = 1,\dots,N$ (where N is the number of nodes of the model).
3. Compute the matrix A (19)
4. Compute the singular value decomposition of A, $svd(A) = ULV^T$
5. Compute $\mathcal{R}$ as $\mathcal{R} = UV^T$
6. If the convergence is achieved, i.e. $A = A^T$, or the maximum number of iterations allowed is reached, end the algorithm, otherwise repeat from 2

The maximum number of iterations parameter was set to 10 for the results showed in section 3.

2.1.3 Estimating the eigenvalue model

Let $G^{\mathscr{C}} = \{G_1, G_2, \dots, G_m\}$ be a set of graphs belonging to the same class $\mathscr{C}$, and let $\{\Phi_i^{\mathscr{C}}, \Lambda_i^{\mathscr{C}}\}$, $i = 1,\dots,m$, their spectral representation. The diagonal of the eigenvalue matrix $\Lambda_i^{\mathscr{C}}$ contains the eigenvalues $\{\lambda_1^i, \lambda_2^i, \dots, \lambda_d^i\}$ of the i-th graph of the set. Let

$$\Lambda^{\mathscr{C}} = \begin{bmatrix} diag(\lambda_1^{\mathscr{C}}) \\ diag(\lambda_2^{\mathscr{C}}) \\ \vdots \\ diag(\lambda_m^{\mathscr{C}}) \end{bmatrix}$$

be a $m \times d$ matrix containing the firsts d non-zero eigenvalues of the spectral representation. We assume that all the j-th eigenvalues of $\Lambda_i^{\mathscr{C}}$, with $j = 1,\dots,d$, are distributed as a log-normal distribution, as shown in 5. We do a maximum likelihood estimate for the model parameters resulting in:

$$\hat{\mu} = \frac{\sum_j \ln x_j}{d}, \quad \hat{\sigma}^2 = \frac{\sum_j (\ln x_j - \hat{\mu})^2}{d} \tag{22}$$

2.2 Prediction

Once the models are computed, we can combine them in order to classify a graph which does not belong to the training set used to compute $\{\Phi^{\mathscr{C}}, \Lambda^{\mathscr{C}}\}$. Let G^* be such graph. Let Φ^* and Λ^* be the spectral decomposition of the Laplacian of G^*. Thanks to the assumption

of independence between the two models, we can define the prediction as the posterior probability

$$P(\mathscr{C} \mid G^*) = P(\Phi^* \mid \Phi^{\mathscr{C}})P(\Lambda^* \mid \Lambda^{\mathscr{C}}) \quad (23)$$

Once both the above mentioned probabilities are computed, i.e. the probabilities with respect to the eigenvector model and to the eigenvalue model, and still assuming the independence between them, we can compute the conditional distribution with respect to the class $\mathscr{C}$ using equation 23. But since both $P(\Phi^* \mid \Phi^{\mathscr{C}})$ and $P(\Lambda^* \mid \Lambda^{\mathscr{C}})$ come from a log-derivation (equation 25 and 26), it can be rewritten as

$$logP(\mathscr{C} \mid G^*) = \ell_{\mathcal{L}}(\Phi^* \mid \Phi^{\mathscr{C}}) + \ell_{\mathcal{L}}(\Lambda^* \mid \Lambda^{\mathscr{C}}) \quad (24)$$

In particular, the eigenvector model log-likelihood is defined as

$$\ell_{\mathcal{L}}(\Phi^*|\Theta^{\Phi}) = \prod_{i=1}^{n} P(x_i) = \sum_{i=1}^{n} \log P(\bar{x}_i|\Theta^{\Phi}) \quad (25)$$

where n is the number of nodes of the graph G^*, while $\bar{x}_i$ is the row vector containing all the i-th coordinates of the eigenvector matrix.

The eigenvalue model log-likelihood is defined as

$$\ell_{\mathcal{L}}(\Lambda^*|\mu_i^{\Theta}, \sigma_i^{\Theta}) = \prod_{i=1}^{d} P(\lambda_i) = \sum_{i=1}^{d} \log P(\lambda_i) \quad (26)$$

with μ_i^{Θ} and σ_i^{Θ} which are the parameters estimated using 22.

Finally, a decision rule is applied in order to predict the membership of a graph to a certain class. In particular, for this work we classify the graphs assigning them to the most probable class (i.e. the class that yields the higher value).

3 EXPERIMENTAL RESULTS

We now evaluate the proposed model comparing it with a number of well-known alternative classification methods. More specifically, we compare our structure-based classifier with some popular graph kernels, like the unaligned QJSD kernel [Bai et al., 2013], the Weisfeiler-Lehman kernel [Shervashidze et al., 2011], the graphlet kernel [Shervashidze et al., 2009], the shortest-path kernel [Borgwardt and peter Kriegel, 2005], and the random walk kernel [Kashima et al., 2003]. Note that for the Weisfeiler-Lehman we set the number of iterations $h = 3$ and we attribute each node with its degree.

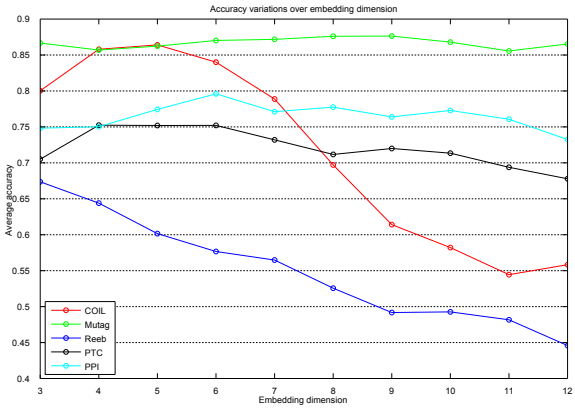

Figure 2: Average classification accuracy on all the datasets as we vary the embedding dimension for both the eigenvalues and eigenvectors matrices.

The experiments were run on the following datasets: the **PPI** dataset, which consists of protein-protein interaction (PPIs) networks related to histidine kinase [Jensen et al., 2008] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*). The **PTC** (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [Li et al., 2012] (here we use the 344 graphs in the MR class). 3) The **COIL** dataset, which consists of 5 objects from [Nene et al., 1996], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points. The **Reeb** dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [Biasotti et al., 2003]. Finally, the **Mutag** (Mutagenicity) dataset, which consists of graphs representing 188 chemical compounds, and aims to predict whether each compound possesses mutagenicity [Shervashidze et al., 2011]. Since the vertices and edges of each compound are labeled with a real number, we transform these graphs into unweighted graphs.

We use a binary C-SVM to test the efficacy of the kernels. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy $\pm$ standard error. We use a similar approach for the cross validation of our method. We perform a 10-fold cross validation over the datasets, using the proposed model. We tested our method using different numbers of eigenvectors and eigenvalues, which

Table 1: Classification accuracy ($\pm$ standard error) on unattributed graph datasets. OUR denotes the proposed model. SA QJSD and QJSU denote the Quantum Jensen-Shannon kernel in the aligned [Torsello et al., 2014] and unaligned [Bai et al., 2013] version, WL is the Weisfeiler-Lehman kernel [Shervashidze et al., 2011], GR denotes the graphlet kernel computed using all graphlets of size 3 [Shervashidze et al., 2009], SP is the shortest-path kernel [Borgwardt and peter Kriegel, 2005], and RW is the random walk kernel [Kashima et al., 2003]. For each classification method and dataset, the best performance is highlighted in bold.

Datasets	PPI	PTC	COIL5	Reeb	MUTAG
OUR	**79.60 ± 0.86**	**76.80 ± 1.52**	**86.41 ± 0.38**	**67.36 ± 1.52**	**87.74 ± 0.47**
QJSD	68.86 ± 1.00	55.78 ± 0.38	69.83 ± 0.22	35.03 ± 0.26	81.00 ± 0.51
SA QJSD	68.56 ± 0.87	57.07 ± 0.34	69.90 ± 0.22	35.78 ± 0.42	82.11 ± 0.30
WL	79.40 ± 0.83	56.86 ± 0.37	29.08 ± 0.57	50.73 ± 0.39	77.94 ± 0.46
GR	51.06 ± 1.00	55.70 ± 0.18	66.49 ± 0.25	22.90 ± 0.36	81.05 ± 0.41
SP	63.25 ± 0.97	56.32 ± 0.28	69.28 ± 0.42	55.85 ± 0.37	83.36 ± 0.52
RW	49.93 ± 0.83	55.78 ± 0.07	11.83 ± 0.17	15.98 ± 0.42	79.61 ± 0.64

can be seen as one of our free parameter. Furthermore, we tested the model with different levels of sub-sampling, that is, we sub-sampled all the graphs of the datasets (both training and test set) and apply our classification method to it.

Fig. 2 shows the average classification accuracy ($\pm$ standard error) on all the datasets as we vary the number of eigenvectors used. As you can see, every dataset behave differently based on the number of eigenvectors involved. In particular, for the COIL5 dataset, the use of more eigenvectors yields worst results, which means that the eigenvectors associated to the smaller non-zero eigenvalues of the spectra, models the classes better, while the subsequent ones just add noise to our representation. In the contrary, the Mutag dataset benefits from increasing the number of eigenvectors (and eigenvalues) involved in the creation of the class model.

Fig.3 shows the average classification accuracy ($\pm$ standard error) on all the datasets as we vary the percentage of sub-sampling applied to each graph of each dataset. In particular, the first accuracy measure corresponds to the application of our model on the spectral decomposition of the graphs where only 10% of the nodes were preserved. All the datasets (except for Mutag and PPI datasets) reach worse levels of accuracy with a lower number of nodes, meaning that the structural information given by each node of the model is useful for classification purpose. Conversely, the other datasets are more robust to sub-sampling.

Table 1 shows the average classification accuracy ($\pm$ standard error) of the different kernels and of our method on the selected datasets. The proposed model yields an increase of the performance with respect to the confronted graph kernels on all the used datasets. In particular, we obtained similar results with respect to the Weisfeiler-Lehman graph kernel on the PPI dataset. This is probably due to the use of the node labels in order to mitigate the localization problem and thus improving node localization in the evaluation process. Even though our model does not exploit node attributes, we were able to outperform all the kernels on all the other datasets.

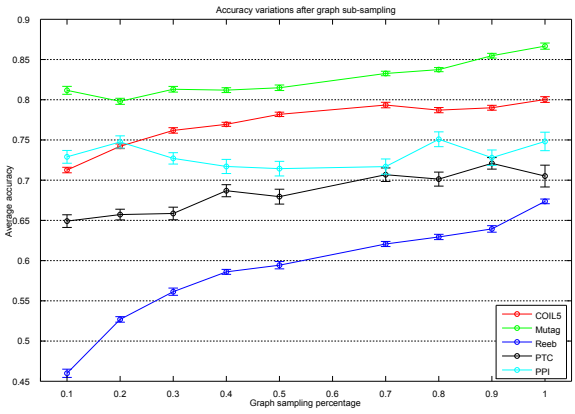

Figure 3: Average classification accuracy (with the interval segment representing the $\pm$ standard error) on all the datasets as we vary the percentage of sub-sampling applied to each graph of each dataset.

4 CONCLUSIONS

In this paper we have introduced a novel model of structural representation based on a spectral description of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. We showed how the defined separate models for eigenvalues and eigenvectors could be used within a statistical framework to address the graphs classification task. We tested the defined method against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

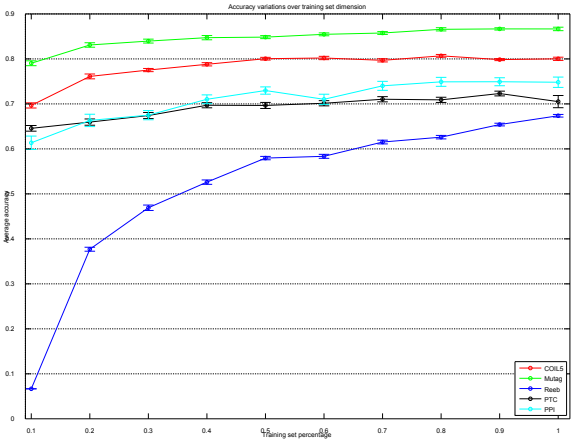

Figure 4: Average classification accuracy (with the interval segment representing the $\pm$ standard error) on all the datasets as we vary the percentage of graph of the training set used to build the model.

REFERENCES

Aubry, M., Schlickewei, U., and Cremers, D. (2011). The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633.

Bai, L., Hancock, E., Torsello, A., and Rossi, L. (2013). A quantum jensen-shannon graph kernel using the continuous-time quantum walk. In Kropatsch, W., Artner, N., Haxhimusa, Y., and Jiang, X., editors, *Graph-Based Representations in Pattern Recognition*, Lecture Notes in Computer Science, pages 121–131. Springer Berlin Heidelberg.

Biasotti, S., Marini, S., Mortara, M., Patan, G., Spagnuolo, M., and Falcidieno, B. (2003). 3d shape matching through topological structures. In Nystrm, I., Sanniti di Baja, G., and Svensson, S., editors, *Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203. Springer Berlin Heidelberg.

Bonev, B., Escolano, F., Lozano, M., Suau, P., Cazorla, M., and Aguilar, W. (2007). Constellations and the unsupervised learning of graphs. In Escolano, F. and Vento, M., editors, *Graph-Based Representations in Pattern Recognition*, volume 4538 of *Lecture Notes in Computer Science*, pages 340–350. Springer Berlin Heidelberg.

Borgwardt, K. M. and peter Kriegel, H. (2005). Shortest-path kernels on graphs. In *In Proceedings of the 2005 International Conference on Data Mining*, pages 74–81.

Bunke, H., Foggia, P., Guidobaldi, C., and Vento, M. (2003). Graph clustering using the weighted minimum common supergraph. In Hancock, E. and Vento, M., editors, *Graph Based Representations in Pattern Recognition*, volume 2726 of *Lecture Notes in Computer Science*, pages 235–246. Springer Berlin Heidelberg.

Estrada, F. and Jepson, A. (2009). Benchmarking image segmentation algorithms. *International journal of computer vision*, 85(2):167–181.

Friedman, N. and Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50(1-2):95–125.

Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Roth, E., Simonovic, M., Bork, P., and Mering, C. V. (2008). String 8 a global view on proteins and their functional interactions in 630 organisms.

Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328. AAAI Press.

Li, G., Semerci, M., Yener, B., and Zaki, M. J. (2012). Effective graph classification based on topological and label attributes. *Stat. Anal. Data Min.*, pages 265–283.

Luo, B., Wilson, R. C., and Hancock, E. R. (2006). A spectral approach to learning structural variations in graphs. *Pattern Recognition*, 39(6):1188 – 1198.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library (COIL-20). Technical report.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*

Shervashidze, N., Vishwanathan, S. V. N., Petri, T. H., Mehlhorn, K., and et al. (2009). Efficient graphlet kernels for large graph comparison.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Todorovic, S. and Ahuja, N. (2006). Extracting subimages of an unknown category from a set of images. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 927–934.

Torsello, A. (2008). An importance sampling approach to learning structural representations of shape. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

Torsello, A., Gasparetto, A., Rossi, L., and Hancock, E. (2014). *Transitive State Alignment for the Quantum Jensen-Shannon Kernel*.

Torsello, A. and Hancock, E. (2006). Learning shape-classes using a mixture of tree-unions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(6):954–967.

Torsello, A. and Rossi, L. (2011). Supervised learning of graph structure. In Pelillo, M. and Hancock, E. R., editors, *SIMBAD*, volume 7005 of *Lecture Notes in Computer Science*, pages 117–132. Springer.

White, D. and Wilson, R. (2007). Spectral generative models for graphs. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 35–42.