Pattern Recognition 48 (2015) 3357–3370

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Unfolding Kernel embeddings of graphs: Enhancing class separation through manifold learning

Luca Rossi [a,*], Andrea Torsello [b], Edwin R. Hancock [c]

[a] School of Computer Science, University of Birmingham, UK
[b] Department of Environmental Science, Informatics, and Statistics, Ca' Foscari University of Venice, Italy
[c] Department of Computer Science, The University of York, York YO10 5DD, UK

ARTICLE INFO

Article history:
Received 14 February 2014
Received in revised form
19 March 2015
Accepted 20 March 2015
Available online 31 March 2015

Keywords:
Kernel learning
Kernel unfolding
Graph kernels
Manifold learning

ABSTRACT

In this paper, we investigate the use of manifold learning techniques to enhance the separation properties of standard graph kernels. The idea stems from the observation that when we perform multidimensional scaling on the distance matrices extracted from the kernels, the resulting data tends to be clustered along a curve that wraps around the embedding space, a behavior that suggests that long range distances are not estimated accurately, resulting in an increased curvature of the embedding space. Hence, we propose to use a number of manifold learning techniques to compute a low-dimensional embedding of the graphs in an attempt to unfold the embedding manifold, and increase the class separation. We perform an extensive experimental evaluation on a number of standard graph datasets using the shortest-path (Borgwardt and Kriegel, 2005), graphlet (Shervashidze et al., 2009), random walk (Kashima et al., 2003) and Weisfeiler–Lehman (Shervashidze et al., 2011) kernels. We observe the most significant improvement in the case of the graphlet kernel, which fits with the observation that neglecting the locational information of the substructures leads to a stronger curvature of the embedding manifold. On the other hand, the Weisfeiler–Lehman kernel partially mitigates the locality problem by using the node labels information, and thus does not clearly benefit from the manifold learning. Interestingly, our experiments also show that the unfolding of the space seems to reduce the performance gap between the examined kernels.

1. Introduction

Graph-based representations have become increasingly popular due to their ability to characterize in a natural way a large number of systems which are best described in terms parts and binary relations. Concrete examples include the use of graphs to represent shapes [5], metabolic networks [6], protein structure [7], and road maps [8]. However, the rich expressiveness and versatility of graphs comes at the cost of added complexity and a reduced toolset of available pattern analysis algorithms. In fact, our ability to analyze data abstracted in terms of graphs is severely limited by the restrictions posed by standard pattern recognition techniques, which require data to be representable in a vectorial form. There are two reasons why graphs are not easily reduced to a vectorial form: first, there is no canonical ordering for the nodes in a graph, unlike the components of a vector. Hence, correspondences to a reference structure must be established as a prerequisite. Second, the variation in the graphs of a particular class may manifest itself as subtle changes in structure. Hence, even if the nodes or the edges of a graph could be encoded in a vectorial manner, the vectors would be of variable length, thus residing in different spaces.

The first 30 years of research in structural pattern recognition have been mostly concerned with the solution of the correspondence problem as the fundamental means of assessing structural similarity [9]. With the similarity at hand, similarity-based pattern recognition techniques such as the nearest neighbor rule can be used to perform recognition and classification tasks, or graphs may be embedded in a low-dimensional pattern space using either multidimensional scaling or alternative non-linear manifold leaning techniques.

Another alternative is to extract feature vectors from the graphs providing a pattern-space representation. There are a number of ways in which this can be done: one approach is to extract structural or topological features from the graphs under study. Graph spectral features extracted from the eigenvalues and eigenvectors of the adjacency or Laplacian matrices have been shown to be effective here [10,11]. Again, manifold learning techniques have been used in the literature to provide a way to unfold the pattern space and map the data onto low-dimensional spaces where the structural classes are well separated.

* Corresponding author.
E-mail addresses: l.rossi@cs.bham.ac.uk (L. Rossi), torsello@dsi.unive.it (A. Torsello), erh@cs.york.ac.uk (E.R. Hancock).

http://dx.doi.org/10.1016/j.patcog.2015.03.018
0031-3203/

The famous kernel trick [12] has shifted the problem from the vectorial representation of data, which now becomes implicit, to a similarity representation. This has allowed standard learning techniques to be applied to data for which no easy vectorial representation exists. More formally, once we define a positive semi-definite kernel $k : X \times X \to \mathbb{R}$ on a set X, there exists a map $\phi : X \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$, such that $k(x,y) = \phi(x)^\top \phi(y)$ for all $x, y \in X$. Also, given the kernel value between $\phi(x)$ and $\phi(y)$ one can easily compute the distance between them by noting that $\|\phi(x), \phi(y)\|^2 = \phi(x)^\top \phi(x) + \phi(y)^\top \phi(y) - 2\phi(x)^\top \phi(y)$. Thus, any algorithm that can be formulated in terms of dot products between the input vectors can be applied to the implicitly mapped data points through the direct substitution of the kernel for the dot product. For this reason, in recent years pattern recognition has witnessed an increasing interest in structural learning using graph kernels. However, due to the rich expressiveness of graphs, this task has also proven to be difficult, with the problem of defining *complete* kernels, i.e., ones where the implicit map ϕ is injective, sharing the same computational complexity of the graph isomorphism problem [13].

While the graph kernels proposed in the literature provide effective ways to generate implicit embeddings, there is no guarantee that the data in the Hilbert space will exhibit better class separation. This is of course a consequence of the complexity of the structural embedding problem and the limits for efficient kernel computations already analyzed by Gärtner et al. [13]. One evidence of this is the fact that the multidimensional scaling embeddings of several graph kernels show the so-called *horseshoe effect* [14] (see Fig. 1), i.e., the data tends to cluster tightly along a curve that wraps around the embedding space. This particular behavior is typically produced by a consistent underestimation of the real distances of the problem, i.e., the geodesic distances on the manifold, and it implies that the data gets placed onto a highly non-linear manifold embedded in the Hilbert space. The horseshoe is in fact the locus of intersection between the manifold and the plane used to visualize the data, and the high curvature is a result of the dimensionality compression on the data, which reduces the degrees of freedom of the points and forces them to cluster along the observed curve. We should also stress that this behavior can be a consequence of kernel normalization, a common procedure through which the data points are projected from the Hilbert space onto the unit sphere. This in turn creates an artificial curvature of the space that can create or exacerbate the observed horseshoe effect. Note, however, that while in general the non-linearity of the mapping is used to improve local class separability, a large global curvature can result in a folding of the manifold that can reduce long range separability.

For this reason, it is natural to investigate the impact of the locality of distance information on the performance of these kinds of kernels. To this end, given a set of graphs, we investigate the use of several manifold learning techniques to embed the graphs onto a low-dimensional vectorial space, in an attempt to unfold the embedding manifold, and increase class separation. More specifically, we investigate the use of four popular non-linear manifold learning techniques, namely Isomap [15], Laplacian Eigenmaps [16], Diffusion Maps [17] and Local Linear Embedding [18]. The selected techniques approach the manifold learning problem from radically different angles, with Isomap attempting to preserve the global distances and LLE trying to maintain the local neighborhood geometry.

Experiments on several standard datasets demonstrate that, as expected, the improvement is kernel and dataset dependent, with some kernels and datasets generally gaining very significant improvements in performance, while other not exhibiting significant variation or even modest reduction. Most importantly, the unfolding of the manifold invariably reduces the performance gap between the kernels examined. This suggests that the instances of kernels in the literature do not differ by any intrinsic difference in descriptive power, but in the level of warping of the embedding space. See Fig. 2 for an example where the non-linear mapping to a high-curvature manifold reduces the linear separability of data.

The remainder of this paper is organized as follows: Section 2 introduces some related work in graph kernels and manifold learning, while Section 3 illustrates the unwrapping idea and provide a more in-depth description of the kernels and manifold learning techniques used in this study. Section 4 illustrates the experimental results, while

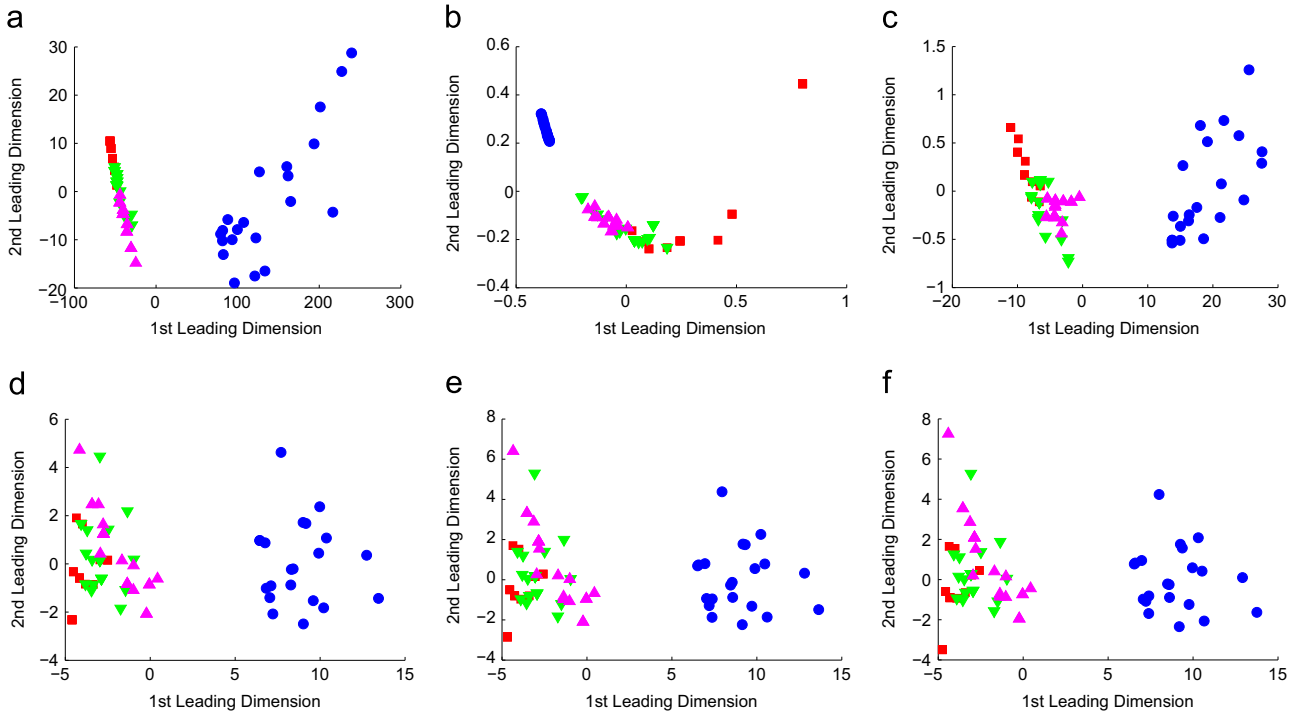

Fig. 1. The MDS embeddings of the shortest-path [1], graphlet [2], random walk [3] and Weisfeiler–Lehman [4] kernels (with $h = 1, 2, 3$) on the COIL dataset. (a) Shortest-path, (b) graphlet, (c) random walk, (d) WL_1, (e) WL_2, (f) WL_3.