Image and Vision Computing 29 (2011) 716–729

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

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

Inexact graph matching based on kernels for object retrieval in image databases[☆]

Justine Lebrun, Philippe-Henri Gosselin [*], Sylvie Philipp-Foliguet

ETIS, CNRS, ENSEA, Univ Cergy-Pontoise, F-95000 Cergy-Pontoise, France

ARTICLE INFO

Article history:
Received 3 August 2009
Received in revised form 21 July 2011
Accepted 29 July 2011

Keywords:
Online
Interactive
Database
Content-based
Object retrieval
Image retrieval
Machine learning
Kernel methods
Graph matching
Inexact match

ABSTRACT

In the framework of online object retrieval with learning, we address the problem of graph matching using kernel functions. An image is represented by a graph of regions where the edges represent the spatial relationships. Kernels on graphs are built from kernel on walks in the graph. This paper firstly proposes new kernels on graphs and on walks, which are very efficient for graphs of regions. Secondly we propose fast solutions for exact or approximate computation of these kernels. Thirdly we show results for the retrieval of images containing a specific object with the help of very few examples and counter-examples in the framework of an active retrieval scheme.

1. Introduction

One of the goals of the content-based image retrieval is to retrieve images containing a particular object or type of object, animal or person, whose shape can be very variable and set in a background also very variable. Global signatures are not a good way to solve this problem, especially if the context (background) brings no information. Approaches based on points of interest are interesting, but must be used with a high number of points to be efficient, and thus have a very high computational complexity. A promising approach is to represent an object by a set of regions characterized on one hand by intrinsic features (such as color, texture or shape), and on the other hand by spatial relations between them. The adjacency graph of regions constitutes a structure well adapted to represent objects in their infinite variability. However the segmentation into regions is very difficult, since there is no unique solution (it depends on the level of detail expected for this segmentation) and it is very sensitive to changes in the lighting, in the scale and in the aspect of the object. The number and the characteristics of the regions representing the same object are thus very variable from one image to the other. The problem of retrieving images including a type of object can thus be considered as a problem of inexact graph matching.

A retrieval system needs a similarity measure and a retrieval engine. The most popular – because the most efficient – way to perform classification or browsing in a database is the Support Vector Machines (SVM). SVM are state-of-the-art large margin classifiers which have demonstrated remarkable performances in image retrieval, when associated with adequate kernel functions.

The problem of graph comparison is a topic which has been widely studied in the literature for several decades [1]. One reason is that this problem occurs in many domains as various as computer or social networks, chemistry, or pattern recognition. Another reason is that graphs may be of very various kinds, in their size, their structure, in the type of information they represent and so on and thus they gave rise to many different methods to compare and classify them, all methods passing through graph matching.

A first way to classify methods of graph matching concerns the structure of the graphs: there are two main categories of methods, depending whether the structure of both graphs is the same or if it may differ. The first category addresses the graph isomorphisms for which both graphs have the same number of vertices and the same number of edges, each vertex of one graph being matched with one and only one vertex of the other graph and the same for the edges. There is a wide literature on the problem of finding the best isomorphism between graphs or sub-graphs [2]. This type of method is used for example in chemistry, or in computer-aided design, where vertices and edges are affected with symbolic labels, such as “carbon” or “hydrogen” for vertices, and “over” or “under” to characterize edges in pattern recognition. The constraint of having exactly the

[☆] This paper has been recommended for acceptance by Maja Pantic.
[*] Corresponding author. Tel.: +33 1 30 73 62 98; fax: +33 1 30 73 66 27.
E-mail address: gosselin@ensea.fr (P.-H. Gosselin).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.07.008

same structure is often too strong and is relaxed in the second category where one vertex can match zero, one or several vertices of the other graph. The problem of comparing graphs with unlabeled vertices and edges is a NP-hard problem. When the vertices and the edges are labeled with symbols, the same problem is much simpler, since the combinatorial is much smaller: a vertex with label *l* is only matched with a vertex with the same label *l*, but it is still a NP-hard problem [3]. In our problem of image retrieval from datasets, we deal with graphs whose both vertices and edges are assigned with vectors of values. And we need not only to compare graphs in terms of structure but we need a similarity also taking into account the similarity between vertices and between edges.

The problem we address in this paper has two main challenges: compare graphs of various structures and deal with vertices and edges attributed with numerical values. As vertices and edges include numerical information, these graphs are called Attributed Relational Graphs (ARG). In our case, vertices represent regions of the image and edges represent adjacencies between regions (neighbor). Edges are directed since they are described by information such as: how much one region is over another one. There are no multiple edges between two vertices (two regions), but loops are possible in order to allow multiple matches (one vertex to several).

Because of the computational cost, algorithms to compute distances between graphs are either complete (they give the optimal solution, possibly with an exponential complexity) or incomplete (the complexity is polynomial but the optimal solution is not guaranteed [4]). Concerning the former ones, most methods use search trees and filtering to prune these trees. A* or "branch and bound" algorithm is then used to solve the problem [5–7]. In [8] A* algorithm is used to perform ARG isomorphism for an image retrieval task. A way to find the isomorphism between graphs is to represent them in a canonical way and then to compare these representations. The algorithm developed by McKay [9] is "regarded by many authors as the fastest isomorphism algorithm available today" [1]. Another solution is graph editing [10], which consists in deforming one graph into the other one. The drawback of complete methods is that, because of the computational cost, they are limited to small graphs [5]. Another algorithm based on graph deformations is the Graph Transform Matching, which is applied to image registration [11]. In many cases, such as clustering or similar document retrieval, the exact distance between graphs is not crucial and an approximation is sufficient. In incomplete methods, combinatorial optimization algorithms are used, with quadratic optimization like Softassign [12], or with estimation of distribution [13], or with taboo search [14], etc. Recently Vishwanathan et al. [15] proposed a method to compute a graph kernel from kernels on walks, which improves the time complexity. It is particularly efficient for sparse graphs, but limited to graphs with unlabeled vertices.

Recent approaches of graph comparison consider graphs as sets of substructures such as chains, walks, trees and even graphlets (small subgraphs). As we are interested in matching only a part of the image (the object and not its background), this approach seems able to measure a similarity between sets of regions with their layout. We thus propose to build kernels on graphs from kernels on walks to compute the similarity between images. In the previous papers using random walks [16–19,15], authors only compare walks of equal length. But in our application, we need to compute a similarity between graphs of different orders, which means that one vertex can be matched with several vertices of the other graph, this will be achieved by allowing loops in the walks.

We will show that the search tree is a representation well suited to a recursive building of the walks in a graph and that the branch and bound algorithm allows a fast computation of the best match. Moreover with this algorithm and the similarities we propose, we are able to compute either the exact distance or an approximation.

The novelty of this paper is firstly to propose new kernels between graphs and between walks, which are more efficient and faster than existing ones (Section 2). Secondly we propose solutions to the inexact graph matching problem for attributed graphs of regions (Section 3). Thirdly we show results for the retrieval of images containing a specific object with the help of very few examples and counter-examples in the framework of an active retrieval scheme (Section 4).

2. Kernels on graphs

Kernel-based methods, such as Support Vector Machines (SVM), have shown their robustness for image retrieval and many other domains, thanks to convex minimization criterion. Kernel functions can be seen as similarity functions, which respect properties known as Mercer properties [20].

The idea of syntax-driven kernels [3] as opposed to model-driven kernels is to define a kernel on graph from kernel on parts of the graphs. Such a kernel was first defined by Haussler with the convolution kernel [21]. Then Kondor et al. [22] defined kernels over discrete structures which can be regarded as the discretization of Gaussian kernels. Since 2003 many different kernels have been defined that can be arranged according to the kind of structure they consider:

- random walks in [17–19,15]
- paths in [23]: a path is a walk which does not go twice through the same vertex
- trees in [24,25]
- graphlets in [26]: graphlets are subgraphs of small order, typically 3 to 5 vertices. In [26], they only capture the structure of the graph, they do not carry any information on vertices and edges.

Most of these kernels have been designed for chemical or bioinformatics applications, where vertices and edges carry very few information, usually only a label and sometimes a vector of small dimension (less than 4 attributes). Moreover these methods lead with graphs of small order, except [26] which deals with graphs of several dozens or even several hundreds of vertices (but unlabeled).

There are two main approaches to define kernels on graphs, depending on the way the embedding of the graph into a vector space is performed.

The first approach is explicit. This means that only a subset of features extracted from the graphs can be considered (edge number, walks, spectrum…). In order to choose such features, prototypes are built using techniques like K-Means, PCA [27], MIL-based techniques [28] or randomized forests [29]. From a set of prototypes or of frequent patterns, an explicit embedding consists for example [30] in computing the distance to each of the prototypes and then to use a classical vector-based kernel. In [31] the vertices are embedded into a vector space thanks to a membership function to the pattern, this membership function can either be binary or obtained by diffusion of the pattern through the edges of the graph.

This approach bounds the dimension of vectors in the induced space, since they need to be explicitly stored. Moreover it leads to a global parametrization (such as the number of prototypes) which needs to be tuned for each database or each query. Then, the comparison of two graphs always depends on this global parameter. A solution to this problem can be to perform an online computation of the prototypes during the retrieval [32,33].

The second approach performs an implicit embedding of graphs into a vector space, which means that the vectors in the space induced by the kernel function are never computed. This can be done using spectral techniques for matching pairs [34] or for high order matching, with tensors [35]. Other kernels were proposed, which behave like the similarity functions based on votes, but with respect to mathematical properties [36–38].

In this paper, we focus on this last approach and especially on kernels based on random walks, since they are in our opinion the most adapted to compare graphs whose information is carried by the vertices and the edges, rather than by the structure of the graph.