Distance-Based Generalisation Operators for Graphs [⋆]

V. Estruch C. Ferri J. Hernández-Orallo M.J. Ramírez-Quintana

DSIC, Univ. Politècnica de València , Camí de Vera s/n, 46020 València, Spain.
{vestruch,cferri,jorallo,mramirez}@dsic.upv.es

Abstract. Distances and similarity functions between structured data-types, such as graphs, have been widely employed in machine learning since they are able to identify similar cases or prototypes from which decisions can be made. In these distance-based methods the justification of the labelling of a new case a is usually based on expressions such as "label(a)=label(b) because case a is similar to case b". However, a more meaningful pattern, such as "because case a and b have properties x and y" is usually more difficult to find since the connection of this pattern with the distance-based method might be inconsistent [4]. In this paper we study possible consistent generalisation operators for the particular case of graphs embedded in metric spaces.

Keywords: distance-based methods, generalisation operators, graph-based representation, graph-edit distance, metric spaces.

1 Introduction

While in some learning problems the data can be described by a single-fixed row of nomimal or numerical attributes, in most others, such as biomedicine or web mining, a structured representation language is needed.

In general, for structured-data domains (e.g. graph-based instances), those properties inherent to the sort of data (e.g. common subgraphs, trees, cycles, etc.), might be important to achieve a competitive solution. This circumstance has motivated that some learning techniques which directly deal with structured data have been developed (multi-relational data mining [3]). Among them, distance-based methods are really popular for this purpose because several distance functions can be found for different sorts of data. However, unlike the ILP approaches, these methods do not give an explanation of their predictions. It is due to the fact that matches between two objects (e.g. two molecules) are encoded by a number (their distance). Unfortunately, model comprehensibility would be useful in some contexts. Imagine, for instance, in molecule classification, how interesting it would be to describe a cluster of molecules by saying

[⋆] This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN 2004-7943-C04-02, ICT for EU-India Cross-Cultural Dissemination Project under grant ALA/95/23/2003/077-054, Generalitat Valenciana under grant GV06/301 and UPV under grant TAMAT.

what chemical structures these molecules have in common instead of saying that they are closed according to a certain distance measure used in the clustering process.

Providing the possibility of this kind of descriptions for a distance-based algorithm would imply to incorporate a pattern language and generalisation operators [7]. But in this case, the model (expressed in a hopeful comprehensible pattern language) which generalises a set of elements could be inconsistent with the distance employed. The idea was initially considered in [4] by introducing the notion of distance-based binary generalisations. In [5], a more general framework in order to handle n-ary generalisation operators and to introduce the notion of minimal distance-based generalisations is presented.

In this paper, we use some of the concepts introduced in [5], to study generalisation operators and pattern languages for graph-based representations embedded in metric spaces. For this purpose, we consider two graph distance functions: the first one is described in [2, 8], and then we propose a second distance which is a bit more general. Next, a pattern language for graphs is introduced. This language is utilised to represent, in a comprehensible way, the generalisation computed by the generalisation operator. Next, for each metric space, a distance-based generalisation operator is defined according to our framework. Studying these generalisation operators will let us know how difficult extracting consistent patterns can be in each metric space.

2 Preliminaries

In this section we briefly review some basic concepts about graphs and graph distances. For any concept not explicitly included we refer the reader to [2].

A graph is a 4-tuple $G = (V, E, \mu, \nu)$ where V is a finite set of vertices, E is a set of edges (each one denoted as a pair of vertices belonging to $V \times V$), and μ and ν are functions which assign labels to vertices and edges respectively. The number of nodes of a graph $G = (V, E, \mu, \nu)$ is given by $|V|$ and it is denoted as $|V_G|$. The number of edges of a graph G is given by $|E|$ and is denoted as $|E_G|$. Given a graph $G = (V, R, \mu, \nu)$, a subgraph of G is a graph $S = (V_S, E_S, \mu_S, \nu_S)$ such that $V_S \subseteq V$, $E_S \subseteq E \cap (V_S \times V_S)$, and μ_S and ν_S are the restrictions of μ and ν to V_S and E_S respectively, that is $\mu_S(v) = \mu(v)$ (res. $\nu_S(e) = \nu(e)$) if $v \in V_S$ (res. $e \in E_S$) and undefined in otherwise.

A graph G_1 is isomorphic to a graph G_2 if there is an edge (and label) preserving bijection between all vertices in G_1 and G_2. Let G, G_1 and G_2 be graphs. G is a common subgraph of G_1 and G_2 if it is a subgraph of G_1 and G_2. A common subgraph G of G_1 and G_2 is maximal, denoted as $mcs(G_1, G_2)$, if there exists no other common subgraph G' such that G is a subgraph of G'. A subgraph G_1 of a graph $G = (V, E, \mu, \nu)$ is said to be induced by a set of vertices $W \subseteq V$ if for any pair of vertices w_1 and w_2 of W, (w_1, w_2) is an edge of G_1 if and only if $(w_1, w_2) \in E$, that is, G_1 is isomorphic to G. The concepts of common subgraph and maximal common subgraph are trivially extended to subgraphs induced by a set of vertices. We denote a maximal common subgraph of G_1 and G_2 induced by a set of vertices as $vimcs(G_1, G_2)$. That is, $vimcs$ looks for a common set of vertices W in G_1 and G_2 such that W induces the maximal common subgraph and returns this common graph. Figure 1 illustrates with an example the above concepts. Note that, in general, there can be more than one maximal common subgraph induced by a set of vertices.

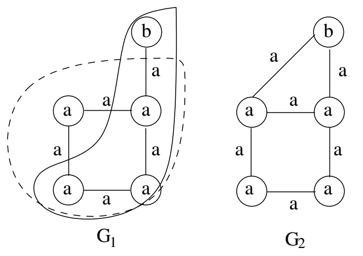

Fig. 1. G_1 is the maximal common subgraph of G_1 and G_2 whereas the subgraphs marked with dashed points and lines are the two maximal common subgraphs of G_1 and G_2 induced by a set of vertices.

We denote as d_1 and d_2 the distances we are working with. The first one is defined in terms of the minimal cost mapping transforming one graph into other. It is shown that d_1 can be expressed as $d_1(G_1, G_2) = |V_{G_1}| + |V_{G_2}| - 2|V_{vimcs(G_1,G_2)}|$ (see [2]). By modifying the definition of cost mapping employed in [2] we have derived a more general distance d_2 which can be calculated as $d_2(G_1, G_2) = |V_{G_1}| + |V_{G_2}| - 2|V_{mcs(G_1,G_2)}| + |E_{G_1}| + |E_{G_2}| - 2|E_{mcs(G_1,G_2)}|$.

Regarding the graphs G_1 and G_2 in Figure 1, we have $|V_{G_1}| = |V_{G_2}| = 5$, $|E_{G_1}| = 5$, $|E_{G_2}| = 6$, $V_{mcs(G_1,G_2)} = 5$, $E_{mcs(G_1,G_2)} = 5$ and $V_{vimcs(G_1,G_2)} = 4$. Hence, $d_1(G_1, G_2) = 2$ and $d_2(G_1, G_2) = 1$.

The computation of both distances in a feasible time might be a difficult task because the computation of mcs is needed and it is NP-complete. However, we are not interested in its calculus but we will use them to illustrate that extracting meaningful patterns in the metric space of graphs (G, d_2) is easier than in (G, d_1). The reason is that, as we will see, unlike mcs, the $vicms$ of two graphs is not unique, and we would have to take all of them into account in order to define a distance-based generalisation operator in (G, d_1).

3 Distance-based generalisation operators

In this section we present the main concepts related to our proposal of a generalisation framework based on distances. For more details see [5].

Our approach aims to define generalisation operators for data embedded in a metric space (X, d). These operators are denoted as $\Delta(E)$, where E is a finite set of elements ($|E| > 2$) of X to be generalised. In principle, a generalisation of E will be a particular set in 2^X containing E. But, for the sake of comprehensibility, the generalisation computed by $\Delta(E)$ will be expressed by a pattern p belonging to a pattern language $\mathcal{L}$. In fact, every pattern p represents a set of elements of

X and it is denoted by $Set(p)$. In this way, we can say that an element $x \in X$ is covered by a pattern p, if $x \in Set(p)$.

Additionally, if for every E, $\Delta(E)$ computes a generalisation of E "explaining" the distances among the elements of E, we will say that Δ is a distance-based generalisation operator. Then, the objective will be to find possible distance-based generalisation operators for the metric space (X, d).

For this purpose, we will focus on a particular kind of metric spaces, the connected ones. Informally speaking, this property means that given two elements in the metric space, we can always go from one to the other through elements belonging to the space such that these elements are at a minimal distance. In order to formally define what a connected space is, we need to introduce the notion of δ-path and the length of a δ-path.

Given a metric space (X, d), $I(X) = inf\{d(x, y) : \forall x, y \in X, x \neq y\}$ denotes the infimum distance of X. Let δ be a real positive number such that $\delta = I(X)$. Then, if $I(X) > 0$, a $\delta - path$ is a sequence of elements $P = \{x_i\}_{i=0}^{n>0}, \forall x_i \in X$, if $d(x_i, x_{i+1}) \leq \delta$ for all $0 \leq i \leq n-1$. Informally, if $(I(X) = 0)$, a $\delta - path$ is a continuous finite curve belonging to X. We also need the concept of the length of a δ-path P which is denoted by $L(P)$. If $P = \{x_i\}_{i=0}^{n>0}$, $L(P) = \sum_{i=0}^{n-1} d(x_i, x_{i+1})$. If P is a curve, $L(P)$ is just the length of the curve.

Let (X, d) be a metric space, two elements $x, y \in X$ are connected by a δ-path or equivalently, are δ-path connected, if there exists a δ-path $P = \{x_i\}_{i=0}^{n>0}$ such that $x_0 = x$ and $x_n = y$. Next, we will say that X is a connected metric space, if for every pair of elements x and y belonging to X, they are δ-path connected with $\delta \geq I(X)$. From this definition, we can say that a set $S \subseteq X$ is connected, if for every pair of elements in S they are δ-path connected, being $\delta = I(X)$. In what follows, we will use the term of x-connected space meaning that the metric space is connected and its $I(\cdot) = x$.

The notion of connected spaces and sets plays a key role in our approach since much too specific generalisations can be rejected (see Example 1).

Example 1. Let us suppose we are clustering graphs with the distance d_1 defined in Section 2. Imagine that each graph represents an organic compound and we would be interested in extracting some patterns saying which kind of molecules can be found in a cluster. One of the obtained clusters consists of two molecules m_1 and m_2 which are depicted in Figure 2.

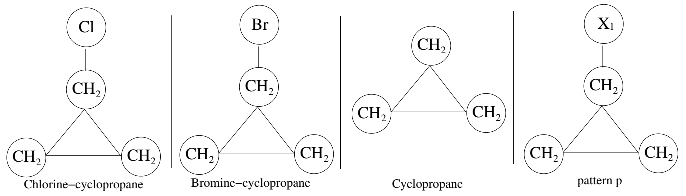

Fig. 2. The pattern p does not cover the cyclopropane molecule.

Let us obtain a pattern explaining the data distribution in this cluster. For this purpose, one could think of the pattern p (see Figure 2) saying "all the molecules with a cyclopropane structure and an extra atom". But this pattern might be much too specific. Considering that the molecules are really graphs in the space (G, d_1), we could think that the pattern p overfits the data since the cyclopropane molecule, which would be placed "between" [1] m_1 and m_2, that is, $d_1(m_1, \mathit{cyclopropane}) = d_1(\mathit{cyclopropane}, m_2) = 1$, is not covered by this pattern. Perhaps, a more natural pattern would be that one saying "all the molecules built from cyclopropane".

The last reasoning can be modelled in terms of connections. We know that (G, d_1) is a 1-connected space [6], and clearly, the set given by $\mathit{Set}(p)$ is not connected because the elements m_1 and m_2 are not connected by means of a 1-path included in $\mathit{Set}(p)$. That is, m_1 and m_2 are, at least, 2-path connected since $d_1(m_1, m_2) = 2$. However, $\mathit{Set}($"all the molecules built from cyclopropane"$)$ would be connected.

Finally, in order to define distance-based generalisation operators, the concept of "nerve" of a set of elements E is needed. A nerve of E, denoted by $N(E)$, is simply a connected[2] graph whose nodes are the elements belonging to E. Now,

Definition 1. ***(Distance-based generalisation operator)*** *Let (X, d) be a connected metric space and let $\mathcal{L}$ be a pattern language. Given a mapping $\Delta : E \rightarrow p \in \mathcal{L}$, we will say that Δ is a (proper or hard) distance-based generalisation operator if, for every $E \subseteq X$, $E \subset \mathit{Set}(p)$, $\mathit{Set}(p)$ is a connected set and there exists a nerve $N(E)$ such that,*

- ***(proper)*** *For every pair of elements x, y in E such that they are directly linked in $N(E)$, $\Delta(E)$ includes* ***some*** *$I(X)$-path P connecting x and y such that $d(x, y) = L(P)$.*
- ***(hard)*** *For every pair of elements x, y in E such that they are directly linked in $N(E)$, $\Delta(E)$ includes* ***all*** *$I(X)$-path P connecting x and y such that $d(x, y) = L(P)$.*

Definition 1 can be difficult to understand. Let us see an example with a binary sets of elements $E = \{x, y\}$. In this case a proper distance-based operator will be that one computing a generalisation of E which includes some of the paths built from the elements placed "between" x and y. The generalisation is hard, if it includes all the paths built from the elements placed "between" of x and y. In what follows, we will refer to them as proper or hard operators. On the other hand, independently to the operator, we can say that a pattern generalises E in a proper or a hard way if $\mathit{Set}(p)$ satisfies the conditions above (for further details see [5]).

The distinction between proper and hard is due to the fact that hard generalisations explains the distance among the elements better than the proper ones because it takes into account all the elements "between" two given elements and not only some of them as proper generalisations do. In fact, in some cases,

[1] Formally, given three elements x, y and z belonging to a metric space (X, d), we say that z is in "between" of x and y if $d(x, y) = d(x, z) + d(z, y)$.

[2] Here, the term connected refers to the well-known property for graphs.

proper generalisations do not always have an intuitive interpretation in terms of the distance involved [5].

4 Generalising set of graphs

In this section, we study some distance-based generalisation operators for graphs embedded in the metric spaces (G, d_1) and (G, d_2). First, for each metric space and using the pattern language $\mathcal{L}$ that will be defined below, we try to characterise the hard operators. From these hard operators, we will see how complicated the metric space is in order to compute distance-based patterns.

The pattern language $\mathcal{L}$ we are working with is composed of two types of patterns: the first-kind ($\mathcal{L}_1$) and the second-kind ($\mathcal{L}_2$) patterns. The so-called first-kind patterns will be a set of graphs built from an alphabet of labels containing constant and variable symbols. Regarding the second-kind patterns, these are expressed in terms of the first ones and they are specially introduced to improve the expressiveness of ($\mathcal{L}_1$).

Definition 2. ***(First-kind patterns*** $(\mathcal{L}_1)$***)*** *Given G the set of all the graphs over an alphabet of constant symbols A. If X is a set of variable symbols such that for all $X_i \in X$, X_i represents any constant symbol in A, then the language of first-kind patterns $(\mathcal{L}_1)$ is defined as the set of all the graphs over the alphabet $A \cup X$.*

Roughly speaking, the first-kind pattern is the intensional representation of a set of graphs sharing a particular topological structure, just as we show in the following example.

Example 2. Given the first-kind pattern language $(\mathcal{L}_1)$ defined from the set of constant symbols $A = \{a, b\}$ and the set of variable symbols $X = \{X_1, \ldots, X_n, \ldots\}$, consider the patterns p_1 in Figure 3.

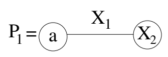

Fig. 3. An example of first-kind pattern.

This pattern represents only those graphs g in G made up of one edge and two vertices such that one of the vertices is labelled by the symbol a.

Although $\mathcal{L}_1$ permits quite interesting patterns, it still possesses some limitations. That is, imagine that we want to denote all the graphs in G having a subgraph in common. Despite the fact that this request seems usual, there is no pattern in $\mathcal{L}_1$ expressing it. For this reason, the class of the second-kind patterns is introduced.

Definition 3. ***(Second-kind patterns*** $\mathcal{L}_2$***)*** *Given the language of the the first-kind patterns $\mathcal{L}_1$, the language of the second-kind patterns $\mathcal{L}_2$ is defined as, $\mathcal{L}_2 = \{[p] : \forall p \in \mathcal{L}_1\} \cup \{\top\}$, where $[p]$ denotes all the graphs g in G having a subgraph covered by p and $\top$ denotes the whole space G.*

Example 3. The second-kind pattern p depicted in Figure 4 represents the set of all the graphs in G containing the path[3] $a - a - b$.

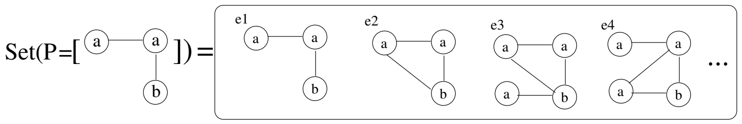

Fig. 4. An example of second-kind pattern using the notation $[\cdot]$.

Next, let us define distance-based generalisation operators for (G, d_1) and (G, d_2).

4.1 Generalisation operators for (G, d_1)

Before defining a generalisation operator, we have proved that the metric space (G, d_1) is a 1-connected metric space [6]). Now, let us characterise hard generalisation operators in (G, d_1) using $\mathcal{L}$. Note that, for every finite set of graphs $\{g_i\}_{i=1}^{n>2}$ in G, an operator $\Delta(\{g_i\}_{i=1}^{n>2})$ will return a pattern belonging either to $\mathcal{L}_1$ or to $\mathcal{L}_2$. It can be shown that first-kind patterns do not represent connected sets (see [6]) and therefore, according to Definition 1 they cannot represent a set computed by a generalisation operator. Then, the only possibility is that Δ computes a second-kind pattern. Thus, a hard operator is given by (see [6]):

$$\Delta(\{g_i\}_{i=1}^{n>2}) = \begin{cases} [p] & \text{if conditions (1) and (2) hold} \\ \top & \text{otherwise.} \end{cases}$$

where conditions (1) and (2) are:

(1) p is a subgraph of the $mcs(\{r_i\}_{i=1}^{n})$, where $mcs(\{r_i\}_{i=1}^{n}) \neq \emptyset$ and each r_i denotes one of the possible $vimcs(\{g_i\}_{i=1}^{n})$.

(2) there exists a nerve $N(\{g_i\}_{i=1}^{n})$ such that for every pair of graphs, g_i and g_j, directly linked in the nerve N, all the possible $vimcs(g_i, g_j)$ are included in $\{r_i\}_{i=1}^{n}$.

As we can appreciate, the above conditions are extremely restrictive. For instance, regarding the two graphs depicted in the Figure 1, the pattern $[a - b]$ generalises G_1 and G_2 since they have the edge $a - b$ in common. However, this pattern is not hard because the common squared subgraph is an element "between" G_1 and G_2 but it is not covered by the pattern. In fact, p does not satisfy condition (1) since $[a - b]$ is not a common subgraph of the $vicms(G_1, G_2)$. This fact gives us an idea about how difficult is computing hard operators in this space. Imaging an algorithm implementing Δ, this would have to check if a subgraph of a set of graphs G is in its turn a subgraph of the $vicms(G)$. According to this observation, the algorithms in the literature approaching the maximum common subgraph among a set of graphs [1] cannot be used as an implementation of Δ because it can not be ensured that the returned subgraph belongs to the intersection of the $vicms$.

[3] The concept of path referred here, is the well-known concept of path of a graph.

4.2 Generalisation operators for (G, d_2)

The metric space (G, d_2) is 1-connected as well [6]. One of the advantages of working with this metric space is that hard generalisation operators can be characterised in a more natural way using $\mathcal{L}$. Doing a similar analysis that in the previous subsection, hard operators can be defined in (G, d_2) as (see [6]):

$$\Delta(\{g_i\}_{i=1}^{n\geq 2}) = \begin{cases} [p] & \text{if } p \text{ is a subgraph of the } mcs(\{g_i\}_{i=1}^{n\geq 2}) \\ \top & \text{otherwise.} \end{cases}$$

Unlike the space (G, d_1), the hard operators can be defined easier and implemented by using one of the several algorithms in the literature for searching common subgraphs among a set of graphs since now Δ directly uses the mcs of the set of graphs instead of the $vicms$ of the set of graphs. Any subgraph returned by these algorithms can be perfectly used to define a hard generalisation.

5 Conclusions

Graph based learning is a challenging field due to the growing interest shown by several disciplines in mining data represented by means of graphs. However, most of the algorithms dealing with graphs, specially distance-based, do not return a model using graph features (e.x. common paths, walks, etc.) explaining why a sample has been labelled or grouped in one way. Although this kind of models is useful from a comprehensibility point of view, obtaining them could lead to inconsistency problems with the distance employed. In this work, we use some of the concepts of [5] to analyse which consistent models can be obtained when graphs are embedded in metric spaces.

References

1. E. Bengoetxea. Inexact graph matching using estimation of distribution algorithms, *PhD thesis*, 2003.
2. H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8): pp. 689–694, 1997.
3. S. Dzeroski and N. Lavrac, editors. *Relational Data Mining.* Springer-Verlag, 2001.
4. V. Estruch, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana. Distance based generalisation. In *Proc. of ILP*, volume 3625, pp. 87–102, 2005.
5. V. Estruch, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana. On the relationship between distance and generalisation. Tech. report, DSIC, UPV, http://www.dsic.upv.es/%7Eflip/#Papers, 2006.
6. V. Estruch, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana. Some results about generalisations of graphs embedded in metric spaces. Tech. report, DSIC, UPV, http://www.dsic.upv.es/%7Eflip/#Papers, 2006.
7. A. Hotho and G. Stume. Conceptual clustering of text clusters. In *Proc. of the WS on Frachgruppentreffen Maschinelles Lernen*, pp. 37–45, 2002.
8. A. Robles-Kelly and E.R. Hancock. Graph edit distance from spectral seriation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):365–378, 2005.