Analogical Structure Matching on Cartographic Data

Leo Mulhare, Diarmuid O'Donoghue, Adam C. Winstanley,

Department of Computer Science,

National University of Ireland, Maynooth,

Maynooth, Co. Kildare, Ireland.

{lmulhare, dod, adamw}@cs.may.ie

Abstract

We describe the application of analogical structure matching to the problem of classifying objects in structured cartographic data. The reasons for and the requirements of such a classification are firstly outlined. The attributes on which the structural matching will operate and the representation of this data in Prolog are then described. A brief mention is made of the extraction of these attributes from the sample data. Our domain-specific Cartographic Structure Matching Algorithm is then introduced and explained. The fusion of our algorithm's results with other classification techniques is mentioned, and some examples of the detection of misclassified polygons are provided. We finally provide a preliminary evaluation of our classification technique and suggest some future developments.

Introduction

Britain's national mapping agency, Ordnance Survey, is in the process of re-engineering its large-scale cartographic data in to a topologically structured format known as the Digital National Framework™ (DNF™) [Ordnance Survey]. At present, this high-resolution map data consists of spatially referenced point and line features, and text labelling. Points model real world features that cover a small area, such as post-boxes or telephone poles. Lines occasionally represent linear features such as fences, but they usually denote the boundaries between discrete areas, such as between a road and footpath. The conversion of these data sets to DNF format requires that the enclosed areas between lines be modelled as polygon features. Cartographic data containing explicit polygons enables the intelligent analysis of important features such as buildings, roads and fields. This richer quality data format is easier to update and greatly increases the usefulness of map data.

Each geographic feature within cartographic data must be classified as being a member of a particular class, known as a *feature code*. Examples of feature codes are “phone-box”, “wall of building” and “garden”. While the identification of polygons within line data can be automated with little difficulty, the classification of the resulting polygons is far from trivial. Some area features can be classified based on the feature codes of the lines that define them, but many require laborious manual classification. A sample DNF polygon data set, with unclassified polygons highlighted, can be seen in figure 1. As Ordnance Survey’s large-scale map database contains millions of polygons, feature-coding by hand is a very expensive and time consuming task. Automated classification techniques are clearly required. Work is ongoing within our department here at NUI Maynooth on the application of computer vision techniques to the polygon classification problem [Keyes, Winstanley]. A fusion of the results of three separate shape recognition techniques is currently being employed, and a high success rate in classification has been achieved. By integrating these results with those from other classification techniques, we hope to develop an even more robust polygon feature-coding tool.

Figure 1. *A Sample DNF data set. Unclassified polygons are shown in dark grey.*

The use of analogy has been shown to be central to any systems that can learn and solve novel problems. This paper describes our application of analogical reasoning to the polygon classification problem. Before we describe the cartographic domain and our classification algorithm, we briefly introduce the use of analogy for classification. Classifying GIS data closely mimics the geometric analogies made famous by Evans ANALOGY program [Evans]. The prime distinction is that attributes play a significant role in our system, whereas ANALOGY largely avoided such comparisons. Consider the following proportional analogy, of the form A is-to B as C is-to some unknown D. In figure 2, we can see that the Source domain consists of a "before and after" pair of diagrams. The transformation described by this pair of diagrams depicts the transformation of a plain T-shaped object, to a darkly shaded object of the same shape. The target domain consists of a single diagram that must undergo the same transformation [Bohan, O'Donoghue].

Figure 2. A Geometric Analogy Involving Attributes

In terms of cartographic information, we see the source domain as specifying the constraints under which the central (unclassified) object may be classified (or coloured). Only if all source domain objects are matched against the given target, and all the matching objects have the same colour (i.e. classification) can the central object be coloured (classified). In cartographic structure matching, we use a combination of the topology and classification of adjacent polygons as the attributes we base our mapping on. Our Cartographic Structure Matching Algorithm is similar to Keane & Brayshaw’s Incremental Analogy Machine [Keane], but we must deal with some commutative relationships and we require a retrieval phase to select a suitable template structure to support inference [O’Donoghue, Winstanley]. In the next section, we describe the relationships between a polygon and its neighbours. We also discuss the rules that govern our pattern-matching process.

Polygonal Context

When considering adjacency between polygons, there are two separate topological relationships to be considered. We define these as:

1. ***Line Adjacency***

   Two polygons are line-adjacent if they share a bordering line.

2. ***Point Adjacency***

   Two polygons are point-adjacent if they are not line-adjacent but they meet at 1 or more points.

Examples of the two types of adjacency can be seen in figure 3.

Figure 3. *The polygons on the left are line-adjacent, while the shaded polygons on the right are point-adjacent.*

Each polygon (and each point and line) in DNF standard data is given a unique identification number, called a *toid* (TOpograhical IDentifier). Individual polygons are identified by their toids during the matching process. In attempting to structurally match the contexts of two given polygons A and B, a polygon line-adjacent to A may only be mapped to one line-adjacent to B. The same restriction holds for point-adjacent neighbours. The matching of polygons is further restricted by their feature codes, so that a “building” polygon adjacent to A can only be matched to a polygon adjacent to B if the latter also has the “building” feature code. The context of a polygon X is a description of the toids and feature codes of the polygons adjacent to X and of the adjacencies between those polygons. We now describe our representation of this data.

Modelling Context

When designing a framework for analogising we must firstly decide what attributes of individual objects we wish to perform matching upon. Initially, we have chosen to model a polygon's context in terms of the adjacencies between it and its neighbours and the adjacencies between those neighbours. The context of a given polygon X is described with the following information:

- A list of polygons that are line-adjacent to X, specified by their toids and feature codes.
- A list of polygons that are point-adjacent to X, specified by their toids and feature codes.
- A list of pairs of neighbouring (that is, line-adjacent or point-adjacent to X) polygons that are line-adjacent to each other, specified by their toids.
- A list of pairs of neighbouring polygons that are point-adjacent to each other, specified by their toids.

Prolog, with its built-in depth-first search mechanism has been chosen to implement structural matching between polygons. All context information listed above is recorded within a single predicate. While this introduces some data redundancy, it has the advantage of reducing the amount of searching the Prolog interpreter has to do (In contrast, The Incremental Analogy Machine [Keane] represents an objects attributes as a hierarchically structured collection of predicates). We also include the current feature code of the polygon being described within our context predicate, as this attribute is transferred from the source to target domain when we perform an inference. In addition, the lengths of the four lists that hold the context information are recorded within the predicate. The structure matching algorithm uses these numbers to reduce its search space, as two polygons cannot be analogous if these lists are not the same length.

The predicate that records a given polygon X's context has the following structure:

```
context(toid of X,
        feature code of X,
        length of 1st list,
        length of 2nd list,
        length of 3rd list,
        length of 4th list,
        [toid & feature code pairs of line-adjacent
polygons],
        [toid & feature code pairs of point-adjacent
polygons],
        [pairs of toids of neighbours that are line-adjacent
to each other],
        [pairs of toids of neighbours that are point-adjacent
to each other]).
```

In the next section, we describe how these context predicates are derived from cartographic data.

Context Extraction from Cartographic Data

The context of individual polygons, as required for structural matching is derived from source data using the tool ArcView GIS. ArcView is a program used for the visualisation, editing and analysis of spatially referenced data, specifically cartographic data. ArcView's built-in scripting language *Avenue* is being used in the extraction of the required information. The sample data is in ESRI's *Shapefile* format, which represents polygons as a list of Cartesian co-ordinates. As this format does not provide any explicit links between neighbouring polygons, it is necessary to compare each polygon X with every other polygon in the data to identify those objects that are adjacent to X. This is accomplished using the spatial methods of Avenue's *Polygon* class.

Avenue's polygon intersection methods are used to identify the objects line-adjacent and point-adjacent to a particular polygon. The same technique is then used to identify the adjacency relationships between these neighbouring polygons. This information is recorded as a context predicate, as previously described, and saved to a Prolog source file. This file can then be loaded into a Prolog session and analysed by our structure matching algorithm. We now provide an example of how our structure matching algorithm can classify a polygon through inference.

Walkthrough of Structure Matching Process

In figure 4 a graphical representation of the full contexts of two polygons with toids 1 and 6 can be seen. All polygons are colour-coded with their current classification. Polygon 6 is currently unclassified. Polygon 1 is classified (as a building), and will be used as a template in an attempt to infer the feature code of polygon 6. The context of polygon 1 is our source domain, while the context of polygon 6 is our target domain.

Figure 4. *An example of a template polygon context and an unclassified polygon context.*

- We firstly attempt to uniquely map each polygon that is line-adjacent to 1 to a polygon with the same feature code that is line adjacent to 6. We succeed with the mapping [[2,10],[3,8]].
- Trying to generate a similar mapping for point-adjacent neighbours, we develop the mapping [[4,9],[5,7]], which we add to the mapping generated in the previous step to get [[2,10],[3,8],[4,9],[5,7]].
- We now try to map each pair of 1’s neighbours that are line-adjacent to each other to a pair of 6’s neighbours that are line-adjacent to each other, using the list of mappings we have developed. We succeed with [2,4] → [10,9], [2,5] → [10,7], [3,4] → [8,9], [3,5] → [8,7] and [4,5] → [9,7].
- Repeating the above step for point-adjacent neighbour pairs, we succeed with [2,3] → [10,8].

- We have now generated a mapping from the source domain to the target domain that preserves the structure of the source. We can infer that the feature code of the source, “building”, is also the feature code of the target.

The Cartographic Structure Matching Algorithm is now described in detail.

The Structure Matching Engine

Our analogy algorithm takes as arguments two polygon toids and searches for the context predicates that describe the topological localities of the polygons those toids reference. If it can find a full structural mapping between the two sets of neighbouring objects, the two polygons contexts are analogous, and knowledge can be transferred from one domain to the other. The algorithm fails if one of the polygons is found to have a neighbour of unknown class, as we do not allow inferences to be made on incomplete data.

The Cartographic Structure Matching Algorithm

For any two given polygons A and B whose contexts have been identified:

1. Check that the lengths of the four lists that describe A’s context correspond with the lengths of the lists that describe B’s context.
2. Create an exclusive mapping from a toid in A’s list of line-adjacent neighbours to a toid in B’s list of line-adjacent neighbours which has the same feature code. Add mapping to a list of mappings. Repeat until all toids in the two lists have been mapped.
3. Create an exclusive mapping from a toid in A’s list of point-adjacent neighbours to a toid in B’s list of point-adjacent neighbours which has the same feature code.

two lists have been mapped.

4. Using the list of mappings produced by stages 2 & 3, check that a pair of toids in A's list of line-adjacent neighbours maps to a pair of toids in B's list of line-adjacent neighbours. If not, backtrack and generate a different mapping list between A and B's neighbours. Repeat until all pairs of toids in the two lists of line-adjacent neighbours can be mapped.

5. Using the list of mappings produced by stages 2 & 3, check that a pair of toids in A's list of point-adjacent neighbours maps to a pair of toids in B's list of point-adjacent neighbours. If not, backtrack and generate a different mapping list between A and B's. Repeat until all pairs of toids in the two lists of point-adjacent neighbours can be mapped. At this point, a full structural mapping has been established between the contexts of A and B. This established analogy can now support inference from one domain to the other.

As mentioned earlier, the first stage of the process prevents unnecessary searching. It is included for the sake of efficiency and is not necessary for the functioning of the algorithm. Stage 2 tries to generate a mapping from A's line-adjacent neighbours to B's line-adjacent; stage 3 does likewise for point-adjacent neighbours. The penultimate stage checks if these mappings can translate A's list of pairs of line-adjacent neighbours to B's corresponding. Again, the final stage does likewise for the lists of pairs of point-adjacent neighbours.

The mappings that are created in stages 2 and 3 are represented in Prolog as a list of two toids. The first toid is that of a polygon adjacent to A and the second is that of a polygon adjacent to B. In searching for a structural match between two objects, it must be ensured that all mappings from one object's attributes to the other objects attributes are of a

1-to-1 nature. Therefore, when our algorithm creates a mapping between two toids, it removes both toids from the corresponding lists of neighbours.

In applying the established mappings in stages 4 and 5 the commutative nature of line-adjacency and point-adjacency must be considered, as the ordering of pairs of adjacent neighbours is arbitrary. In checking if [1,2] maps to [3,4], we must check if either 1 maps to 3 and 2 maps to 4 or if 1 maps to 4 and 2 maps to 3. It should be noted that there may be more than one full structural mapping between two particular contexts. In the next section, we show some misclassification detection results that have been achieved.

Experimental Results

In the following 2 cases classification errors have been located in a particular data set through the inspection of polygons that are found to have unusual contexts.

Figure 5.1, 5.2. *Sample DNF data sets. Buildings are highlighted on the left, roads on the right.*

Common sense tells us that one building cannot be contained within another building without any space in between. A search for such a context has revealed such a misclassification, as seen in figure 5.1. In this case, it appears that the largest highlighted polygon in the view has been erroneously classified as a building. Similarly, we can say that each section of road must be connected to another stretch of road, hence a road’s usefulness. Any road polygon that is not line-adjacent to another road polygon must be either misclassified itself or be line-adjacent to one or more road polygons that are in need of reclassification. In figure 5.2 we see such an unconnected road polygon that has been identified through analogy.

Preliminary Evaluation

The Cartographic Structure Matching Algorithm has been used to generate a set of template polygon contexts, which can suggest the most likely classifications for polygons that structurally match them. A large data set of high quality (well-classified) polygons was used as a "training set" for this task. This consisted of an urban data set of over 46,000 polygons and a mainly rural data set of over 6,000 polygons. A very small number of these polygons were found to be still unclassified, and the contexts of any polygon adjacent to these were not considered. As the number of neighbours a polygon has increases, the number of possible topological arrangements of these neighbours (as described by the adjacencies between them) quickly grows. This causes a corresponding steep increase in computational complexity when attempting structural matching on these contexts. To avoid this, we arbitrarily excluded polygons with more than 10 neighbours from the training set. Many of these contexts would provide poor templates anyway, as their more complex topologies are, in general, more unique.

Each time a context was found in the data set that was not isomorphic with an existing template context, that context was recorded as a template itself. Associated with each template is a record of the number of polygons of each feature code that it structurally matched. For any given polygon P (within the training data) whose context matches template T, the probability that P is of class X can be calculated as:

$$\textbf{Probability P is of class X} = \frac{\textbf{\# polygons of class X matching T}}{\textbf{\# polygons matching T}}$$

Obviously, the larger and more representative the training set that is used, the more confident we can be about the results.

Figure 6. *Chart showing statistical information on all templates generated from training data.*

Figure 6 shows a scatter graph of all the templates derived from the training data, plotted as the number of matches a particular template achieves against the percentage of those matches attributable to the feature code which provides the greatest number of matches for that template. The X-axis measures the confidence in the recorded classification probabilities. The geometrical patterns evident along this axis are caused by the quantization inherent in the fractions being plotted along the Y-axis. It can be seen from the graph that the more matches a template achieves, the more likely it is to suggest an accurate classification. What this graph does not show is the distribution of the templates. There are over 15,000 templates plotted within the graph, but 12,000 of them occur only once within the data. These 12,000 points are all plotted at the same co-ordinate, (1, 100). It seems that a good proportion of polygons can be precisely classified based on a relatively small number of frequently occurring templates. Most templates have had a low number of matches, so that large training sets of data will be required to evaluate their usefulness.

Future Work

Our next task is to evaluate the usefulness of our structure matching technique as a classification tool. This involves using a set of templates derived from training data, as described in the previous section, to classify objects within test data sets.

A logical extension of the current fine-grained structure matching technique would be to investigate a more generalised form of mapping. This might involve allowing matches within a certain threshold. Machine learning algorithms could be used to implement a more generic classification system. This would allow additional information on the context of an object to be used: length of shared borders, area, and distances between centres of polygons could be considered. The neighbours' neighbours etc. could also be examined. A learning algorithm would decide which attributes are more relevant.

Conclusion

We began by establishing the need for an automated tool for classifying polygons in cartographic data, and introduced the notion of using analogy to infer the required information. The Cartographic Structure Mapping Algorithm has demonstrated the ability to support this transfer of knowledge between domains within a geographical framework. Many polygons can be classified based on a small number of common templates, but much training data is required to fully evaluate our current model. It is envisaged that broadening our representation of context and using machine learning to generate a rule-based expert system will produce improved results. Of course, analogy alone cannot be expected to solve the classification problem, rather, it will be a classifier playing a part in a robust decision making process that models the human mind.

Acknowledgements

We would like to thank Ordnance Survey (Great Britain), for the use of their DNF data sets.

References

Bohan A, O'Donoghue D, 2000, "A Model for Geometric Analogies using Attribute Matching", in *Proceedings of AICS 2000 – 11th Artificial Intelligence and Cognitive Science Conference*, NUI Galway, Ireland

Evans TG, 1968, "A Program for the Solution of a Class of Geometric Analogy Intelligence-Test Questions", in *Semantic Information Processing* Ed M Minsky (MIT Press)

Keane MT, 1990, “Incremental Analogising: Theory and Model”, in *Lines of thinking: Reflections on the psychology of thought (Vol. 1)* Eds KJ Gilhooly, MT Keane, R Logie, G Erdos (Wiley, New York)

Keyes L, Winstanley AC, 2001, “Using Moment Invariants for classifying shapes on large-scale maps” *Computers, Environment and Urban Systems* **25** 119 - 130

O’Donoghue D, Winstanley AC, 2001, “Finding Analogous Structures in Cartographic Data”, *4th AGILE Conference on Geographic Information Science*, Czech Republic

Ordnance Survey, The Digital National Framework, http://www.ordsvy.gov.uk/dnf