3D Visualization of Relation Clusters from OWL Ontologies

Jahangheer S. Shaik[a], Cartik R. Kothari[b], David J. Russomanno[a], and M. Yeasin[a]

[a]Department of Electrical and Computer Engineering
Herff College of Engineering
The University of Memphis
Memphis, TN 38152-6574

[b]Department of Computer Science
Neumont University
South Jordan, UT 84095

Abstract *- This paper presents an enhanced methodology to visualize clusters of relations from OWL ontologies. The relations are clustered by comparing their intrinsic semantics. The intrinsic semantics of every relation from an input ontology is explicitly specified by a framework of 32 common elements; each element captures a specific aspect of the relationship between a relation's domain and range. Using this framework, each relation can be represented in a 32 dimensional "relation space." Relation clusters in 32 dimensions are projected to 3 dimensions using an automated 3D star coordinate visualization technique. Results from the application of the proposed algorithm to visualize relation clusters from the IEEE SUMO Ontology are presented in this paper; and discussed in the context of their potential utility to knowledge reuse and interoperability on the Semantic Web.*

Keywords: Ontologies, OWL, Relation Clusters, Coordinate-Based Visualization.

I. Introduction

Knowledge ontologies provide a formal mechanism to conceptualize declarative knowledge. Ontologies include associations, also known as relations, and classes that can model both domain-specific and domain-independent concepts of some universe of discourse. To capture the intrinsic semantics of every relation from an input ontology it often requires a high-dimensional relation space. The visualization of relations defined within an ontology may lead to a better understanding, re-use, and interoperability of knowledge repositories.

A simple idea is to use the expressive power of a 3D representation to provide visualization of relations for a knowledge engineer. Effective visualization techniques enable the user to observe and detect the underlying relation distributions, patterns and similarities. One of the popular approaches to the visualization of complex, high-dimensional data is the use of a data projection method. The projected data, while providing enhanced understanding, should also preserve the local and global attributes of the data. To be pragmatic, projection techniques must be scalable, computationally efficient and free of human intervention. This paper presents an automated, 3D visualization algorithm to visualize relations that have been described with 32 elements to capture meta-knowledge about a set of relations from input OWL ontologies.

Data projection methods attempt to take data from a high-dimensional space and map it into a low-dimensional space with minimum error [1-4]. A great deal of effort is devoted to this subject and a number of useful methods have been reported in the literature with varying degrees of success [1, 2, 4-12]. One of the popular methods in data projection and visualization is the star coordinate system [9].

When using 3D star coordinates, each dimension is represented as a vector radiating from the center of the sphere to the circumference of the sphere. All the dimensions are normalized between [0 1]. Each data point (relation) has its dimensions (32 elements describing meta knowledge about the relation) lying along these vector axes radiating into 3D space. Objects lying on these axes are resolved into components along the *x*, *y*, and *z* coordinate axes, respectively. Individual contributions of objects pertaining to a data point along these axes are summed to obtain the points along these axes. Hence, a multi-dimensional data point is now represented in 3D. Initially, all the vectors are placed on a 3D sphere with equal weights given to all vectors. If no proper pattern emerges, the choices of the angles, as well as the lengths of the vectors are chosen randomly. In this iterative process, the vectors are rearranged in the 3D space at random angles and lengths. For each combination of angles and lengths, data is projected differently in 3D space. One or more of these combinations may produce a meaningful pattern. If a pattern emerges, depending on the orientation of the vectors and their length, one can infer the relationships between the dimensions and also their relative importance.

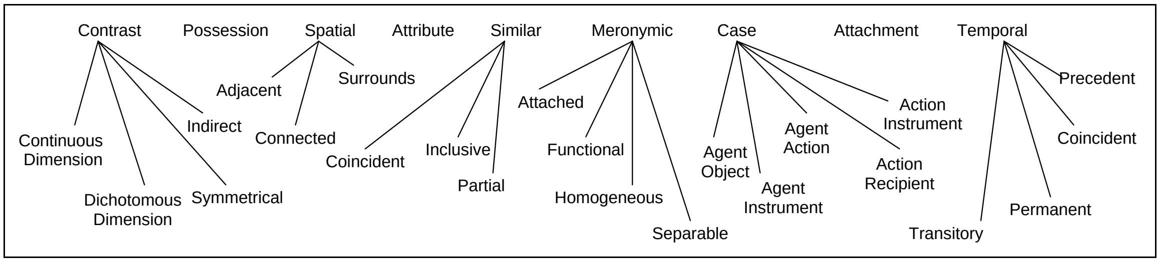

Figure 1: Two–tier taxonomy of elements to define relation semantics (RSEP elements)

A. Elements and Intrinsic Nature of Relations

In this paper, elements are used to explicitly describe the intrinsic nature of relations. Consider the relation instance *made-of* (*bicycle-tire*, *rubber*). Elements can be used to explicitly describe several aspects of the relationship between the domain and the range in a given relation. The *rubber* entity cannot be separated from the *bicycle-tire* entity in the *made-of* instance. In general, this aspect of the relationship between the domain and range of the relation *made-of* can be described by the *separable* element, which in this example is assigned a value *no* to denote "no, the entities in *made-of* cannot be separated from one another." Similarly, the fact that *rubber* is a component of the *bicycle-tire* can be specified by another element *meronymic*, which is assigned a value *yes* to describe this relation. The fact there is no temporal or time-based aspect to the relationship between the domain and range entities in the *made-of* relation is specified by a third element *temporal*, which is assigned *N/A*, or not applicable.

In this manner, different aspects of the relationship between the domain and range entities in a relation can be explicitly represented using elements, which take appropriate values from the value set {*yes*, *no*, *N/A*} to describe the intrinsic nature of relations. Elements provide a framework for better understanding of the intrinsic nature of relations, which can be utilized by knowledge modelers for fine-grained semantics in ontology development.

Kothari and Russomanno [13] have developed a two–tier taxonomy of elements that can be used to explicitly specify the intrinsic nature of relations in ontologies. Comprising 32 elements in all, this element taxonomy is shown in Figure 1.

These elements are used by a knowledge elicitation prototype called the Relation Semantics Elicitation Prototype (RSEP) [14] to elicit appropriate values to describe relations (extracted from input ontologies) from knowledge providers. Subsequently, these elicited values are appended to the input ontologies, where the intrinsic semantics of relations have not been previously specified. Because of their association with RSEP, these elements are called RSEP elements.

Figure 2 shows an example of the explicit specification of the intrinsic nature of a relation *ancestor* (*Organism*, *Organism*) from the IEEE Suggested Upper Merged Ontology (SUMO) [15] using RSEP elements. The remainder of the paper is organized as follows: Section II provides a brief description of the relation cluster methodology. Section III describes the proposed visualization technique and Section IV provides an analysis of the results. Finally, Section V provides concluding remarks.

II. Relation Clusters

Elements provide an organizing framework to capture additional semantics of relations, which can be used to quantify similarities and differences among relations. This is illustrated by the development of relation clusters, which partition the "relation space" in an input ontology.

Given value assignments for the 32 RSEP elements for relations in an input ontology, each relation can be visualized as a point in a 32 dimensional "relation space." Differences in the intrinsic semantics of these relations can be quantified in the form of Hamming distances [16] between them. Algorithm 1 specifies how relations in an input ontology can be clustered using Hamming distances.

The Hamming distances between every relation pair in the input ontology can be coupled with a suitable threshold value to decide if a given relation pair can be clustered together. The Agglomerative Hierarchical Clustering Algorithm [17] has been implemented in a prototype (called the Relation Clustering Prototype) [18] to partition the "relation space" of an input ontology into relation clusters.

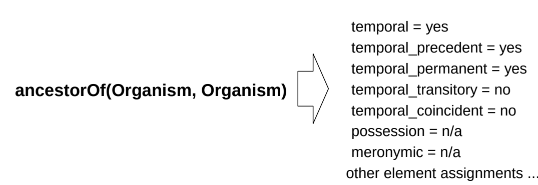

Figure 2: Example element specification

The Relation Clustering Prototype outputs clusters of relations from an input OWL ontology in which the intrinsic semantics of relations have been explicitly specified by RSEP elements. Figure 3 displays a sample output from testing the Relation Clustering Module on the IEEE SUMO. Note that the Relation Clustering Module outputs its results in textual format, detailing the clusters, the inter-cluster Hamming distances and the relations in each cluster. This is because these clusters (and the relations in them) are in 32 dimensions and cannot be easily rendered into a graphical display format.

Algorithm 1: Cluster Ontology Relations.

Input	A set of relations with element value assertions to describe each relation and a threshold value *T*
Step 1	For any given pair of relations, compare corresponding element values assigning a score of 0 if they are identical, +1 if both of them are either *yes* or *no* but not identical or +2 if they are not identical and one of them is *N/A*
Step 2	Sum up these scores to compute discrimination coefficient for the given pair of relations
Step 3	Perform Steps 1 and 2 for all pairs of relations in the input set and obtain discrimination coefficients for each pair
Step 4	Place every relation in the input set in its own cluster
Step 5	Given the set of discrimination coefficients, sort this list in ascending order keeping track of the relation pairs they correspond to
Step 6	For a given discrimination coefficient *D*, cluster two clusters together if *D* ≤ *T* and connect this new cluster to its children
Step 7	Repeat Step 6 until all the discrimination coefficients have been exhausted
Step 8	Repeat Steps 5 ~ 7 for increasing values of *T* until all the relations are in one cluster
Output	A tree of clusters

```
Cluster 1:
1 wife
14 spouse
48 husband
51 holdsDuring
-------------
Cluster 3:
16 sister
18 sibling

Cluster 1 → Cluster 3 = 2
```

Figure 3: Relation clustering module output

The Star 3D projection technique has been used to reduce the 32 dimensional relation clusters into 3 dimensions that can be easily displayed for human interpretation. The next sections detail the steps of the star 3D visualization technique and its implementation to visualize the relation clusters from IEEE SUMO.

III. 3D Relation Visualization Method

The proposed visualization method has three major components. First, for the sake of clarity, the mathematical formulation of the 3D star coordinate projection is presented. Following this, the 32 dimensional element representations of the relations from IEEE SUMO are projected onto a 3D format using the proposed 3D star coordinate method. To obtain the relation clusters, the SOM algorithm is applied in the projected space. Finally, the quality of clusters is obtained using the Dunn and Davies-Bouldin index [19] and has been used as a measure to choose the best configuration of the 3D projection. The projection providing best quality clusters is selected as the better projection among other alternatives. The subsequent subsections describe the development and formulation of the proposed algorithm in detail.

A. 3D Star Coordinate Projection

The schematic diagram of the proposed 3D star coordinate based projection is shown in Figure 4. It is assumed that all the dimensions are radiating from the center of a hypothetical sphere at random angles in 3D space and are of random length. The attributes of individual data points are scaled and rotated accordingly.

For a given data point, each of its attributes (points along all dimensions of that data point) is multiplied with the component along a particular direction (x, y, z). Individual contributions are integrated along the direction to form a projected space as shown in equation 6.

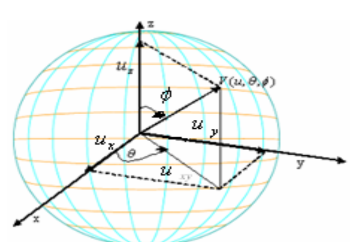

Figure 4: 3D Star Coordinate Projection

Various stages and steps involved in the formation of 3D star coordinate system are succinctly summarized as follows:

Step I: Initialization:

1. Arrange the coordinate axes on the sphere with all vectors radiating into the 3D space.
2. The angles between the vectors are random. Each of the dimensions is assumed to be along these vectors.
3. Star coordinate system may be mapped to Cartesian coordinate system by defining a point in the 3D space representing the origin (O_x, O_y, O_z) and n 3 dimensional vectors $A_n = \langle a_1, a_2, a_3, \ldots\ldots\ldots a_n \rangle$ representing the axes.
4. The lengths of the vectors are random. The lengths may vary between one unit vector and 5 times a unit vector.

Step II: Computation of Projection

The projection of the data point in n dimensions to 3 dimensions is obtained by summing up the unit vectors (u_x, u_y, u_z) on each coordinate weighted by their respective data element.

$$u_{xy} = \sqrt{\left(u_x{}^2 + u_y{}^2\right)}, \qquad [1]$$

$$u = \sqrt{u_{xy}{}^2 + u_z{}^2}\,, \qquad [2]$$

$$u_x = u \sin\phi \cos\theta\,, \qquad [3]$$

$$u_y = u \sin\phi \sin\theta\,, \qquad [4]$$

$$u_z = u \cos\phi\,. \qquad [5]$$

Let '*N*' be the number of data points and '*n*' be the dimension of each feature that needs to be visualized. The data matrix '*D*' is of dimension *N x n* and its elements d_{ij} represent the components of *n* dimensional data points.

$$\begin{aligned}
&\mathit{for}\ \mathrm{j}=1:\mathrm{N},\\
&\mathrm{X_p} = O_x + \sum_{i=1}^{n} u_{xi}(d_{j,i} - \min(D(:,i))),\\
&\mathrm{Y_p} = O_y + \sum_{i=1}^{n} u_{yi}(d_{j,i} - \min(D(:,i))),\\
&\mathrm{Z_p} = O_z + \sum_{i=1}^{n} u_{zi}(d_{j,i} - \min(D(:,i))),\\
&P_j(X,Y,Z) = (X_P, Y_P, Z_P).\\
&\mathit{end}
\end{aligned} \qquad [6]$$

O_x, O_y and O_z are the coordinates defining the present origin of the system. The notation $\min(D(:,i))$ stands for the minimum value in all rows and i^{th} column. X_P, Y_P, Z_P are the projections along (x, y, z) axes, respectively. The vectors in $'n'$ dimensional space are projected into 3D space given by $P(X,Y,Z)$. It is now easy to visualize various dimensions radiating at random angles in a pseudo sphere, more than one such combination provides insight into the underlying distribution of the data while providing relationships between various attributes involved.

B. Best Projection Algorithm

The current approaches reported in [2, 9] require human intervention to decide the best configuration of projection from the high-dimensional space to the low-dimensional space. To alleviate this problem, an automated algorithm is developed to find the best projection for visualization of relation clusters from the IEEE SUMO. The algorithm for choosing the best projection is summarized as follows:

Let us consider Q_0 being the variable carrying an initial measure of the quality of clusters and variable *I* is the number of iterations:

$\mathit{For}\ i = 1 : I,$

Step 1: Initialize the algorithm as shown in 3D star projection algorithm.
Step 2: Obtain $P(X,Y,Z)$ as given by equation 6, a projection in 3D space.
Step 3: Apply SOM clustering algorithm in projected space.
Step 4: Apply validation measure and calculate quality of clusters Q_1 given by Dunn index and Davies-Bouldin index.
$\mathit{if}\ (\mathrm{Q}_1 > Q_0)$ then, $Q_0 = Q_1$
Store parameters for present iteration.
end,
end.
Select parameters corresponding to the highest Q_1.

IV. Empirical Analysis

The objectives of the empirical analysis are as follows:

i) To reduce the dimensionality of 32 dimensional relational space into 3 dimensions ensuring the best projection without human intervention.
ii) To find the similarity among relations in the reduced dimensional space.

Since the orientation of vectors is chosen randomly, it is easy to note that the proposed algorithm must iteratively find the best configuration based on cluster quality indices using the algorithm outlined in the previous section. Figure 5 (a) and (b) show the intermediate results with some emergence of a pattern. While the results from Figures 5 (a) and (b) show some pattern, they fail to provide crisp boundaries between relation clusters. Nevertheless, this shows the evolution of the formation of relation clusters in this iterative process. The result in Figure 6 shows the best projection obtained using the proposed automated algorithm. A visual analysis of Figure 6 indicates the formation of crisper relation clusters.

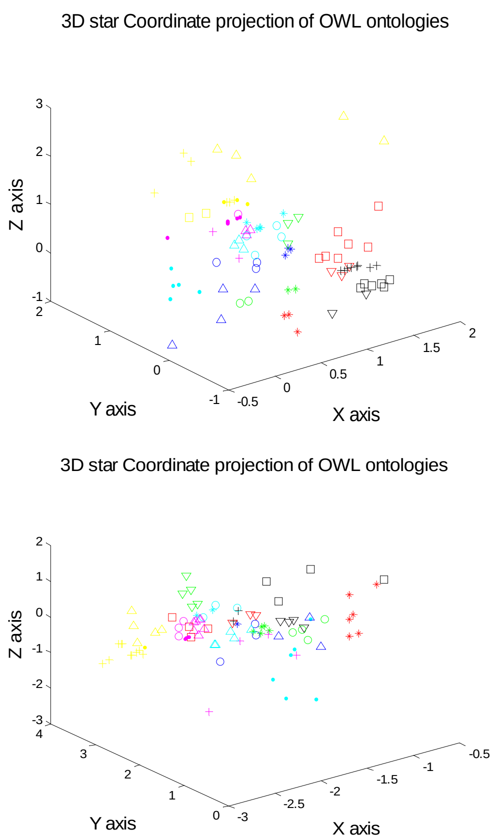

Figure 5: The evolution of relation clusters in 3D space showing the intermediate results

Table 1 shows the clustering of relations as obtained by SOM applied on top of the proposed 3D star coordinate projection algorithm. Better quality clusters are obtained when the number of clusters is chosen to be 24 as specified by the Dunn index and Davies-Bouldin index, respectively [19]. Most of the relation objects that fell into a cluster have some similarity among them as shown in Table 1.

Information conveyed via visualization may be of utility to a knowledge engineer when performing the following tasks: i) assessing the semantics of a set of relations available in a given ontology; ii) determining whether the definition of a new relation as an extension of a given ontology is warranted based upon its similarity to already defined relations; iii) ontology integration in which similarities and differences of concepts and relations defined in two ontologies are compared for possible integration into a single, unified ontology; and iv) interoperability of agents using knowledge repositories that have committed to different ontologies.

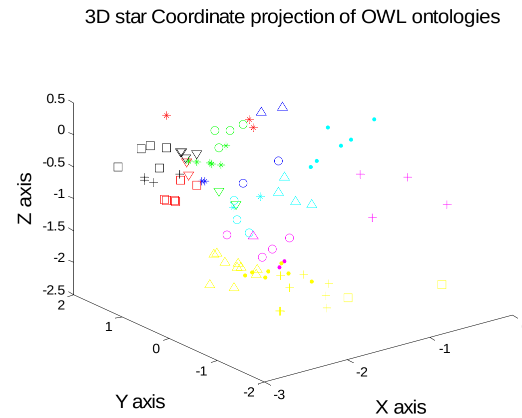

Figure 6: The best projection of relation clusters using the proposed 3D visualization algorithm

V. Conclusions

This paper developed and validated an enhanced 3D visualization algorithm to uncover the intrinsic semantic relations in an ontology. An algorithm to choose the best configuration of the 3D projection was developed by using a number of cluster validation indices. In particular, the Dunn index and the Davies-Bouldin index are combined in a weighted fusion scheme to yield a unified score for cluster quality. The number of iterations was set to 100 and the projection configuration corresponding to the highest clustering index is considered as the best projection. The utility of the proposed 3D algorithm is demonstrated by visualizing the relations clustered from the SUMO ontology. Any input OWL ontology could be used provided that its relations have values asserted for the 32 relation elements.

Empirical analysis indicates that the automated 3D visualization algorithm revealed the implicit semantic similarities among relations and enabled enhanced visualization of relations from an ontology. The current version of the automated 3D visualization algorithm relies on finding the best projection configuration on the basis of quality of clusters produced.

A good cluster quality does not always guarantee the intended result as the quality is estimated based on distance measures. It may be possible to incorporate a method in which semantic meaning of the formed clusters is used as a measure of the quality of a projection. It is often seen that relationships among data points are nonlinear; hence, a good non-linear projection scheme such as Locally Linear Embedding (LLE) on top of 3D visualization might help to unfold such nonlinear relationships. These ideas will be addressed in future study, as well as from the analysis of applying the technique on other input ontologies.

Table 1: IEEE SUMO relation clusters generated by the proposed 3D visualization algorithm

Cluster Number	Relations
1	wife, spouse, husband
2	wants, needs, holds during
3	property, measure, lineMeasure
4	height, expressedInLanguage
5	result, agent, frequency, causes
6	earlier, during, before, finishes
7	meetsTemporally, ancestor, developmentalForm, relatedInternalConcept
8	parallel, larger, identicalListItems, equivalentContentinstance, equivalentContentClass, equal
9	monetaryValue, hasPurpose, date
10	sister, sibling
11	desires, daughter, brother, parent
12	believes, considers attends
13	knows, instance, inScopeOfInterest, hasSkill, disjoint
14	properlyFills, fills
15	mother, father
16	faces, connected, angleOfFigure
17	inhabits, exploits, employs
18	uses, copy, coOccur
19	top, surface, side
20	wears, time, resource, leader, instrument, home
21	traverses, publishes, inhibits, grasps, experiencer, editor, authors
22	transactionAmount, successorAttribute, member, material, inList, hole, geometricPart, engineeringSubcomponent, element
23	overlapsTemporally, beforeOrEqual
24	temporalPart, stays, starts, refers, part

Acknowledgments

The authors acknowledge the Herff College of Engineering Fellowship program and the faculty start-up grants from The University of Memphis for partially funding this research.

References

[1] D. Asimov, "The grand tour: a tool for viewing multidimensional data," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, pp. 128-143, 1985.

[2] J. Bertin, *Graphics and Graphic Information Processing*: Walter de Gruyer & Co, Berlin, 1981.

[3] I. Guyon, "An Introduction of Variable and Feature Selection," *Journal of Machine Learning Research*, 3(Mar), pp. 1157-1182, 2003.

[4] A. Inselberg, "N-dimensional graphics, part I-lines and hyperplanes," IBM Scientific Center, Los Angeles, CA, 1981.

[5] D. D. Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. P. W. Duin, "Supervised Locally Linear embedding," *ICANN*, pp. 333-341, 2003.

[6] J. Shaik and M. Yeasin, "Functionally Classifying Genes from Microarray Data Using Linear and Non-linear Data Projection," To appear in *Proc. of the 4th ACS/IEEE Int'l Conf. on Computer Systems and Applications (AICCSA 06)*, 2006.

[7] C. Unsalan and A. Ercil, "Shapes of features and a modified measure for Linear Discriminant Analysis," *OCPR00*, vol. 2, pp. 410-413, 2000.

[8] E.M. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," *A Practical Approach to Microarray Data Analysis*, Kluwer: Norwell, MA, 2003.

[9] E. Kandogan, "Star Coordinates: A Multi-Dimensional Visualization Technique with Uniform Treatment of Dimensions," In *Proceedings of the IEEE Information Visualization Symposium*, 2000.

[10] O. Kouropteva, O. Okun, and M. Pietikainen, "Supervised locally linear embedding algorithm and support vector machine," In *ESANN'2003 Proceedings- European Symposium on Artificial Neural Networks*, pp. 229-234, 2003.

[11] G. Loftus, "A picture is worth a thousand p values: On the irrelevance of hypothesis testing in the micro computer age," *Behavior Research Methods, Instruments, and Computers*, vol. 25, no. 2, pp. 250-256.

[12] J. Beddow, "Shape coding of multidimensional data on microcomputer display," In *Proc. of 1st Int'l Conf. on Visualization*, San Francisco, CA, 1990.

[13] C. R. Kothari and D. J. Russomanno, "A Relation Semantics Elicitation Prototype for the Semantic Web," In *Proc. of the 2005 Int'l Conf. on Artificial Intelligence (ICAI 2005)*, Las Vegas, NV, pp. 682-688, 2005.

[14] C. R. Kothari, D. J. Russomanno, and P. Tran, "Interactive Elicitation of Relation Semantics for the Semantic Web," In *Proc. International Conference on Systems, Computing Sciences and Software Engineering*, Bridgeport, CT, 2005.

[15] I. Niles and A. Pease, "Towards a Standard Upper Ontology," In *Proc. of the 2nd Int'l Conf. on Formal Ontology in Inference Systems (FOIS 2001)*, pp. 2-9, 2001.

[16] R. Hamming, "Error-detecting and Error-correcting Codes," *Bell System Tech. J*, vol. 29, pp. 147-160, 1950.

[17] J. Hartigan, *Clustering Algorithms*, New York: John Wiley, New York, 1975.

[18] C. R. Kothari and D. J. Russomanno, "Clustering Relations within OWL Ontologies," In *Proc. of the 9th IASTED Int'l Conf. on Software Engineering and Applications (SEA 2005)*, Phoenix, AZ, pp. 36-41, 2005.

[19] D.L. Davies and D.W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, 224–227, April 1979.