Graph Matching by Self-organizing Feature Maps

Stéphane HERBIN *

Ecole Normale Supérieure de Cachan
Centre de Mathématiques et de Leurs Applications
Groupe DIAM
CNRS, URA 1611
61, Av. Président Wilson
94235 Cachan Cedex, France

Abstract

Many problems in pattern recognition can be formulated as searching a correspondence between two graphs. The exact solution is often limited by a combinatorial explosion, and suboptimal iterative algorithms may give good approximations. This paper proposes an application of the self-organizing feature map (SOFM) algorithm to graph mapping and gives a set of measures able to characterize the neighborhood preservation ratio of the resulting correspondence. The role of graphs as a representational object in neural modelling is discussed.

1 Graph matching

1.1 Correspondence between graphs

Graph is a useful data structure able to represent relational information between entities. It is defined by a set of vertices V and a set of edges E between vertices. The stucture of a graph $G = (V, E)$ is characterized globally by an *adjacency* matrix $A(G)$, where a non zero element at the i-th line and j-th column means that vertices i and j are somehow related. We will stay in the framework of undirected graphs, which implies that $A(G)$ is symmetric.

Many problems, especially in pattern recognition and computer vision, can be formulated as a correspondence between two graphs. Define a *matching* to be a bipartite graph between two sets of vertices V and V' from two graphs G and G'. The adjacency matrix of the resulting global graph has the form

$$\begin{pmatrix} A(G) & B^T(G,G') \\ B(G,G') & A(G') \end{pmatrix}$$

where $B(G, G')$ is the adjacency, possibly real valued and not square, matrix of the bipartite graph. Finding a matching between two graphs is equivalent to finding a matrix B subject to some constraints.

A special case of matching is a *mapping*, which is a one to one correspondence between vertices. The elements of the B matrix, then, contain only one non zero value per column.

Many interesting combinatorial problems in graphs (i.e. deciding whether one graph contains a specific subgraph, finding the maximal complete subgraph) are NP-complete. The problem of deciding whether two graphs are isomorphic is not proved to be NP-complete but is, however, conjectured to be NP [5]. This issue leads to the idea that an iterative, possibly suboptimal, algorithm may be useful in such optimization problems. The algorithm presented in this paper is designed to find a mapping which preserves globally neighborhood relationship between vertices. The optimization space is the set of real valued B matrices between graphs in which the mapping will be embedded.

1.2 Characterization of mappings

This section introduces the concept of *continuity* of mapping between two graphs of any kind and proposes a set of measures to characterize it.

*email: herbin@diam1.ens-cachan.fr. Part of this work was conducted while the author was at Beckman Institute, Theoretical Biophysics Group, University of Illinois at Urbana-Champaign, 405 N. Mathews Ave.,Urbana, IL 61801.

Neighborhood sequences

In general topology, a mapping will be said to be continuous if the inverse mapping of any open set in the image space is an open set in the input space. In particular, any neighborhood of a given mapped point should have as its inverse mapped image a neighborhood in the input space. The idea in giving a scale to the continuity of a mapping is to measure the violation of these neighborhood requirements.

Let m be a mapping from an input space $\mathcal{A}$ onto an image space $\mathcal{B}$. For any point y in $m(\mathcal{A})$, let $\mathcal{V}_B(y)$ be an ordered basis of neighborhoods of y, $\mathcal{V}_B(y) = (V_1(y), V_2(y), \ldots)$, and for any point x in $\mathcal{A}$, let $\mathcal{W}_A(x)$ be an ordered basis of neighborhoods of x, $\mathcal{W}_A(x) = (W_1(x), W_2(x), \ldots)$. For all x and y, we have the order relations $V_k(y) \subset V_{k+1}(y)$ and $W_k(x) \subset W_{k+1}(x)$.

A measure of continuity of the mapping will be derived from a comparison between the two ordered *neighborhood sequences* $\mathcal{W}_A(x)$ and $m^{-1}\{\mathcal{V}_B(m(x))\}$. The continuity of the inverse mapping will be characterized by a comparison between the two neighborhood sequences $m\{\mathcal{W}_A(x)\}$ and $\mathcal{V}_B(m(x))$.

When the two spaces $\mathcal{A}$ and $\mathcal{B}$ possess a metric, one approach is to compare the neighborhood diameters in all sequences [1]. Set-theoretic comparisons between neighborhoods are more suited to combinatorial structures such as graphs.

Set-theoretic comparisons

We will consider from now on a graph matching as a one-to-one correspondence mapping between nodes. The basic neighborhoods $\mathcal{W}_A(x)$ and $\mathcal{V}_A(y)$ can be defined as sets of balls centered in x and y with increasing radii; for instance, $V_k(y) = \mathbf{B}_B(y, k)$ and $W_k(x) = \mathbf{B}_A(x, k)$, if the diameters of the input and output graphs are similar. A rescaling of the radii may be necessary to fit the same number of neighborhoods into both graphs.

The difference between two neighborhoods with the same rank in the neighborhood sequences could be measured by computing the cardinal of the sets resulting from various set theoretic operations, instead of metric comparison. Define a set of numbers $Q_{\text{direct}}(x, k)$, based on symmetric set difference as:

$$Q_{\text{direct}}(x, k) = 2\,\frac{\left|m^{-1}\left(V_k(m(x))\right) \cap W_k(x)\right|}{\left|m^{-1}\left(V_k(m(x))\right)\right| + |W_k(x)|}$$

Those numbers characterize the local neighborhood deformations of the direct mapping $m : \mathcal{A} \longrightarrow \mathcal{B}$. Similar numbers, $Q_{\text{inverse}}(y, k)$, can be defined for the inverse mapping, by comparing the two neighborhood sequences $m\{\mathcal{W}_A(x)\}$ and $\mathcal{V}_B(m(x))$. The optimum of the above numbers is obtained when $Q_{\text{direct}}(x, k) = 1$, for all couples (x, k).

The interesting property of the similar numbers measures defined in [1] was their ability to detect whether the intrinsic dimension of the output map was too small or too large to represent the whole input space; the numbers were symmetric around 0. It is possible also to produce set theoretically numbers which will characterize the adequacy of the output map as

$$R_{\text{direct}}(x, k) = 2\,\frac{\left|m^{-1}\left(V_k(m(x))\right) \backslash W_k(x)\right| - \left|W_k(x) \backslash m^{-1}\left(V_k(m(x))\right)\right|}{\left|m^{-1}\left(V_k(m(x))\right)\right| + |W_k(x)|}$$

Those numbers characterize more globally the neighborhood deformation. The optimal mapping corresponds to a set of numbers which are close to 0. Negative numbers would mean that the mapping is locally contracting. Again, similar numbers could be defined to characterize the inverse mapping.

All those series of numbers can be averaged out to give a global characteristization of the neighborhood preservation ratio of the mapping. Some examples of the above measures can be found in [7].

2 Graph matching by SOFM

2.1 Self-organizing feature maps

Self-organizing feature maps (SOFM) is a simple adaptive algorithm able to perform a discrete and continuous representation of a high-dimensional space by trying to match iteratively the topology of a fixed space, the actual map, generally a one- or two-dimensional lattice, and the topology of an unknown high-dimensional vector space [8].

The formation of the map is designed to follow two principles: 1) neurons of the output layer tend to specialize to one input through a long range competition, and 2) neighboring output neurons tend to represent neighbor instances in the natural space. This objective can be achieved using a stochastic gradient-like algorithm, in which the modified coefficients at each step are the weights of the connections linking the most active output neuron and its neighbors to the input space. The resulting map is able to represent the most important

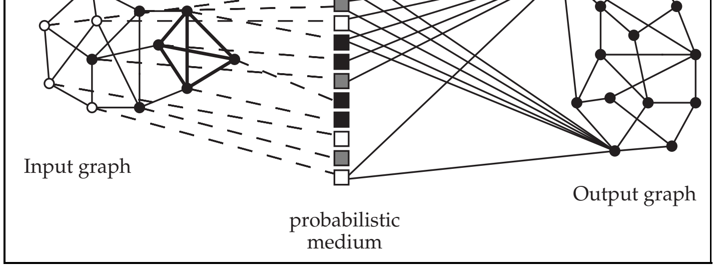

Figure 1: Correlated pattern approach of graph matching.

neighborhood relationships between the instances of the input space. If the metric of the input space is derived from a similarity distance, the algorithm ensures that neighboring neurons in the output map represent similar instances in the input space: this means that the *inverse* mapping, from the output layer to the feature space, is continuous. This may not be the case, however, of the direct mapping [10, 9].

SOFM can also be regarded as a *vector quantization* algorithm; the weights attached to each output neuron is an element that quantifies an input space by trying to minimize an error of the form $E[W] = \int ||\mathbf{v} - \mathbf{w}_{s(\mathbf{v})}||^r dP(\mathbf{v})$, where $\mathbf{v}$, is an input vector, $\mathbf{w}$, is a weight vector associated with neuron $s(\mathbf{v})$, the neuron having its weight vector closest to the input stimulus. The map, therefore, acts as a probability-geometry transducer since it encodes in a geometrically ordered way with greater care more probable events; the geometry of the input is used only implicitly in the probabilistic distribution of the input space. A probabilistic space can be used as a *medium* in which to encode geometric information which in turn will be represented later geometrically on the map. This is the property that will be used to construct the graph matching.

2.2 Graph as a set of correlated patterns

A graph is mathematical object representing *globally* the relations that may exist between a given set of elements. In this paper, the existence of an edge will mean that the two vertices linked are, somehow, correlated, and that this correlation is transitive. In the vocabulary of neural networks, the existence of an edge is the manifestation of the possibility that two nodes are active at the same time, and the transitivity property simply states that the activity may spread over the graph along the connecting edges. With this view in mind, a graph can be considered as the global summary of the *possible* patterns of node activities.

The patterns of activity that produce the gobal description of a graph may be recovered by simple means. An easy way to obtain them is by computing and combining the powers of the adjacency matrix $A^n(G)$, which counts the number of paths of length n joining two vertices [3]. The lines (or columns) of the matrix $P = \sum_{k=0}^{N} \gamma A^k(G)$, where γ is some decreasing coefficient in $]0,1[$ and N is a fixed spreading length, give a set of real valued vectors which can be considered as patterns of activities, each coordinate corresponding to a certain level of activity. It is the overall *correlation* between patterns of activity that produces the graph structure.

2.3 Matching of patterns

The self-organizing map algorithm is essentially an iterative gradient-like algorithm subject to a topological constraint; this means that one of its fundamental features is the continuous character of the space in which the gradient is computed. Graphs, however, are essentially discrete mathematical objects. The problem is thus trying to embed the discrete problem in a continuous world in which the iterative operations could be meaningfully performed.

The information used from the output space is in terms of distances between pairs of neurons; what is actually needed for the definition of its topology is simply an array of interneuron distances. In the case of graphs, the distance used between neurons is the shortest path length between vertices of a graph.

The only questionable space, therefore, is the input since all of the computations, i.e., the weight adjustments, are made in it. The description of a graph as a set of correlated patterns leads naturally to the choice of the input space as a d-dimensional vector space where d is the order of the input graph. The input data will then consist in a set of vector representation of correlated patterns.

Figure 1 shows the basic constituents of the probabilistic approach of the graph matching adjustment. The two graphs are considered completely connected by weighted connections, as in a classical feed-forward network scheme. The graph matching is defined as a bipartite valued graph described by a weight matrix. During the process some weights tend to vanish naturally, and some connections get reinforced.

The global definition of the graph matching follows 3 steps:

1) Create the set of input vectors corresponding to the correlated patterns.

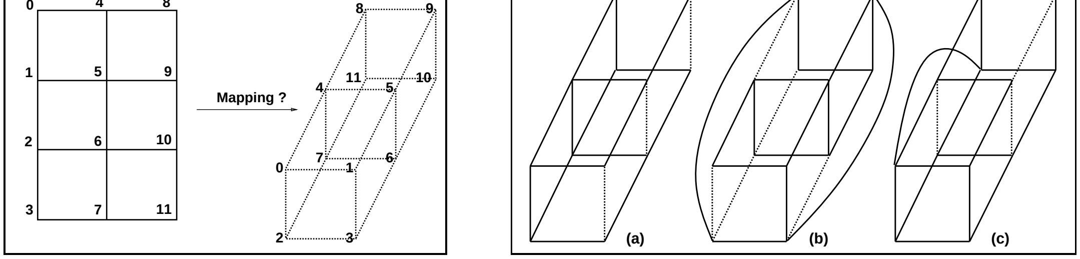

Figure 2: The problem of mapping a 3 × 4 grid on $P_3 \times C_4$ (left), and examples of the possible (right).

2) Run the SOFM.
3) Define the graph matching using the weights.

The intrinsic symmetry of the graph matching problem allows one to enforce the process to satisfy more bicontinuity constraints; the weight connections can be considered undirected, i.e., the connection linking the input node u to the output node v can be considered as also linking v to u. If W is the matrix of weights where the j-th column corresponds to the weights attached to the j-th output neuron, W^T, the transpose of W, is a matrix of weights where the i-th column corresponds to the weights attached to the i-th input node. Both the input and output graphs can alternatively exchange their roles; this would allow updating the weights considered in a given iteration as W and in a further one as W^T, increasing in this way the bicontinuity of the graph matching.

2.4 Example

In this section, we will study in detail the simple example of the mapping of a 3×4 grid onto the graph $P_3 \times C_4$, as depicted on Figure 2. In this problem, the two topologies can not match rigorously, although the grid is clearly a subgraph of the output graph. We will see that the definition of a best neighborhood preserving graph embedding is sometimes not a clear intuitive notion.

The definition of the mapping was first tested on the feed-forward probabilistic approach defined above. The input vectors were chosen to represent the subgraphs induced by the balls of diameter 1 centered on each vertex, and all of the subgraphs defined as simple edges. For instance, the subgraph spanned by the vertices $0, 1, 2, 5$ was chosen as an input, but not the subgraph spanned by $0, 1, 2, 3$.

The matching between the two graphs is defined by a matrix of weights joining each input to each output vertex. Table 1 shows a typical weight matrix obtained after learning. The number at the i-th row and j-th column is the weight corresponding to the connection between input vertex j and output vertex i. The largest weight of the input column determines on what output vertex the input node is mapped, if the matching is to be defined as a one-to-one correspondence, as a mapping. Some examples of one-to-one correspondences are depicted in Figure 2 showing different ways found by the algorithm in which to embed the grid on a $P_3 \times C_4$. Table 1 correponds to Figure 2(b).

Table 1: Typical weight matrix

	Input labels											
Output	0	1	2	3	4	5	6	7	8	9	10	11
0	.	.	.	.	.	.	.	0.45	.	.	0.52	1.00
1	.	.	.	.	.	.	0.68	.	.	0.20	1.00	0.20
2	.	.	0.65	.	.	0.20	1.00	0.20	.	.	0.20	.
3	.	.	.	0.33	.	.	0.94	0.99	.	.	.	0.33
4	.	.	.	.	0.19	.	.	.	0.99	0.73	.	.
5	.	.	.	.	.	0.46	.	.	0.17	1.00	0.53	.
6	.	0.59	1.00	0.40	.	.	0.19	.	.	.	.	.
7	.	.	0.13	0.80	.	.	.	0.73	.	.	.	.
8	0.18	.	.	.	0.99	0.18	.	.	0.68	.	.	.
9	.	.	.	.	0.49	1.00	0.43	.	.	.	.	.
10	0.27	1.00	.	.	.	0.66	.	.	.	.	.	.
11	1.00	0.49	0.27	.	0.41	.	.	.	.	.	.	.

More generally, the weight matrix describes a "one-to-many" correspondence made *continuous* by the learning algorithm. Note that the one-way algorithm inspired by the SOFM produces continuous but *not* bicontinuous maps in general: the direct image of a neighborhood in the input graph is not in general a neighborhood in the output graph (see for instance the neighborhood centered at input vertex 6). This limitation allows one in fact to discover some unusual ways in which to embed a 3×4 grid in the graph $P_3 \times C_4$; the most intuitive way to do it would be as shown in Figure 2(a). This embedding was not actually found by the one-way algorithm but by its bicontinuous version. The simpler continuous version algorithm produced only the two graph embeddings as drawn in Figure 2(b) and (c). By computing the continuity measures, it can be checked that embedding (a) is in fact less continuous than the two other embeddings; however, when the continuity measures are computed for the inverse and direct mappings, it can be checked that embedding (a) achieves the best average performance. These simple examples show that the best neighborhood preserving mapping is not realized necessarily by preserving the greatest number of edges.

3 What should graphs represent

This section examines the role of graphs as a representational object in neural modelling. Graphs have been used in computer science and engineering mainly in two ways: as a network and as a relational structure.

3.1 Graph as network

The expression "neural network" originates in the physiological evidence of the electrical nature of the nervous system activity and in the anatomical structure of the basic constituents of the neural matter. A directed graph, in this context, is a representation of the neural *circuitry.*

This paper advocates that, if graphs are to be used simply as a rather iconic representation of the neural anatomy, we are missing most of their expressing and structuring power.

Besides the fuzzy semantics assigned to the correspondences neuron-vertex and synapse-edge, there is little evidence that a single network, as used currently in the field of artificial neural networks, could be isolated as a meaningful functional unit dedicated to a specific task in the brain. The reason for that is twofold: 1) no neuron has a specific and immutable functional role since it may contribute, at the same time, to several concurrent functions, or to a single one depending on the context, and 2) no subnetwork can be really identified from the amorphous nervous anatomy, and especially in the cortex.

Let's detail this last argument in the light of one anatomical finding. It has been calculated that no neuron is farther than two or three synapses away from any other neuron in the cortex [2]; this experimental measure shows that the diameter of the total graph of real neural connections is less than 3. A result on random graphs shows that almost all graphs have diameter 2.[1] This means that a precise pattern of connectivities of the neural graph may be difficult to distinguish from a poorly structured random graph. The problem of isolating a functional graph in the neural graph is emphasized by another result saying that any graph can be embedded as an induced subgraph in a diameter-minimal[2] supergraph of diameter 2; this could suggest that any circuit network potentially exists in the global neural architecture, which is fortunate if the central nervous system wants to be adaptive, but dreadful for studying its functionalities.

3.2 Graph as relational structure

Most neurophysiologists believe that the electrical activity of the neurons is the main feature in the information processing of the brain. Electrical signals in neurons take the form of spikes with various frequencies. A first view considers a neuron as an input-output device which fires with a frequency correlated with the certainty that a given event has occurred. Another view, considers that the useful notion to explain the nature of the neural code is *correlation* between neuron firings; there seems to be more and more evidence of this fact [4, 6, 11]. Interactions between neurons are weak, even when neurons are adjacent and coactivated in relation to the same external stimulus. To transmit its information, each individual neuron requires the cooperation of many others firing *synchronously* with it.

These experimental observations suggest that the right level of description of the neural activity is in terms of transient *assemblies* of correlated neurons. Graph language and concepts can be useful to neural models if edges are thought of as representations of correlations among neuron activities. The graph representation of a neural network is only relational and does not intend to represent the details of the physical connectivity scheme (excitatory or inhibitory connections), nor the dynamics of the activity. Correlation is a global notion resulting from a posteriori data analysis; a node will still stand for a schematic neuron, but an edge in a correlation graph will represent simply the possibility that two neurons fire synchronously. Therefore, *correlation graphs*, by comparison to the thinner anatomical structures, are macroscopic descriptions. It seems obvious, however, that physically disconnected neurons cannot have any correlation, and therefore can not be linked in the graph.

[1] The precise result [3] requires some definitions: $\Gamma(n,p)$ is the set of graphs with order n whose edges are chosen independently with probability p. For a property $\mathcal{P}$, we say that a random graph $G \in \Gamma(n,p)$ has property $\mathcal{P}$ *almost surely* if Pr(G has property $\mathcal{P}$)$\to 1$ as $n \to \infty$. It can be proved that if $p^2 n - 2 \lg n \to \infty$ and $n^2(1-p) \to \infty$, then $G \in \Gamma(n,p)$ almost surely has diameter 2. The conditions are true in particular for $p = 1/2$.

[2] A diameter-minimal graph G is such that for all edges $e \in E(G)$, $diam(G - e) > diam(G)$.

3.3 Dynamic graphs

If the basis of the neural code is the concept of temporal correlation, the formal tools that should be developped for neural modelling ought to include both dynamic and relational expressing capabilities. More precisely, they should be able to express explicitly the temporal evolution of the relational structure of neural activity. What should be developped is a theory of dynamic correlation graphs, independently of the physical, electrical and chemical phenomena that produced the correlation.

The only attempt to fulfill part of this objective is the pioneering work of von der Malsburg and Bienenstock [12, 13] who introduced the concepts of fast synaptic plasticity and dynamic links. Their ideas gave rise to several applications in pattern recognition, but could not generalize to effective mathematical models actually needed if one wants to understand, describe and reproduce the phenomenology of neural activity.

One ambition of this paper is to promote the use of graphs, and more generally of combinatorial relational structures, as the basic mathematical objects for neural modelling, and leave aside the notion of network, too readily associated with the circuitry of the nervous system. The inherent properties of graphs have not been generally studied per se in the neural network litterature, and there is still a huge amount of results to explore that could shed light on some aspects of the nervous system behavior and organization.

References

[1] H.-U. Bauer and K. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, **3**:570–579, 1992.

[2] V. Braitenberg and A. Schuz. *Anatomy of the Cortex*, volume **18** of *Studies of Brain Function*. Springer Verlag, Berlin, 1991.

[3] F. Buckley and F. Harary. *Distance in Graphs*. Addison-Wesley, Redwood City, Ca, 1990.

[4] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H.J. Reitbock. Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, **60**:121–130, 1988.

[5] M.R. Garey and D.S. Johnson. *Computers and Intractability — A guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.

[6] C.M. Gray and W. Singer. Stimulus specific neuronal oscillations in the cat visual cortex. *Proceedings National Academy of Science U.S.A.*, **86**:1698–1702, 1989.

[7] S. Herbin. Representation and formation of a structure of concepts – a graph theory and neural network approach. Master's thesis, University of Illinois at Urbana-Champaign, 1992.

[8] T.K. Kohonen. *Self-organization and Associative Memory*. Springer Verlag, Berlin, third edition, 1989.

[9] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, **7**(3):507–522, 1994.

[10] H. Ritter and K. Schulten. Convergence properties of kohonen's topology conserving maps: fluctuations, stability, and dimension selection. *Biological Cybernetics*, **60**:pp.59–71, 1988.

[11] E. Vaadia, E. Ahissar, H. Bergman, and Y. Lavner. Correlated activity of neurons: a neural code for higher brain functions? In J. Kruger, editor, *Neuronal Cooperativity*, Springer Series in Synergetics. Springer Verlag, 1991.

[12] C. von der Malburg. Synaptic plasticity as basis of brain organization. In J.-P. Changeux and M. Konishi, editors, *The Neural and Molecular Bases of Learning*. John Wiley & Sons, 1987.

[13] C. von der Malsburg and E. Bienenstock. Statistical coding and short-term synaptic plasticity: a scheme for knowledge representation in the brain. In E. Bienenstock et al., editors, *Disorderd Systems and Biological Organization*. Springer-Verlag, 1986.