Modification of the Growing Neural Gas Algorithm for Cluster Analysis

Fernando Canales and Max Chacón

Universidad de Santiago de Chile; Depto. de Ingeniería Informática,
Avda. Ecuador No 3659 - PoBox 10233; Santiago - Chile
fernando.canales.cifuentes@gmail.com, mchacon@diinf.usach.cl

Abstract. In clusters analysis, a problem of great interest is having methods that allow the representation of the topology of input space without the need to know additional information about it. This gives rise to *growing competitive neural* methods which are capable of determining the structure of the network autonomously during the process of training. This work proposes a variation of the *Growing Neural Gas* (*GNG*) algorithm, calling *GNG with post-pruning* (*GNG-PP*), and a method of clustering based on the search for topological neighborhoods generated by the former. These were combined in a three-phase process to clustering the *S&P100* set, which belongs to the macroeconomic field. This problem has a high dimensionality in the characteristics space. Its results are compared to those obtained by *SOM*, *Growing Cell Structures* (*GCS)*, and a non-neural method. Evaluation of the results was made by means of the *kappa* coefficient, using as evaluation set the GICS industrial classification. The results show that when using the proposed methods the best clustering are generated, obtaining a kappa coefficient of *0.5643* in the GICS classification.

Keywords: clustering, vectorial quantization, GNG, S&P100.

1 Introduction

The discovery of structures that allow the representation of data spaces has led to the creation and use of a large variety of techniques.

The most widely used methods for this purpose are those of *unsupervised competitive self-learning*, in particular neural networks, which are capable of creating topological representations by means of the distribution of a set of neurons over the input data, capturing most of the relations of the original space [1].

This is known as *vectorial quantization* and allows reducing the original data set to a smaller one, but equally representative, allowing work to be done on the vectors instead of doing it directly on the data. By means of this technique it is possible to solve, for example, the data clustering problem [2].

The most traditional *competitive learning* method is that of Kohonen's *Self-Organizing Maps* (*SOM*), which however present some limitations in practical problems because they require previous knowledge to define the structure of the network, i.e., its configuration and the number of neurons.

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 684–693, 2007.

In view of this, neural methods arise that incorporate a new philosophy in their operation: the *growth of neurons*. In these it is the network itself what determines autonomously its structure, whether it is the required *number of neurons*, the *connections* between them, or the possible eliminations of both [3].

Examples of these are the *Growing Cell Structures* (GCS) and *Growing Neural Gas* (GNG) networks

In this paper a proposal is made of a variation of the GNG algorithm, called *GNG with post-pruning* (GNG-PP), which allows eliminating and repositioning neurons so that vectorial quantization is improved. Furthermore, a clustering method is proposed whose operation is based on the topological information acquired during the training process, through the same neural method, called *Neighborhood Clustering*.

These methods will be applied to the clustering of data by means of a three-phase process. First, quantize the input space by means of a GNG network with post-pruning (GNG-PP). In a second stage, use the *Neighborhood* method to clustering the quantization vectors, and finally, associate the data with the closest vectors according to a measure of distance, identifying them with the cluster to which the related vector belongs.

To evaluate the results obtained, use was made of the *S&P100* set, belonging to the macroeconomic field, which contains the stock market variation indices of *Standard & Poor's* stock market of the 100 largest companies in the USA in terms of capital market. This data set has the peculiarity that each subject (company) is represented in a very high space dimensionality with 249 characteristics, which transforms it into an icon for evaluation. The clustering were evaluated by means of the *kappa* coefficient because the real classification of the companies was known, in this case the *Global Industry Classification Standard* (GICS).

Finally, the results are compared with those obtained by a traditional neural method (SOM), a growing one (GCS), and a non-neural one, which has been found to be one of the most efficient in the treatment of these kinds of problems.

2 Methods

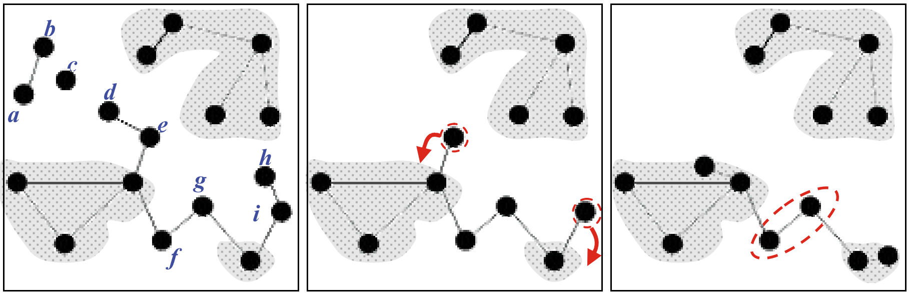
2.1 Growing Neural Gas with Post-Pruning (GNG-PP)

The GNG algorithm [4] gets a series of characteristics from other self-organizing methods (SOM [5], NG [1, 6] and GCS [7]) for quantizing the input space.

But it incorporates others like the no need to predefine a topological structure or to maintain the consistent structure of the network during the training. It also introduces the concepts of *local error* for each neuron, and *age* for each connection, allowing them to be created and eliminated at any time, giving the network greater flexibility in the representation sought [3].

Another of its characteristics is that it bases its topological preservation capacity in obtaining the *induced Delaunay triangulation* (IDT) [8], which allows the input space to be divided into regions or clusters of vectors (during its vectorial quantization process), generating neural connections *only in those areas of space where data are found*. It is precisely the generation of the IDT what allows GNG to find clusters in the data space.

However, one of the risks of working with growing networks is that an inadequate training termination criterion can be chosen, and therefore the model obtained would not truly represent the input space. An example of this could be to use very few training steps or a very high range of quantization error.



(a) (b) (c)

Fig. 1. Post-pruning process: (a) Identification of the non useful neurons. (b) Elimination of neurons and identification of new coordinates. (c) Final GNG model. In each image the shaded regions represent input data distributions.

To solve this problem it is proposed to carry out a *post-pruning* of the GNG models once the training stage has ended, with the purpose of eliminating and/or re-localizing the neurons that do not contribute to decreasing the quantization error. The general operation of the method is the following:

i) As initial information it uses the neural model (neurons and connections) obtained by GNG.

ii) The closest vector is associated with each datum by means of the calculation of a distance measure.

iii) The neurons that are not useful for minimizing the quantization error, i.e. those to which no data were associated in the previous step, are identified, and they are assigned to set V_{in}. In Figure 1.a it would be $V_{in}=\{a,b,c,d,e,f,g,h,i\}$.

iv) The neurons of V_{in} are eliminated and/or relocalized. In this step, one of three cases may occur:

- If a disconnected neuron is found, such as *c* in the example, it is *eliminated.*
- If neurons connected only to neurons belonging to V_{in} are found, they are also *eliminated* together with their connections. In the example, *a, b*, *d* and *h*.
- If neurons connected at least to a *useful* neighbor are found, they are not eliminated (in the example, *e, f, g* and *i*). Here, two cases must be distinguished:
  a) If the neuron has only one *useful* neighbor, then it will be *relocated* in a zone where it can help to decrease the quantization error, but without losing the connection (neurons highlighted in Figure 1.b). The new location is given by a differential of the position of the *useful* neuron to which it is connected.
  b) If the neuron is connected to more than one *useful* neighbor, it cannot be displaced (neurons highlighted in Figure 1.c).

The treatment process of the *non useful* neurons is done in the same order in which it was presented, with the purpose of *relocalizing* the largest possible number of neurons, eliminating first all the model's *leftover* nodes.

2.2 Clustering by Neighborhoods

Although growing methods are capable of finding the clusters in the input space, they do not provide information on *which neurons are part of each cluster*. To solve this a method is proposed that identifies the groups of vectors from the following concepts:

Direct and indirect neighbors. The former are those that have a direct connection that joins them, while the latter, in spite of not being connected directly, are related by means of direct neighbors common to them (see Figure 2.a).

Neighborhood. It is formed by the set of *direct* and *indirect* neighbors of a set of neurons. In the case of Figure 2.b there are 2 neighborhoods, *A* and *B*.

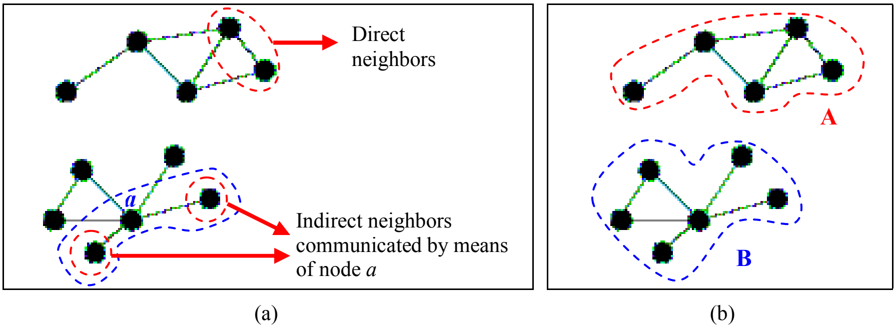


Fig. 2. Neighborhood relations: (a) Direct and indirect. (b) Neighborhoods.

The general operation of the method is the following:

i) Initialize the label index: $i=1$.

ii) Look for the first neuron $v \in A$ not associated with any cluster, where A corresponds to the structure or set of neurons of the network.

iii) Determine the direct neighbors of neuron v:

$$N_d(v) = \{ \forall i \in A \mid (v,i) \in C\} \tag{1}$$

where C is the set of connections of the structure. Figure 3 shows an example in which $N_d(v)=\{a,b,c,d,e\}$.

iv) Determine the direct neighbors of each neuron of the set $N_d(v)$ that do not belong to the same set:

$$N_d(w) = \{ \forall j \in A \mid (w,j) \in C \wedge j \notin N_d(w) \}, \ \forall w \in N_d(v) \tag{2}$$

In the example we have that $N_d(b)=\{f,g\}$, therefore the indirect neighbors of v will be: $N_i(v)=\{f,g\}$.

v) Join in set N the direct and indirect neighbors of v. In the example, it would be $N=\{a,b,c,d,e,f,g\}$.

vi) Label, in set M, all the nodes belonging to N (including neuron v), associating them to neighborhood i:

$$M(k) = i, \forall\, k \in N \cap v \tag{3}$$

In the example it would be $M(a)=\{1\}$, $M(b)=\{1\}$, ..., $M(v)=\{1\}$, as shown in Figure 3.b.

vii) Continue the revision, returning to step iii), with the following unrevised element in N. In the example, it would be to continue with $v=a$.

viii) If there are no unrevised elements in N, increase the label index: $i=i+1$.

ix) If there are unmarked neurons of A, return to step ii), otherwise the algorithm is ended. In the example, it would be to continue with neuron q (see Figure 3.b).

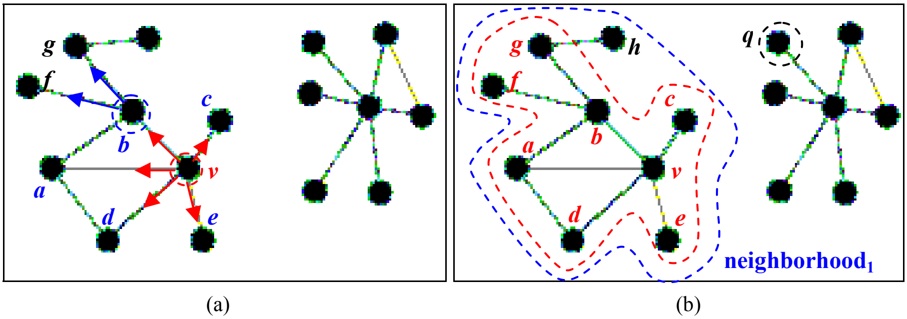


Fig. 3. Clustering by neighborhoods: (a) Direct and indirect neighbors of neuron 'v'. (b) Mark of the neurons associated with 'v' (inner segmented line), identification of '*neighborhood$_1$*' (outer segmented line) and next unmarked neuron (node q).

2.3 Clustering Strategy

A clustering in phases approach presented in [2] will be used:

Phase 1: Vectorial quantization

In this phase a vectorial quantization of the data input space is made, generating a structure formed by vectors and neighborhood relations between them reflecting their topological characteristics.

Phase 2: Clustering of quantization vectors

In this phase the clustering of the vectors obtained in the previous phase takes place. In this case it is proposed to use the *by Neighborhood* method for this purpose.

Phase 3: Association of data and vectors

Once all the model's neighborhoods have been identified, each datum is associated with the nearest vector from a distance measure (for example, *Euclidian*), identifying them with the cluster or neighborhood to which the related vector belongs.

3 Experimental Results

3.1 Data

Standard & Poor's 100 [1] (S&P100) index is one of the main of *stock market* indicators in the USA, which measures the performance of the largest 100 companies (over US$ 6 trillion) in terms of market capitalization.

For any given company, the values of the S&P index are related to the time series of the price of its own stock in a given time period.

In this work the set of data was calculated, as indicated in Inostroza-Ponta et al. [9], i.e., the experimental value y_i at time t is given by:

$$y_i(t)=\frac{P_i(t-h)-2\cdot P_i(t)+P_i(t+h)}{P_i(t-h)} \tag{4}$$

where $P_i(t)$ corresponds to the price of the stock of company i in week t, h represents the interval used to calculate the price variation of the stock (in this case it corresponds to one week), and $P_i(t - h)$ is the normalization to eliminate any influence introduced by the current stock price.

In this way the experimental set was formed by 100 registers (one per company) and 249 columns or dimensions (associated to the value y_i). In this case use was made of the S&P indices between the years 1999 and 2004.

3.2 Clustering Evaluation

The *kappa coefficient* was used to obtain a measure of the quality of the clustering obtained. This is an indicator of the agreement between the *values estimated by the model* and the *true values of the evaluation set* [10].

It was chosen to use this index because the real classification of the S&P set was known beforehand. In this case the evaluation set corresponds to the *Global Industry Classification Standard* [2] (GICS), which classifies the companies in four levels, with different subclassifications according to it (see Table 1). However, in this study only the first two levels will be considered.

Table 1. Classification of companies by level according to GICS

Level No.	Level Name	No. of subclassifications
1	Sector	10
2	Group of industries	22
3	Industries	39
4	Industrial branch	53

[1] www2.standardandpoors.com/portal/site/sp/en/us/page.topic/indices_rtv

[2] www2.standardandpoors.com/portal/site/sp/en/us/page.topic/indices_gics

3.3 Results

The clustering were made by means of the strategy of three phases, varying the vectorial quantization algorithm for *phase 2* and using a *Euclidian* distance relation in *phase 3*.

In this case neural methods were used to quantize the space: a fixed one with topological structure and predetermined shape (SOM), a growing one with rigid dimensional structure (GCS), and the proposed algorithm (GNG-PP).

In the case of the SOM, the clustering strategy was to use each neuron of the network as a group by itself [2], so two configurations were used, one of 5x2 and one of 5x4 neurons (SOM-1 and SOM-2 models, respectively), because it was attempted to obtain a sensitive clustering both at level 1 and at level 2 of the GICS classification.

In both cases it was decided to use hexagonal lattices because in them, in contrast with rectangular lattices, the neurons have not only vertical and horizontal connections, so the evolution of their neighborhood zones affects a greater number of neurons at a time, achieving greater capacity to adapted to the input space.

In the case of growing methods, the values of the training parameters were defined from their function within the corresponding algorithm [4, 7, 11]. In the case of the learning rates (ε_b and ε_n) small values were chosen with the aim of moving the neurons from their random initial positions, with some balance, in all directions. It must always be true that $\varepsilon_b >> \varepsilon_n$, because otherwise it would be the neighbors and not the winner neuron that would move faster toward the input vector, reflecting the existing topology inadequately.

With respect to the decrease in the local error rates of each neuron (α and β), their values are associated with the purpose of increasing the influence of the most recent errors in order to avoid an uncontrolled growth of the local errors.

In the case of the growth parameter λ, use was made of values associated with the capacity of each network to generate the vectors clusters by means of pruning neurons during the training. In this way, in the case of GCS the network was increased every 500 training steps, because in each elimination of leftover neurons it is possible to eliminate many others to maintain consistent the structure of growing cells.

For GNG-PP this was done only every 100 steps, trying to generate models with no more than 100 neurons, avoiding the creation of too many nodes with respect to the total data (in this case only 100 companies). For the same reason, 100 were used as maximum age for each connection (a_{max}).

As to the threshold for the elimination of neurons in GCS (η), its value was used according to a recommendation from the literature [7].

Finally, the termination criterion used for the growing methods was the number of training steps. Because of this and due to the pruning characteristics of each method, more than twice the number of steps was used to train the GCS network compared to GNG-PP, to try to generate a more robust model in terms of the number of final neurons. Table 2 shows the values used for each parameter in each neural model.

Using these training values, the clustering with each growing method were generated, finding the groups autonomously and automatically. In the case of

Table 2. Values of the training parameters for the growing methods

Method	Training	ε_b	ε_n	a_{max}	λ	α	β	η
GCS	110000	0.01	0.00010	-	100	0.5	0.0005	0.09
GNG-PP	50000	0.01	0.00005	100	500	0.05	0.0005	-

Table 3. Classification of companies by level according to GICS

Method	# Groups	# Neurons	Kappa Level 1	Kappa Level 2
GNG-PP	20	98	0.5643	0.4622
Non neural	10	-	0.5242	0.3618
SOM-2	20	20	0.5078	0.3441
GCS	7	124	0.3792	0.2347
SOM-1	10	10	0.3690	0.2349

GNG-PP there were 20 clusters, and in that of GCS there were 7, with 98 and 124 neurons, respectively. Table 3 presents a summary with the results obtained by each method considering only levels 1 and 2 of the GICS classification.

As to the GNG model without *post-pruning*, 21 groups were obtained in 110 neurons, 12 of which were not useful in vectorial quantization. Using *post-pruning*, 3 of them were eliminated and 9 were relocated in positions of the space where they contributed to decrease the *quantization* and *topological* errors [5, 12], in that way improving the characterization of the input space.

Furthermore, the results obtained by a *non neural* method presented in Inostroza-Ponta et al. [9] were added; it considers that each market stock is a node of a graph, and that each edge has a weight associated with the correlation between the stock. Thus, the method divides the graph recursively into disconnected subgraphs forming the clusters. As far as we can tell, this is the most efficient method for solving this problem.

With respect to the visualization of the results, it was decided to use the projection algorithm of the GCS method [7], because the model proposed by Fritzke does not have characteristics that restrict it exclusively to the growing cell method, so it was used without any modification.

The projection for GNG was restricted only to a bidimensional space, favoring its ease of visualization over the possible loss of topological characteristics. This means that it is possible that the distances represented in the projection may not be strictly related to the real positions of the n-dimensional space.

Figure 4 shows the projection obtained by the GNG-PP model for the S&P100 set, giving an approximate idea of what happens in the n-dimensional input space. It also shows the projection obtained by means of a classical method: *Multidimensional Scaling* (MDS) [13]. This carries out the space transformation in a *metric* and *nonmetric* way, and that would indicate if the relation between the initial proximities matrix and the distances of the new space will be *linear* or *nonlinear*.

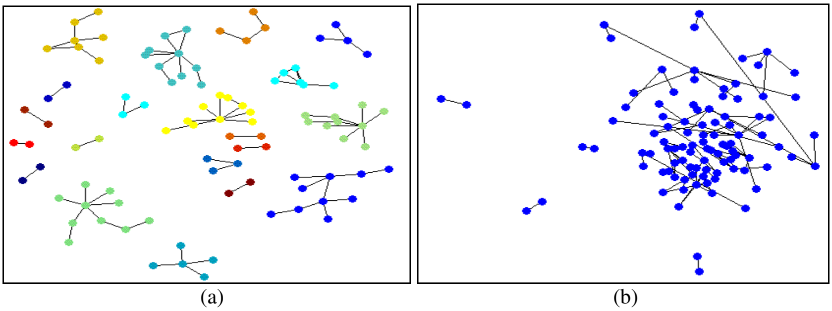


Fig. 4. Projection of the GNG-PP model (generated from the S&P100 set) obtained by: (a) the growing method. (b) MDS.

4 Discussion and Conclusions

After the clustering made of the S&P100 set, it was found that the GNG algorithm with *post-pruning* got the best kappa coefficients, both for level 1 and for level 2 of the GICS classification. This is because GNG has a series of improvements that distinguish it from the other methods used and also make it a method sensitive to the characterization of data spaces.

In the SOM models, since they have a rigid, predefined topological structure and are not capable of making prunings in their network, there are limitations in the results obtained, because they lead to the use of alternatives such as forcing each neuron to be a cluster by itself.

Therefore, a bad choice of the number of neurons would make it lose capacity for the topological representation of the space, generating very poor quality clustering. This is the case of both SOM models, whose results are incapable of improving the results obtained by GNG-PP.

With respect to the use of growing methods, one of their greatest limitations is that they are extremely sensitive to the values of their training parameters, because an incorrect choice of them can generate very poor results or processes that would use much computer time and resources to generate them.

In the case of GCS, its weak results can be due, together with the above, to the fact that at each pruning of the network useful units were eliminated in the vectorial quantization with the purpose of maintaining the cell structure consistent, losing too much topological information.

However, for GNG the influence of its parameters is attenuated, in terms of data clustering, because none of them have a direct influence in the partition of the space. On the other hand, by not depending on a rigid topological structure and allowing each neuron to have different numbers of neighbors, greater flexibility is achieved in the characterization of the original space, giving the possibility of relocating neurons in places where they can help in improving its quantization.

Also, by using *post-pruning*, it is possible to eliminate neurons which, being useless in the characterization of the input space, can generate a distortion in the number of clusters found.

In this way more robust models are constructed, in terms of the quantization of the input space, making use of most of the neurons that were incorporated in the process of growth of the network. In the case of S&P100, the results obtained by the *non-neural* method were even improved, in spite of the fact that GNG-PP bases its operation exclusively on the calculation of distances, and may achieve results as robust as this method that uses optimization techniques in the fractioning of the data set.

As to the projection achieved by GNG-PP, it turns it into an important aid in the cluster analysis, because it makes it possible to appreciate visually how the groups are distributed in the plane, keeping most of the topological relations of the n-dimensional original vector space.

The above is because the space transformation is obtained during the training of the network, and therefore reflects all the changes produced in the model until it represents the input space of the data. This is precisely what does not happen with the MDS projection method, which depends on the type of transformation used (either *metric* or *non-metric*).

Finally, it is important to mention that the evaluation has been made with a case study whose difficulty is centered in its high dimensionality. It would also be of interest to evaluate this method with benchmarks in which the number of cases is significant, such as the case of applications in the field of bioinformatics.

References

1. Martinetz, T., Schulten, K.: A neural gas network learns topology. Artificial Neural Networks, pp. 397–402. Elsevier Science Publishers, Amsterdam, Holanda (1991)
2. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE 3, 11 (2000)
3. Fritzke, B.: Growing self-organizing networks - Why? ESANN, Belgium pp. 61–72 (1996)
4. Fritzke, B.: A growing neural gas network learns topology. Advances in Neural Information Processing Systems, Cambridge, USA (1995)
5. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59–69 (1982)
6. Martinetz, T., Berkovich, S., Schulten, K.: Neural gas network for vector quantization and its application to time-series prediction. IEEE 4, 4, 218–226 (1993)
7. Fritzke, B.: Growing cell structures - a self-organizing network for unsupervised and supervised learning. Neural Networks 1441–1460 (1994)
8. Martinetz, T.: Competitive hebbian learning rule forms perfectly topology preserving maps. In: ICANN 1993, pp. 427–434 (1993)
9. Inostroza-Ponta, M., Berretta, R., Mendes, A., Moscazo, P.: An automatic graph layout procedure to visualize correlated data. In: IFIP 19th World Computer Congress, Artificial Intelligence in Theory and Practice, August 21-24, Santiago. Chile, pp. 179–188 (2006)
10. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics 33, 159–174 (1977)
11. Fritzke, B.: Some competitive learning methods. Biophysics Systems, Institute for Neural Computation, Ruhr-Universitat Bochum, Germany (1997)
12. Fritzke, B.: Kohonen feature maps and growing cell structures - A performance comparison. Advances in Neural Information Processing Systems 5, 115–122 (1993)
13. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27 (1964)