STRING MEASURE APPLIED TO STRING SELF-ORGANIZING MAPS AND NETWORKS OF EVOLUTIONARY PROCESSORS[1]

Nuria Gómez Blas, Luis F. de Mingo, Francisco Gisbert, Juan M. Garitagoitia

Abstract: This paper shows some ideas about how to incorporate a string learning stage in self-organizing algorithms. T. Kohonen and P. Somervuo have shown that self-organizing maps (SOM) are not restricted to numerical data. This paper proposes a symbolic measure that is used to implement a string self-organizing map based on SOM algorithm. Such measure between two strings is a new string. Computation over strings is performed using a priority relationship among symbols; in this case, symbolic measure is able to generate new symbols. A complementary operation is defined in order to apply such measure to DNA strands. Finally, an algorithm is proposed in order to be able to implement a string self-organizing map.

Keywords: Neural Network, Self-organizing Maps, and Control Feedback Methods.

ACM Classification Keywords: F.1.1 Models of Computation: Self-modifying machines (neural networks); F.1.2 Modes of Computation: Alternation and non-determinism.

Introduction

Most well known numeric models are Neural Networks that are able to approximate any function or classify any pattern set provided numeric information is injected into the net. Neural Nets usually have a supervised or unsupervised learning stage in order to perform desired response. Concerning symbolic information new research area has been developed, inspired by George Paun, called Membrane Systems. A step forward, in a similar Neural Network architecture, was done to obtain Networks of Evolutionary Processors (NEP), introduced by Victor Mitrana. A NEP is a set of processors connected by a graph, each processor only deals with symbolic information using rules. In short, objects in processors can evolve and pass through processors until a stable configuration is reach.

Self-Organizing maps are usually used for mapping complex, multidimensional numerical data onto a geometrical structure of lower dimensionality, like a rectangular or hexagonal two-dimensional lattice [2, 3]. The mappings are useful for visualization of data, since they reflect the similarities and vector distribution of the data in the input space. Each node in the map has a reference vector assigned to it. Its value is a weighted average of all the input vectors that are similar to it and to the reference vectors of the nodes from its topological neighbourhood. For numerical data, average and similarity are easily computed: for the average, one usually takes the arithmetical mean, and the similarity between two vectors can be defined as their inverse distance, which is most often the Euclidian one. However, for non-numerical data [4]– like symbol strings – both measures tend to be much more complicated to compute. Still, like their numerical counterparts, they rely on a distance measure. For symbol strings one can use the Levenshtein distance or feature distance.

For strings, one such measure is the Levenshtein distance [1], also known as edit distance, which is the minimum number of basic edit operations – insertions, deletions and replacements of a symbol – needed to transform one string into another. Edit operations can be given different costs, depending on the operation and the symbols involved. Such weighted Levenshtein distance can, depending on the chosen weighting, cease to be distance in the above sense of the word.

[1] Supported by projects CCG08-UAM TIC-4425-2009 and TEC2007-68065-C03-02

Another measure for quantifying how much two strings differ is feature distance [2]. Each string is assigned a collection of its substrings of a fixed length. The substrings the features are typically two or three symbols long. The feature distance is then the number of features in which two strings differ. It should be noted that this measure is not really a distance, for different strings can have a zero distance. Nevertheless, feature distance has a practical advantage over the Levenshtein by being much easier to compute.

A similarity measure is simpler than distance. Any function $\boldsymbol{S : X^2 \rightarrow R}$ can be declared similarity – the question is only if it reflects the natural relationship between data. In practice, such functions are often symmetrical and assign a higher value to two identical elements than to distinct ones, but this is not required.

String Measure

Let *V* an alphabet over a set of symbols. A string *x* of length *m* belonging to an alphabet *V* is the sequence of symbols $a_1a_2\ldots a_m$ where the symbol $a_i \in V$ for all $1 \leq i \leq m$. The set of all strings over *V* is denoted by V^*, the empty symbol is λ and the empty string is denoted by $\varepsilon = (\lambda)^*$.

Let $O{:}x \rightarrow n,\ x \in V,\ n \in N$ a mapping that establish a priority relationship among symbols belonging to *V* , $u \leq v$ iff $O(u) \leq O(v)$. Obviously $O(O(x)) = x,\ x \in V$ and $O(O^{-1}(n)) = n,\ n \in N$, and $O(\lambda) = 0,\ O^{-1}(0) = \lambda$. This mapping can be extended over an string *w* in such a way that $O(w) = O(w_i),\ w_i \in w$. Usually, such mapping *O* covers a range of integer numbers, that is, the output is $0 \leq i \leq k$, where $k = card(S),\ S \subseteq V$. It is important to note that new symbols can be generated provided that given two symbols $a, b \in V\ |O(a) - O(b)| > 1$, and there is no symbol *c* such that $O(a) < O(c) < O(b)$. That is,

$$\mathcal{O}^{-1}(k) = \begin{cases} x \in V & \text{iff } \mathcal{O}(x) = k \\ s_k & \text{i.o.c.} \end{cases}, \text{ with } k \in \mathcal{N}$$

Symbolic measure between two strings $u, v \in V^*$, denoted by $\Delta(u, v)$, with $|u| = |v| = n$ is another string defined as:

$$\Delta(u,v) = \bigcup_{i=1}^{n} \mathcal{O}^{-1}(|\mathcal{O}(u_i) - \mathcal{O}(v_i)|), \text{ where } u_i/v_i \text{ is the } i\text{-th symbol} \in u/v \quad (1)$$

For example, let $u = (abcad)$, $v = (abdac)$, and *O* the index of such symbol in the Latin alphabet, that is, $O(a) = 1$, $O(b) = 2$, $O(c) = 3$, $O(d) = 4$ then $\Delta(u, v) = \lambda\lambda a\lambda a$. If $u = (jonh)$, $v = (mary)$ then $\Delta(u, v) = s_3njs_{11}$, two new symbols s_3, s_{11} are generated (that correspond to $s_3 = c$ and $s_{11} = k$, usually such correspondence is unknown). A numeric value *D* can be define over a string *w*:

$$\mathcal{D}(w) = \sqrt{\sum_{i=0}^{|w|} \mathcal{O}(w_i)^2}, w_i \in w \quad (2)$$

It is clear to proof that:

$D(\Delta(u, v)) = D(\Delta(v, u))$, $D(\Delta(u, u)) = 0$, $D(\Delta(u, \varepsilon)) = D(u)$ and $D(\Delta(u, w)) \leq D(\Delta(u, v)) + D(\Delta(v, w))$.

Mappings *O/D* also define a priority relationship among string in V^* is such a way that

$$u \le v \text{ iff } \sqrt{\sum_{i=1}^{n=|u|} \mathcal{O}(u_i)^2} \le \sqrt{\sum_{i=1}^{n=|v|} \mathcal{O}(v_i)^2}$$

$$u \le v \text{ iff } \mathcal{D}(u) \le \mathcal{D}(v)$$

In short, symbolic measure between two string *u, v* is obtained using *Δ(u, v)*, see equation (2), and numeric measure is obtained using *D(Δ(u, v))*, see equation (1). Let *x, y* ∈ *S* ⊆ *V* two symbols belonging to alphabet, two symbols are complementary, denoted by $(x, y)^{-}$, iff *Δ(x, y) = x* or *Δ(x, y) = y*. Such property can be extended over string, let *u, v* ∈ S^* ⊆ V^* , two strings are complementary, denoted by $(u, v)^{-}$, iff *Δ(u, v) = u* or *Δ(u, v) = v*.

Theorem 1. - Let *u, v* ∈ S^* , *u* and *Δ(u, v)* are complementary iff *O(ui) >= O(vi)* for all *1 ≤ i ≤ n*.

Proof.

$$\Delta(u, v) = \bigcup_{i=1}^{n} \mathcal{O}^{-1}(|\mathcal{O}(u_i) - \mathcal{O}(v_i)|)$$

Hence:

$$\Delta(u, \Delta(u, v)) = \bigcup_{i=1}^{n} \mathcal{O}^{-1}(|\mathcal{O}(u_i) - \mathcal{O}(\Delta(u_i, v_i))|) =$$

$$= \bigcup_{i=1}^{n} \mathcal{O}^{-1}(|\mathcal{O}(u_i) - \mathcal{O}(\mathcal{O}^{-1}(|\mathcal{O}(u_i) - \mathcal{O}(v_i)|)|) =$$

$$= \bigcup_{i=1}^{n} \mathcal{O}^{-1}(|\mathcal{O}(u_i) - (|\mathcal{O}(u_i) - \mathcal{O}(v_i)|)|) =$$

$$= \bigcup_{i=1}^{n} \mathcal{O}^{-1}(\mathcal{O}(v_i)) = v$$

□

Two strings *u, v* ∈ S^* are Watson-Crick complementary (WC complementary), denoted by $(u, v)^{-WC}$, iff $(u, v)^{-}$ for all *1 ≤ i ≤ |u|*.

Theorem 2. - Let *u, v* ∈ S^* , if $(u, v)^{-}$ then $(u, v)^{-WC}$.

Such duality in symbolic/numeric measures, see equations (1 and 2), is a good mechanism in order to implement algorithms on biological DNA strands [5, 6]. Like DNA or amino acid sequences, which are often, subject to research in computational molecular biology. There, a different measure – similarity – is usually used. It takes into account mutability of symbols, which is determined through complex observations on many biologically close sequences. To process such sequences with neural networks, it is preferable to use a measure, which is well empirically founded.

Strings with different lengths

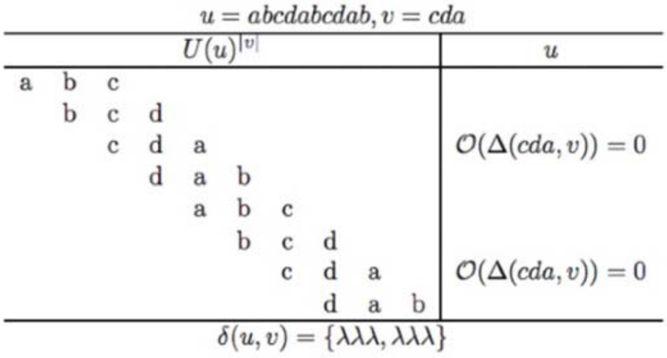
Given two strings *u, v*, such that *|u| = n ≥ |v| = m*, and *U (u)* the set of all substring *w* ⊆ *u* such that,

$$U(u)^m = \{w^{(j)} || w^{(j)}| = m, w = w_1 \cdots w_m, w_i = u_k, i = k + j\} \forall\, 0 \le j \le |u| - m$$

String measure between *u, v*, denoted by *δ(u, v)*, is

$$\delta(u, v) = \{\Delta(s, v) | s \in U(u)^{|v|}, \mathcal{O}(\Delta(s, v)) \le min_{x \in U(u)^{|v|}}\{\mathcal{O}(\Delta(x, v))\}\}$$

In this case, measure *δ* is a set of strings with the lower distance (see table below). Such distance can be read as the set of matching strings with lower distance. This *δ* can be used to identify cutting points (index *j*) over a DNA string when applying a restriction enzyme, from a biological point of view.



$u = abcdabcdab, v = cda$

$U(u)^{\|v\|}$										u
a	b	c								
	b	c	d							
		c	d	a						$\mathcal{O}(\Delta(cda, v)) = 0$
			d	a	b					
				a	b	c				
					b	c	d			
						c	d	a		$\mathcal{O}(\Delta(cda, v)) = 0$
							d	a	b	

$\delta(u, v) = \{\lambda\lambda\lambda, \lambda\lambda\lambda\}$

Let $|u| = |v|$, it is clear that $\delta(u, v) = \Delta(u, v)$ since $U(u) = u$.

Future Work

Some results, in literature, that could be checked with this new measure can be: for an example application of the string SOM, Igor Fisher generated a set of 500 strings by introducing noise to 8 English words: always, certainly, deepest, excited, meaning, remains, safety, and touch, and initialized a quadratic map with the Sammon projection of a random sample from the set [1]. Another real world example is the mapping produced from 320 hemoglobin alpha and beta chain sequences of different species [2]. SOM and LVQ algorithms for symbol strings have been introduced by [5, 6] and applied to isolated word recognition, for the construction of an optimal pronunciation dictionary for a given speech recognizer.

Artificial Neural Networks (ANN) and Networks of Evolutionary Processors (NEP) [9, 10] can be considered as the present and the future of connectionist models. Both of them are based on the idea of simple processors that communicate in order to achieve a global objective. But there are two important facts that must be taken into account:

- ANN are numeric models while NEP are symbolic ones.
- There exists a learning algorithm that control the ANN behavior in order to achieve a desired result while NEP do not incorporate any kind of learning paradigm.

Some ideas of ANN can be translated into NEP architecture since ANNs are considered, in the literature, a good model to solve non-conventional problems. Following this point of view some kind of learning can be added to a NEP to obtain a more general model than simple NEP. Among all the neural networks architectures unsupervised neural networks, cal led Self Organizing Maps (SOM), are the most suitable.

Conclusions

In some applications, like molecular biology, a similarity measure is more natural than distance and is preferred in comparing protein sequences. It is possible that self-organizing neural networks can successfully process such data. It can therefore be concluded that similarity-based neural networks are a promising tool for processing and analyzing non-metric data. This paper has proposed a string measure that can be applied to self-organizing maps or networks of evolutionary processors with the possibility of new symbols generation. Watson-Crick complementary concept was defined using such measure.

Acknowledgements

This work is supported by projects CCG08-UAM TIC-4425-2009 and TEC2007-68065-C03-02.

Bibliography

[1] LEVENSHTEIN L.I, Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics–Doklady 10, (1966) 707–710.

[2] TEUVO KOHONEN, Self-Organization and Associative Memory. Springer, Berlin Heidelberg, (1988).

[3] TEUVO KOHONEN, SOMERVUO P, Self-Organizing Maps of Symbol Strings with Application to Speech Recognition, (1997).

[4] TEUVO KOHONEN, SOMERVUO P, Self-organizing maps of symbol strings, Neurocomputing 21 (1998) 19–30.

[5] MARIA SANCHEZ, NURIA GOMEZ, LUIS MINGO, DNA Simulation of Genetic Algorithms: Fitness Function, International Journal on Information Theories and Applications, 14 (3). ISSN 1310-0513 (2007) 211–217.

[6] NURIA GOMEZ, EUGENIO SANTOS, MIGUEL ANGEL DIAZ, Symbolic Learning (Clustering) over DNA Strings, WSEAS Transactions on Information Science and Applications. 3 (4), ISSN: 1709-0832 (2007) 617–624.

[7] IGOR FISCHER, ANDREAS ZELL, String averages and self-organizing maps for strings, Proceeding of the ICSC Symposia on Neural Computation (NC’2000) May 23-26, 2000 in Berlin, Germany, (2000), 208–215.

[8] IGOR FISCHER, Similarity-based neural networks for applications in computational molecular biology, Lecture notes in computer science, 2779, ISSN 0302-9743, (2003) 208–218.

[9] JUAN CASTELLANOS, FLORIN MANEA, LUIS F. MINGO, VICTOR MITRANA, Accepting Networks of Splicing Processors with Filtered Connections, MCU (2007) 218–229.

[10] FLORIN MANEA, VICTOR MITRANA, Al l NP-problems can be solved in polynomial time by accepting hybrid networks of evolutionary processors of constant size, Inf. Process. Lett. 103(3), (2007), 112–118.

Authors' Information

Nuria Gómez Blas – Dept. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain; e-mail: ngomez@eui.upm.es

Luis Fernando de Mingo – Dept. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain;
e-mail: lfmingo@eui.upm.es

Francisco Gisbert – Dept. Lenguajes, Sistemas Informáticos e Ingeniería del Software, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain; e-mail: fgisbert@fi.upm.es

Juan M. Garitagoitia – Dept. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. De Valencia km. 7, 28031 Madrid, Spain; e-mail: jmgmartin@eui.upm.es