Binary-based similarity measures for categorical data and their application in Self-Organizing Maps

Fernando Lourenço [1], Victor Lobo [1], and Fernando Bação [1]

1 – Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa

g2001167@isegi.unl.pt, vlobo@isegi.unl.pt, bacao@isegi.unl.pt

Abstract

In exploratory data analysis of high dimensional data one Eof the main tasks is the formation of a simplified overview of data sets. Clustering and projection are among the examples of useful methods to achieve this task. However there are several types of data where the use of this measure is not adequate, such as the categorical data. In this paper we will review some of the most common binary-based similarity measures that can be applied to this type of data. These measures are evaluated empirically using the Self-Organizing Maps (SOM) algorithm. The SOM algorithm performs a non-linear mapping from a high-dimensional data space to a low-dimensional space, typically two-dimensional, aiming to preserve the topological relations of the data. A well known data set of Animals from (Ritter and Kohonen 1989) is used.

We tested two different approaches to compute the best matching unit (BMU). Working with binary data in the SOM algorithm poses issues concerning the update method for the neurons and the internal modeling of the data, which we have to be aware. The non-metric properties of some measures must also have our attention.

Some similarity measures produced maps that were very like to each other. Exploring these maps, we may find that the clustering obtained using the SOM provides different perspectives over the data.

1 Introduction

In exploratory data analysis of high dimensional data one of the main tasks is the formation of a simplified, usually visual, overview of data sets. This can be achieved through simplified description or summaries, which should provide the possibility of discovery or identification of features or patterns of most relevance (Lehman 1988) referred in (Murteira 1993). Clustering and projection are among the examples of useful methods to achieve this task.

Classical clustering algorithms produce a grouping of the data according to a chosen criterion, e.g. (Kaski, Nikkila et al. 2000). Projection methods, on the other hand, represent the data points in a lower dimensional space in such a way that the clusters and the metric relations of the data items are preserved as faithfully as possible (Kohonen 2001). In this field most algorithms use similarity measures based on Euclidean distance. However there are several types of data where the use of this measure is not adequate. Such is the case when using categorical data since, generally, there is no known ordering between the feature values, e.g. (Andritsos 2002) . In this paper we will review some of the most common binary-based similarity measures that can be applied to this type of data.

In this study a Self-Organizing Maps (SOM) algorithm is used to evaluate empirically a number of binary-based similarity measures. The SOM algorithm resembles vector quantization algorithms (Kohonen 2001). The major distinction between the SOM and other vector quantization techniques is that in the former the neurons are organized on a regular grid and along with the selected neuron also its neighbors are updated, yielding an ordering of the neurons. The algorithm performs a non-linear mapping from a high-dimensional data space to a low-dimensional space, typically two-dimensional, rectangular grid (Ultsch, Guimarães et al. 1993; Verleysen 1997; Kohonen 2001). This allows the presentation of multidimensional data in two dimensions, assisting the visual exploration of data, e.g. (Kaski, Nikkila et al. 2000). As the SOM compresses data while preserving the most important topological relationships it may be thought as producing some kind of abstractions (Kohonen 2001).

Here the application of several similarity measures to a practical case is presented. Working with binary data poses issues concerning the update method for the neurons and the internal modeling of the data, which have to be dealt with (Lobo 2002).

1.1 Data Types and their Measures

Consider that **x** and **y** are data objects, each one of them with the form: $\mathbf{x} = (x_1, x_2, ..., x_k)$, and $\mathbf{y} = (y_1, y_2, ..., y_k)$, where k is the dimensionality, while each x_i, and y_i, for $1 \leq i \leq k$, is a feature of the corresponding object. The data objects may also be referred as vectors or patterns and the features may also stand for attributes or components. For the purpose of this paper a pattern is conceived as an ordered set of features, and we will be dealing only with fixed cardinality ordered sets (patterns).

A comprehensive categorisation of the different types of features found in most data sets provides a helpful means for identifying the differences among data elements. We present a classification based on two schemes: the *Domain Size* and the *Measurement Scale* (Andritsos 2002) referring (Anderberg 1973).

1.1.1 Classification Based on the Domain Size and Measurement Scale

The classification based on the domain size distinguishes data features based on the size of their domain, that is, the number of distinct values the features may assume. We have the following classes:

A feature is **continuous** if between any two values of the feature an infinite number of values exist. Examples of such features could be the temperature and the color or sound intensity.

A feature is **discrete** if its values can be put into a one-to one correspondence with a subset (or all) of the positive integers. Examples could be the number of children in a family or the serial numbers of books. The class of binary features consists of features whose domain includes exactly two discrete values. They comprise a special case of discrete features, and we present as examples the Yes/No responses to a poll or the Male/Female gender entries of a database.

On the other hand, the classification based on the measurement scale considers the following classes: nominal, ordinal, interval, and ratio. Suppose we have a feature *i* and two tuples ***x*** and ***y***, with values x_i and y_i for this feature, respectively.

A **nominal** scale simply distinguishes between categories. This means that we can only say if $x_i = y_i$ or $x_i \neq y_i$. Nominal-scaled feature values cannot be totally ordered. They are just a generalization of binary features, with a domain of more than two discrete values (e.g. the place of birth and the set of movies).

An **ordinal** scale involves nominal-scaled features with the additional feature that their values can be totally ordered, but differences among the scale points cannot be quantified. Hence, on top of $x_i = y_i$ or $x_i \neq y_i$, we can assert if $x_i < y_i$ or $x_i > y_i$ (e.g. the medals won by athletes).

In this paper we will only discuss the data types that have discrete valued features, with finite domain and which are nominal. From here on we will refer these data type as categorical. Indeed as a matter of convenience we may use the term data types to refer to feature data types.

1.2 Measures of Distance and Similarity for patterns

When talking about patterns, the distance and similarity concepts are in a way reciprocal. In fact we often use the term distance to convey the idea of dissimilarity. Through the use of a association or similarity measure we try to quantify the likeness between patterns (Van Rijsbergen 1979). So when comparing patterns, it is very useful if they are represented in a space that has a metric. This is a property of any set of elements characterized by the distance function between all pairs of elements, denoted $d(\mathbf{x},\mathbf{y})$ for elements **x** and **y** e.g. (Kohonen 2001). For the choice of the distance, the following conditions must hold:

$d(\mathbf{x},\mathbf{y}) \geq 0$, where equality holds if and only if $\mathbf{x}=\mathbf{y}$,

$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$, symmetry;

$d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})$, triangle inequality.

An example of distance which we shall use as reference, is the Euclidean distance $d_E(\mathbf{x},\mathbf{y})$ in a rectangular coordinate system:

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

Naturally, this distance should only be applicable to real-valued patterns.

1.2.1 Categorical Data

In this section, we consider objects whose features are categorical. This poses additional problems concerning similarity measures mainly because generally it is not possible to define a metric space with an implicit distance function.

A categorical measurement can be characterized as a rule for arranging observations into equivalence classes, so that observations falling into the same set are thought of as qualitatively the same (Hays 1981). Each observation is placed in one and only one class, making the classes mutually exclusive and exhaustive. Among all other measurement scales, such as ordinal or interval measurement, categorical measurement is the most basic. In fact, classifying observations into qualitatively different classes is the basis for all quantitative studies (Hu 1998).

Categorical values cannot be ordered in a single way and, therefore, clustering of such data is a challenge. We summarize the characteristics of such data in the following list (Andritsos 2002):

- Categorical Data have no single ordering: there are several ways in which they can be ordered, but there is no single one that is more semantically sensible than others;
- Categorical Data can be visualized depending on a specific ordering;
- Categorical Data define no a priori structure to work with;
- Categorical Data can be mapped onto unique numbers and, as a consequence, Euclidean distance could be used to prescribe their proximities, with questionable consequences though.

One sensitive point is the last one: (Guha, Rastogi et al. 2000) give an example why this entails several dangers. They also show that traditional clustering algorithms that use distance between points for clustering are not appropriate for binary or categorical features. (Leisch, Weingessel et al. 1998) and (Mustapha, Fouad et al. 2000) share the same point of view, referring that Euclidean distance is not adapted to binary data, it is often much more interesting to use an appropriate similarity index (e.g. Hamming and Tanimoto).

2 Similarity measures

We go on now with a brief overview of some common similarity measures.

2.1 Simple Matching

The simple matching is the simplest of all similarity measures (Van Rijsbergen 1979):

$\mathrm{S_M}(A,B)=n(A \cap B)$

where n(X) is the number of elements in set X. In this basic form it does not take into account the sizes of each set.

2.2 Tanimioto

The **Tanimoto** similarity measure has its origin in the comparison of sets e.g. (Kohonen 2001), is also referred as Jaccard e.g. (Van Rijsbergen 1979). If A and B are two unordered sets then the similarity can be measured as the ratio of their common elements to the number of all different elements:

$$\mathrm{S_T}(A,B)=\frac{n(A \cap B)}{n(A \cup B)}=\frac{n(A \cap B)}{n(A)+n(B)-n(A \cap B)},$$

2.3 Hamming

The **Hamming** distance was originally defined for binary codes (Hamming 1950) but can be applicable to any ordered sets of equal length. The dissimilarity measure may be based on the number of different symbols between two sets, e.g.:

x=(p,a,t,t,e,r,n)

y=(w,e,s,t,e,r,n), so $d_H(\mathbf{x},\mathbf{y})=3$.

This measure can be defined by the total mismatches of the corresponding feature categories of two objects. Formally,

$$d_H(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

where

$$\delta(x_i, y_i) = \begin{cases} 0 & if \quad (x_i = y_i) \\ 1 & if \quad (x_i \neq y_i) \end{cases}$$

If the features are binary the Hamming distance, the City block distance, a scaled simple matching measure and the squared Euclidean distance, all become equivalent.

2.4 Others

Most of the similarity measures are defined for binary valued features. In order make use of them, we have to convert our categorical features into binary features. A straightforward method is to spread out the categorical features creating one binary feature for each domain value e.g.(Bação 2002). This is better understood with a simple example. Taking the case of a categorical object **x**=(x_1, x_2), with two features, e.g $x_1 \in$ {A,B} and x_2 ∈ {C,D}, we will get a binary object $\mathbf{x}' = (x'_1, x'_2, x'_3, x'_4)$ where x'_1=1 if x_1=A and x'_1=0 if x_1≠A, x'_2=1 if x_2=B and x'_2=0 if x_2≠B, and so forth with the other features.

With this arrangement of the data we can now fill the contingency table (see Table 1 - Contigency table values) comparing the feature values for each pair of objects **x**, and **y**, where:

a = number of times x_i=1 and y_i=1

b = number of times x_i=0 and y_i=1

c = number of times x_i=1 and y_i=0

d = number of times x_i=0 and y_i=0.

		Object x		
		1	0	sum
Object y	1	***a***	***b***	a+b
	0	***c***	***d***	c+d
	sum	a+c	b+d	a+b+c+d

Table 1 - Contigency table values

Thus our initial categorical patterns may be handled using methods for binary patterns.

We will use several similarity measures. It is worth to note the effort to normalise the measures, e.g. taking into account the size of the sets and the weighting of the matches, thus being generally expressed in a form of coefficients (Van Rijsbergen 1979). We remember also the effect of the coding scheme on the measures. In this subject, we call a binary feature *symmetric* if both values it can assume have equal relevance. The measure of similarity based on this type of features is called *invariant* if does not change with how we code the events e.g. simple matching - Table 2, e.g. (Bação 2002). We also have the *asymmetric* binary features, where one value is more relevant then the other, like in a disease survey, and in this case the measure is *non-invariant* e.g. Jaccard coefficient - Table 3. In this case, it is recommended that we code these events as "1".

The similarity coefficients may be grouped into two classes regarding of how they deal with the negative co-occurrence (value *d* in Table 1). One that considers the negative co-occurrence (Table 2) and the other that does not consider this co-occurrence (Table 3) (Meyer 2002). The ranges shown are taken for a particular data set; for other specific combinations, it can be seen that the denominator becomes zero, which raises difficulties in the application of some measures.

Coefficients	Equation	Range
Simple Matching (Sokal and Michener 1958)	$\frac{a+d}{a+b+c+d}$	[0,1]
Russel and Rao (Russel and Rao 1940)	$\frac{a}{a+b+c+d}$	[0,1]
Rogers and Tanimoto (Rogers and Tanimoto 1960)	$\frac{a+d}{a+d+2(b+c)}$	[0,1]
Hamann (Hamann 1961)	$\frac{(a+d)-(b+c)}{a+b+c+d}$	[-1,1]
Ochiai II (Ochiai 1957)	$\frac{ad}{\sqrt{(a+d)(a+c)(d+b)(d+c)}}$	[0,1]
Sokal and Sneath (Sokal and Sneath 1963)	$\frac{2(a+d)}{2(a+d)+b+c}$	[0,1]

Table 2 – Similarity coefficients considering negative co-occurrence. All references in (Meyer 2002)

Coefficients	Equation	Range
Jaccard (Jaccard 1901)	$\frac{a}{a+b+c}$	[0,1]
Anderberg (Anderberg 1973)	$\frac{a}{a+2(b+c)}$	[0,1]
Czekanowsky / Sorensen-Dice (Dice 1945)	$\frac{2a}{2a+b+c}$	[0,1]
Kulczynski I (Kulczynski 1927)	$\frac{a}{b+c}$	[0,+∞]
Kulczynski II (Kulczynski 1927)	$\frac{a}{2}\left(\frac{1}{a+b}+\frac{1}{a+c}\right)$	[0,1]
Ochiai (Ochiai 1957)	$\frac{a}{\sqrt{(a+b)(a+c)}}$	[0,1]

Table 3 - Similarity coefficients not considering negative co-occurrence. All references in (Meyer 2002)

For an evaluation of several similarity coefficients refer to (Meyer 2002) where the coefficients of Jaccard, Sorensen-Dice, Anderberg and Ochiai gave similar results due to the fact that all of them exclude negative co-occurrences. This was also observed for the Simple Matching, Rogers and Tanimoto, and Ochiai II probably due to the fact of all including the negative co-occurrences (Meyer 2002). Russel and Rao presented results very different from the others because it excludes the negative co-occurrences in the numerator and includes it in the denominator - a reason for not recommending it. Indeed, the negative co-occurrence does not mean necessarily any resemblance or similarity. It is also interesting to note that the Ochiai II measure does not have the symmetry property. Special care has to be taken for particular cases where a divide by zero may occur like in the Kulczynsky measure. The metric and non-metric properties of some of these measures were also studied in (Zhang and Srihari 2003). For the measures we are studying, they found based on experimental results, that the discriminative power was best for the Rogers and Tanimoto, and worse for the Russel and Rao measure.

2.5 Which Measure to Use

Any distance measure could be used for clustering. However, if we have no a priori knowledge concerning the data, there is no reason to prefer any measure over the Euclidean metric e.g. (Sammon 1969). However, for binary data the usual Euclidean distance can be replaced by binary similarity measures that take into account possible asymmetries and therefore provide a different point of view for looking at the data e.g. (Mustapha, Fouad et al. 2000), (Leisch, Weingessel et al. 1998). Each similarity measure has its own properties and generally gives different perspectives of the data turning the matter of choice not trivial (Meyer 2002).

2.6 Benchmark example

In order to test and evaluate the effect of the similarity measures in the SOM, we can use the Animals - Data Set, as referred in (Andreas 2003), is a prominent example taken from (Ritter and Kohonen 1989) and also used in (Kohonen 2001). Being widely used and discussed as a reference data set in numerous papers on related topics, this data set allows easy evaluation and comparison of the achieved results. It consists of 16 artificially designed patterns in R^{13}, describing 16 animals by arbitrarily chosen features, see Table 4 - Animals Data Set.

	Feature	Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
Is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 4 - Animals Data Set

In the Table 5 - Some similarity measures values, we provide the values of the similarity measures between some pairs of animals.

account d	*Dove/Hen*	*Dove/Tiger*	*Tiger/Cow*	*Dove/Cow*
Simple Matching	0,92	0,31	0,77	0,38
Russel and Rao	0,23	0,00	0,23	0,00
Rogers and Tanimoto	0,86	0,18	0,63	0,24
Hamman	0,85	-0,38	0,54	-0,23
Ochiai II	0,47	0,00	0,39	0,00
Sokal	0,96	0,47	0,87	0,56
Don't account d				
Jaccard	0,75	0,00	0,50	0,00
Anderberg	0,60	0,00	0,33	0,00
Kulczynski I	3,00	0,00	1,00	0,00
Kulczynski II	0,88	0,00	0,68	0,00
Sorensen-Dice	0,86	0,00	0,67	0,00
Ochiai	0,87	0,00	0,67	0,00

Table 5 - Some similarity measures values

(Andreas 2003) provides a good overview of this data set. He observes that there are two rather distinct groups of animals described by their patterns, namely birds and mammals. Of these, the birds form the strongest cluster, with all of them having feathers and only 2 legs, separating them from all the other animals. Furthermore, all but the eagles are small animals, a feature that they share with none but the cat among the mammals. The birds themselves can be subdivided into several - less distinct - groups, depending on whether they like to hunt, run, fly and/or swim. Note, however, that many overlaps occur in these substructures. On the other hand, the mammals are strongly characterized by having 4 legs and hair instead of feathers.

The names of the animals (labels) are not part of the features, leading to two pairs of identical input patterns, owl and hawk as well as horse and zebra. The names of the various animals are enclosed as extra features in the original paper (Ritter and Kohonen 1989).

(Su and Chang 2000) group these 16 animals into three classes (birds, carnivores, and herbivores). They found that the 2-D projection of the animal data set (using Sammon's method) is linearly separable.

(Rauber 1999) basically identifies 2 clusters of animals, namely birds and mammals, strongly separated by their number of legs as well as the fact whether they have feathers or fur. Other analysis using SOM can be seen in (Merkl and Rauber 1997).

3 Clustering Data with SOM

The Self-Organizing Maps (SOM) algorithm, also called the *Kohonen network*, the *self-organizing feature map* (SOFM) or the *topological map*, is an unsupervised neural network algorithm developed by Teuvo Kohonen (Kohonen 2001) that provides us with two useful operations in exploratory data analysis. These are clustering, reducing the amount of data into representative categories, and projection (non-linear), aiding in the exploration of proximity relations in the patterns. The SOM algorithm has some resemblance with vector quantization algorithms. The great distinction from other vector quantization techniques is that the neurons are organized on a regular grid and along with the selected neuron also its neighbors are updated. As a result SOM performs an ordering of the neurons (Kaski, Nikkila et al. 2000).

A comprehensive introduction and description of the SOM algorithm can be seen in (Kohonen 2001). Here we will just make a brief presentation. The SOM consists of an array or lattice of elements called neurons or units, usually arranged in a low dimensionality grid (1D or 2D), the map, for ease of visualization. The lattice type may have several forms like rectangular, hexagonal or even irregular. Associated with each unit there is a pattern called model, codebook or reference, having the same dimensionality of the input patterns. The image of an input pattern **x** on the SOM array is the unit $\mathbf{m}_c$ that best matches with **x**. This winning unit is also called best matching unit (BMU). Using a dissimilarity measure $d(\mathbf{x},\mathbf{m}_i)$, this unit has the index:

$$\text{c=arg } \min_i \{d(\mathbf{x},\mathbf{m}_i)\}$$

Next we will briefly describe the original incremental SOM algorithm. In the learning phase de codebook patterns of are updated iteratively following the rule:

$$\mathbf{m}_i(\text{t+1})= \mathbf{m}_i(\text{t})+h_{ci}(\text{t})\,[\mathbf{x}(\text{t})\text{-}\,\mathbf{m}_i(\text{t})]$$

In this process $h_{ci}(\text{t})$ represents a kernel neighborhood function over the lattice, generally wide in the beginning and decreasing in time, defining which units and how much will they be affected by each input pattern.

A variant of the basic SOM is the batch algorithm where the whole training set (input patterns) is gone through at once and only after this the map is updated with the net effect of all the patterns. The updating is done replacing the codebook pattern with a weighted average over the input patterns:

$$\mathbf{m}_i\,(\text{t+1})= \frac{\sum_{j=1}^{n} h_{ic(j)}(t)\mathbf{x}_j}{\sum_{j=1}^{n} h_{ic(j)}(t)}$$

where *c(j)* is the BMU of input pattern $\mathbf{x}_\text{j}$, $h_{i,c(j)}$ the neighborhood function, and *n* is the number of input pattern. The weighting factors are the neighborhood function values.

The codebook patterns are thus updated to follow the input patterns in an ordered way. As a result of this algorithm the codebook patterns of close by units should be similar thus preserving most of the topology relations of the input patterns on the map. We may picture this like an elastic network becoming smoothly fitted to the input pattern distribution. This way the density of the codebook patterns approximate the density of the input patterns.

With due care, we may state some resemblance between the projection properties of the SOM algorithm and of the principal component analysis (PCA). The main difference is the nonlinear projection of the SOM that can be a great advantage in many cases (Verleysen 1997). As has been referred by (Kaski and Kohonen 1998) and (Kohonen 2001) in SOM local factors impact locally (and they will have greater importance) in the projection which doesn't happen in PCA.

In order to interpret the SOM map, it is convenient to calibrate it, thus locating the images of the input patterns on the map. Labeling some map units with this information is helpful for this task. The unified distance matrix methods, U-matrix (Ultsch, Guimarães et al. 1993), are a graphical representation of the SOM map structure and also provide us very useful assistance. The simplest of these methods is to compute for each codebook pattern the mean of the dissimilarity measures to its neighbors. Plotting this data on a 2D map, using a gray or color scheme, we can visualize a landscape with walls and valleys. The walls separate different classes, and patterns in the same valleys are similar (Ultsch, Guimarães et al. 1993).

3.1 Problems in using SOM for Binary Data

The SOM as conceived should live the input patterns space, i.e. the codebook patterns should lie in the space of the input patterns. The original SOM algorithm was defined for real valued patterns. However, when using binary input patterns and as consequence of computation, the codebook patterns will assume non-binary (i.e., real) values. To keep applying the binary similarity measures we have to envisage some way to convert the real valued pattern to a binary one in order to compute the best matching unit (BMU).

A more difficult issue is how to update the codebook patterns if they are binary-value. This is an interesting matter and some alternatives could be toggling some features in a round robin fashion or in probabilistic manner (Lobo 2002). In our present work we do not tackle this problem.

3.2 Proposed SOM architecture

In our experiments we used two approaches to the BMU problem: the “hard” logic and the dot product. In the former method, codebook patterns, despite having real values, are interpreted as binary logic values, using a threshold *t*. For example, if $x_i \geq t$ then we assume it as 1 and 0 otherwise. In the dot product method e.g. (Kohonen 2001), we interpret the real valued vector as a vector of probabilities, where each component value is the probability of that component being one, somehow similar to (Leisch, Weingessel et al. 1998). In this way the features of the patterns (input and codebook) are interpreted as a probability of being one. Doing so, the features of the codebook patterns as the values a, b, c, and d (see Table 1 - Contigency table values) used to compute the dissimilarity measures may be real valued.

3.3 Experimental Results

The following maps represented using U-matrix were trained using the several similarity measures. After training an evaluation of the best matching units (BMUS) using the same measures was done. The codebook patterns were initialized linearly along the two greatest eigenvectors of the covariance matrix of the training data (i.e. using PCA), following practical advice (Sammon 1969), (Vesanto 2000). The similarity measure used to evaluate the BMUS is shown in the footer of the figures.

The simulations were done in MatLab 6.1 using Som Toolbox 5 with the adequate adaptations. The batch train method is used as it is much faster to calculate in Matlab than the normal sequential algorithm, and the results are typically just as good or even better (HUT 2003). Except where explicitly stated, the values and parameters used in the procedures were the default ones, e.g. sheet with hexagonal lattice of 7 x 3 units, constant radius one on gaussian neighborhood, train length of 14 iterations. Practical advice of (Kohonen, Hynninen et al. 1996) was followed, like the form of the array (hexagonal lattice with oblong form of 7 by 3 map units). Useful guidelines for interpreting the map are also found in (Kaski and Kohonen 1998).

3.3.1 Using Euclidean Distance

This is the original application of the SOM where the codebook features are real-valued and are interpreted as so.

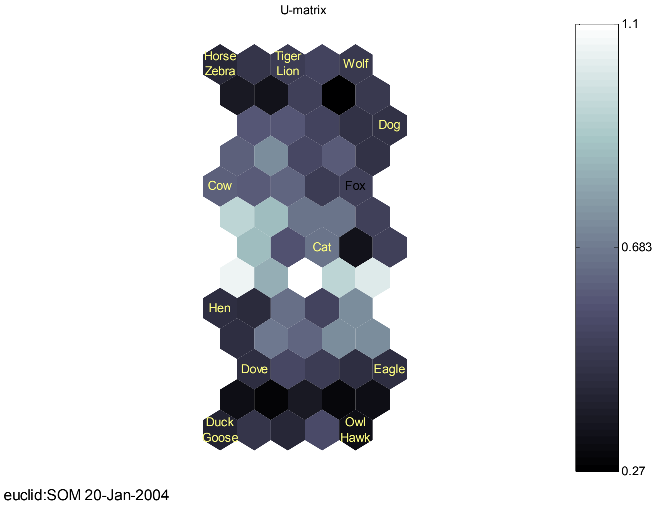
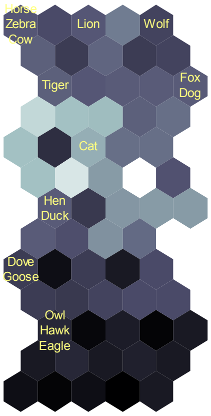


Figure 1 - Euclidean measure.

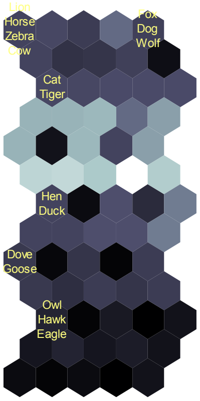
3.3.2 Using “hard” logic

In the following maps the BMUS were computed using other similarity measures. In these cases we used a threshold *t*=0.5.



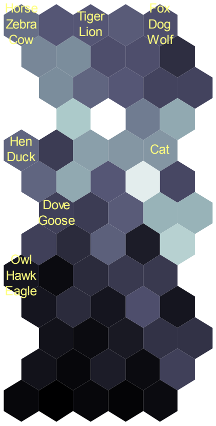
smatch:SOM 20-Jan-2004

Figure 2 - Simple match measure.



rao:SOM 20-Jan-2004

Figure 3 - Rao measure.



ochiai2:SOM 20-Jan-2004

Figure 4 - Ochiai II measure

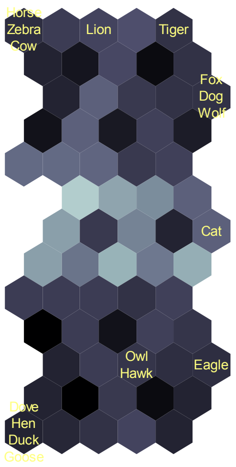
It can be seen that the data in the lower part of the map is not evenly projected and it looks as the map shrinks towards the top left corner. An explanation for this is the fact when we convert several different but similar real valued codebook patterns to binary valued it is much likely that they become equal. Now the SOM sees just one of these codebook patterns, because using the “hard” logic approach makes them indistinguishable. The shrinking to the top left corner is a consequence of Matlab preferring a lower index for the BMU. This shrinking effect will somehow stop when a “wall” is near. The figures presented resume the maps that we very alike:

Map	Similar to
Simple Match	Rogers-Tanimoto, Hamann, Sokal
Ochiai II	Jaccard, Anderberg, Sorensen-Dice, Kulczynski I, Kulczynski II, Ochiai

3.3.3 Using Dot Product Approach

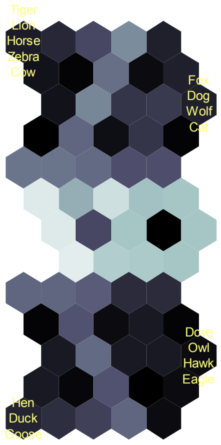
The figures presented resume the maps that we very alike:

Map	Similar to	Account negative co-occurrence (d)
Simple Match	Rogers-Tanimoto, Hamann, Sokal	Yes
Rao	Ochiai II	Yes
Jaccard	Anderberg, Sorensen-Dice, Kulczynski I	No
Kulczynski II	Ochiai	No



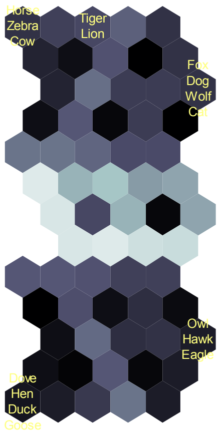
smatch:SOM 20-Jan-2004

Figure 5 – Simple Match measure



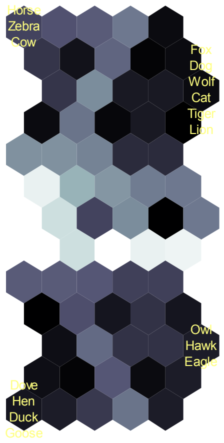
rao:SOM 20-Jan-2004

Figure 6 – Russel and Rao measure



jaccard:SOM 20-Jan-2004

Figure 7 – Jaccard measure



kulcz2:SOM 20-Jan-2004

Figure 8 - Kulczynsky II measure

Comment: –conf. jan 2003

In most of these maps we can easily distinguish at least four great clusters.

3.4 Analysis of Results

The topographic error gives the percentage of vectors for which the BMU and the second BMU are not neighboring map units (Vesanto 2000). This gives an idea of how well the map preserves the topology in the input patterns. Remember that, in theory, two neighbor codebook patterns should be mapped in two neighbor units. For reference, the topographic error obtained with the Euclidean measure was zero. It is interesting to note the great topographic errors of some maps. The non-metric properties of some measures may be one explanation for this fact.

Measure / Topographic Error (%)	Logic	Dot product	Comment for Dot Product
Simple Matching	6.25	6.25	Complex, just 2 great clusters, and 8 classes
Russel and Rao	0	31.25	4 great clusters: big & mammal, big & bird
Rogers and Tanimoto	6.25	6.25	Complex, just 2 great clusters, and 8 classes
Hamann	6.25	6.25	Complex, just 2 great clusters, and 8 classes
Ochiai II	0	31.25	4 great clusters: big & mammal, big & bird
Sokal and Sneath	0	6.25	Complex, just 2 great clusters, and 8 classes
Jaccard	0	25	5 clusters: big & mammal, carnivore & bird
Anderberg	0	25	5 clusters: big & mammal, carnivore & bird
Czekanowsky / Sorensen-Dice	0	25	5 clusters: big & mammal, carnivore & bird
Kulczynski I	0	25	5 clusters: big & mammal, carnivore & bird
Kulczynski II	0	25	4 great clusters: carnivore & mammal, carnivore & bird
Ochiai	0	25	4 great clusters: carnivore & mammal, carnivore & bird

4 Conclusions

Based on a preliminary visual analysis of the resulting maps, a number of conclusions can be draw. The first conclusion is that all measures used in this study separate well the mammals from the birds. The second conclusion concerns the differences that can be observed in the map that uses Euclidean distance. This map is quite different from the maps that result from the application of other measures. The U-matrix in the figure 1 (Euclidean distance) shows a uniform distribution of the patterns across the map, being difficult to find clear-cut clusters. In the case of the other measures used (figures 2, 3, 4, 5, 6, 7 and 8) it is easier to detect additional cluster structures on the map.

A detailed analysis reveals the existence of other relief forms, which allow further distinctions. In fact, the maps produced using the dot product method present a clearly defined cluster structure. In the case of the "hard" logic method, results are less encouraging. In this case there are areas of the SOM which are not used at all in the classification process, indicating a poor representation of the underlying space.

When using the dot product method, binary-based similarity measures improve the representation of the most relevant clusters. In this aspect the "hard" logic method is not so good but yields smaller topographic errors. In terms of the results of the dot product method, the most relevant distinction observed between the different measures used is that some emphasize the size of the animals and others the type of feeding.

In the context of SOM it is clear that the range of variation allowed by Euclidean distance cannot be matched by binary-based measures. This means that at this time it is not realistic to use binary-based similarity measures to produce “fine-resolution” clustering, although most of the measures used revealed the ability to distinguish major clusters.

In this work we showed that binary-based similarity measures might provide a different insight into data, effectively revealing interesting patterns and relations in the data.

Keywords: Exploratory Data Analysis, Self-Organizing Maps (SOM), Similarity Measures, Categorical Data.

5 References

Anderberg, M. (1973). Cluster analysis for applications, Academic Press.

Andreas, R. (2003). Cluster Visualization in Unsupervised Neural Networks. **2003**.

Andritsos, P. (2002). Data Clustering Techniques. Toronto, University of Toronto, Dep. of Computer Science.

Bação, F. (2002). Ferramentas de Exploração. Data Mining Geo-Espacial - Pub. Apoio UNIGIS. Lisboa**:** 98.

Dice, L. R. (1945). "Measures of the amount of ecological association between species." Ecology(26): 297-302.

Guha, S., R. Rastogi, et al. (2000). "ROCK: A Robust Clustering Algorithm for Categorical Attributes." Information Systems **25**(5): 345-366.

Hamann, V. (1961). "Merkmalbestand und verwandtschaft sbeziehungen der farinosae. Ein Beitragzum System der Monokotyledonen." Willdenowia **2**: 639-768.

Hamming, R. (1950). Tech. J. 29, Bell Syst.**:** 147.

Hays, W. L. (1981). Statistics. New York, CBS College Publishing.

Hu, X. (1998). General Processing Tree Models. **2003**.

HUT (2003). SOM Toolbox for Matlab, Helsinki University of Technology

Laboratory of Computer and Information Science. **2004**.

Jaccard, P. (1901). "Étude comparative de la distribuition florale dans une portion des Alpes et de Jura." Bulletin de la Societé Voudoise des Sciences Naturelles(37): 547-579.

Kaski, S. and T. Kohonen (1998). Methods for Interpreting a Self-Organized Map in Data Analysis. ESANN'98, 6th European Symposium on Artificial Neural Neural Networks, Brussels, Belgium.

Kaski, S., J. Nikkila, et al. (2000). Methods for Exploratory Cluster Analysis. SSGRR 2000 Int. Conf. on Advances in Infrastruture for Electronic Business, Science, and Education on the Internet, L'Aquila.

Kohonen, T. (2001). Self-Organizing Maps. Berlin, Springer-Verlag.

Kohonen, T., J. Hynninen, et al. (1996). SOM_PAK: The Self-Organizing Map Program Package. Espoo, Finland, Helsinki University of Technology, Laboratory of Computer and Information Science.

Kulczynski, S. (1927). "Classe des Sciences Mathématiques et Naturelles, ,." Bulletin International de l'Acadamie Polonaise des Sciences et des Lettres **Série B (Sciences Naturelles)**(Supplement II): 57-203.

Lehman, E. (1988). Statistics: An Overview. New York, Wiley.

Leisch, F., A. Weingessel, et al. (1998). Competitive Learning for Binary Valued Data. International Conference on Artificial Neural Networks, Skoevde, Sweeden, Springer.

Lobo, V. (2002). Ship noise classification: a contribution to prototype based classifier design. Departamento de Informatica. Lisbon, Universidade Nova de Lisboa.

Merkl, D. and A. Rauber (1997). On the similarity of eagles, hawks, and cows - Visualization of similarity in self-organizing maps. Int'l Workshop Fuzzy-Neuro-Systems'97, Soest, Germany.

Meyer, A. d. S. (2002). Comparison of Similarity Coefficients Used in Cluster Analysis with Dominant Markers Data. Escola Superior de Agricultura Luiz de Queiroz. Piracicaba, Universidade de São Paulo**:** 106.

Murteira, B. (1993). Análise Exploratoria de Dados Estatistica Descritiva. Lisboa, McGraw-Hill.

Mustapha, L., B. Fouad, et al. (2000). Topological Map for Binary Data. ESANN '2000 proceedings - European Symposium on Artificial Neural Networks, Bruges, Belgium.

Ochiai, A. (1957). "Zoogeographic studies on the soleoid fishes found in Japan and its neighbouring regions." Bulletin of the Japanese Societyfor Fish Science **22**: 526-530.

Rauber, A. (1999). LabelSOM: On the Labeling of Self-Organizing Maps. Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), Washington, DC,.

Ritter, H. and T. Kohonen (1989). Self-Organizing Semantic Maps. Biological Cybernetics**:** 241-254.

Rogers, J. S. and T. T. Tanimoto (1960). "A computer program for classifying plants." Science(132): 1115-1118.

Russel, P. F. and T. R. Rao (1940). "On habitat and association of species of anopheline larvae in south-eastern Madras." Journal of Malaria India Institute(3): 153-178.

Sammon, J. W., Jr (1969). "A Nonlinear Mapping for Data Structure Analysis." IEEE Transactions on Computers **C-18**(5).

Sokal, R. R. and C. D. Michener (1958). "A statistical method for evaluating systematic relationships." Bulletin of the Society of University of Kansas **38**: 1409-1438.

Sokal, R. R. and P. H. Sneath (1963). Principles of numeric taxonimy. San Francisco, W.H. Freeman.

Su, M.-C. and H.-T. Chang (2000). Fast Self-Organizing Feature Map Algorithm. IEEE TRANSACTIONS ON NEURAL NETWORKS. **11:** 721.

Ultsch, A., G. Guimarães, et al. (1993). Knowledge Extraction from Artificial Neural Networks and Applications. Transputer-Anwender-Treffen, Aachen, Springer Verlag.

Van Rijsbergen, C. (1979). Information Retrieval, Butterworth-Heinemann.

Verleysen, M. (1997). Feedforward models. Handbook of Neural Computation. E. Fiesler and R. Beale, IOP Publishing and Oxford University Press.

Vesanto, J. e. a. (2000). SOM Toolbox for Matlab 5. Espoo, Helsinki University of Technology**:** 59.

Zhang, B. and S. Srihari (2003). Binary Vector Dissimilarity Measures for Handwriting Identification. SPIE, Document Recognition and Retrieval X, Santa Clara, California.