A Taxonomic Associative Memory Based on Neural Computation

M. Gyllenberg* and T. Koski[1]

Department of Applied Mathematics, University of Turku, FIN-20500 Turku, Finland
[1]Department of Mathematics, University of Technology, S-97187 Luleå, Sweden

Received: 1 November 1994; revised 19 February 1995

A single layer feed forward neural network for associating organisms with binary features to the most typical organisms in a given numerical classification is presented. The network implements an associative memory, which has stored maximal predictivity. The network also represents a neural model for the classification as well as a neurocomputer for numerical identification. The rationale in probabilistic numerical identification of bacteria is explained. After a learning phase based on backpropagation for minimization of the crossentropy between the most typical organisms and the network outputs the memory associates by maximizing a kind of 'probability of belonging' to a taxon.

For numerical experiments we have modified the MATLAB™ ***Neural Networks Toolbox.*** *We consider in particular the expansion and rejection of identification properties of the memory, which are potentially useful in cumulative or continuous classification and identification.*

Introduction

Applications of neural networks in bacterial identification have been recently presented by several authors including Bungay & Bungay (1990), Chun *et al.* (1993) and Rataj & Schindler (1991). Since numerical classification and identification of bacteria mathematically amounts to clustering of vectors, as perhaps first pointed out by Sneath (1957*a,b*), the advantages of and the general rationale for neural computing in bacterial taxonomy are the same as in other similar problems of pattern recognition, see Lippman (1988). The main benefits usually brought forth are that neural nets learn from data and are able to implement complicated decision regions without assumptions about an explicit form for the statistical distribution for the patterns. In addition, neural nets are robust against test errors in input data, a property of significance in bacteriology, see Sneath (1974). All these facts and certain additional circumstances, explained by Bungay & Bungay (1990), Chun *et al.* (1993), Rataj & Schindler (1991) and below, make neural nets an attractive alternative to the well-established methods of bacterial identification developed by among others H.G. Gyllenberg (1963, 1965, 1976), Dybowski & Franklin (1968), Hill (1974), Lapage *et al.* (1970, 1973), Sneath (1979*a,b*) and Willcox *et al.* (1980).

The purpose of this paper is to present a new method of probabilistic bacterial identification. It is based on the idea of *associative memory,* see Kohonen (1989). Roughly speaking an associative memory is a device for storing pairs of given input and output vectors such that when presented with one of the given inputs the memory recalls the corresponding output. The memory recalls the appropriate output vector even when presented with a distorted version of a stored input vector. If an input vector is too different from all the stored input vectors, the memory may be unable to associate any of the stored output vectors to the input. To overcome this problem the memory should be able to expand its storage. For introductions to various aspects of associative memories we refer to Kanerva (1990).

In our discussion of the methodology of identification of bacteria the stored input vectors are binary vectors representing given bacteria. The corresponding output gives the class membership of the input in terms of the centroid or *hypothetical mean organism* (HMO) of the class. After this initial *learning* phase, the memory can identify new bacteria by associating to a feature vector one of the stored HMOs. As the computational implementation of our associative memory we shall use a *single layer artificial neural network.* Learning amounts to choosing the weights in the network. This is done by minimizing the *crossentropy* between the stored output vectors (HMOs) and the corresponding output vectors of the network. This choice is motivated theoretically by the fact that minimization of crossentropy is the only consistent inference procedure to account for new information given in the form of expected values, see Shore & Johnson (1980, 1981). More importantly from the point of view of bacterial taxonomy, we show that crossentropy minimization captures some of the most fundamental principles of probabilistic numerical identification.

The idea of basing classification and clustering on crossentropy minimization is not new. For instance the underlying idea in the paper by of Shore & Gray (1982) is similar to ours. Crossentropy minimization has been considered in the context of neural networks by several authors, we mention only Richard & Lippman (1991).

*corresponding author
Tel: +358 21 633 6567
Fax: +358 21 633 6595
Email: matsgyl@utu.fi

0226-304X/95/070061+06 $03.00/0

BINARY Vol.7 © 1995 BioLine

An associative network for hypothetical mean organisms

Classification and Hypothetical Mean Organisms

The attributes used in most taxonomic systems for bacteria are coded with binary states. Each bacterium is therefore characterized by a *feature vector* $\mathbf{x} = (x_1, \ldots, x_d)$ where each component x_i, $i = 1, \ldots, d$ takes on the value 0 or 1. The dimension d of the binary feature vector is the number of attributes. A *classification* of a set X of feature vectors is a partition $\{c_j\}_{j=1}^{k}$ of X into pairwise disjoint classes c_j. Here k is the number of classes.

The information about the classification can be compressed into a $d \times k$ *reference matrix* $\Theta = \{\theta_{ij}\}$, where θ_{ij} if the relative frequency of the positive (i.e. = 1) state for the ith attribute in the jth class. Obviously $0 \leq \theta_{ij} \leq 1$.

The *hypothetical mean organism* (HMO) $a_j = (a_{1j}, a_{2j}, \ldots, a_{dj})$ of the class c_j is defined [see Gower (1974), Sneath (1979*b*)] by

$$a_{ij} = \begin{cases} 0, \text{ if } \leq \theta_{ij} < 1/2 \\ 1, \text{ if } 1/2 < \theta_{ij} \leq 1 \end{cases} \qquad (1)$$

If $\theta_{ij} = 1/2$ the binary value of a_{ij} can be chosen arbitrarily.

The HMO a_j is the best representative of the feature vectors assigned to c_j in the sense that the HMO a_j makes the maximal number of correct predictions about the members of the taxon c_j, see Gower (1974), and is the *most typical element* in c_j in terms of modelling by maximum entropy, see M. Gyllenberg & Koski (1994*a*).

Let X be a set of feature vectors with n members $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ classified as described above and let for each $\mathbf{x}$ in X, $t(\mathbf{x})$ denote the HMO of the class to which $\mathbf{x}$ belongs. The set

$$\chi^n = \left\{\mathbf{x}^{(p)}, t\left(\mathbf{x}^{(p)}\right)\right\}_{p=1}^{n}$$

then fully describes the classification. It can be interpreted as a set of stored input-output pairs in an associative memory as described in the introduction. χ^n is called the *training set*. The vectors $t^{(p)} = t(\mathbf{x}^{(p)})$ are also called *labels*. By *supervised training* the network is made to learn and store the information represented by the classes $\{c_j\}_{j=1}^{k}$. In the present case this is done by building a neural model of the classification. In the next subsections the architecture of the network and the learning rule will be presented.

Implementation of the Associative Network

A single layer feed forward neural network is simply a rule depending on a set of parameters, which associates to each input vector $\mathbf{x} = (x_1, \ldots x_d)$ an output vector $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}), \ldots, u_d(\mathbf{x})]$. Choosing the standard so called logsigmoid transfer function this rule takes the form

$$u_i(\mathbf{x}) = \frac{e^{w_{i0} + \sum_{l=1}^{d} w_{il} x_l}}{1 + e^{w_{i0} + \sum_{l=1}^{d} w_{il} x_l}}, \quad i = 1, \ldots, d \qquad (2)$$

In the jargon of neural networks u_i is called the output of the ith neuron, w_{il} the *weight*, stored in the neuron, for the connection of the lth input coordinate to the ith neuron and w_{i0} is the corresponding *threshold* (bias). Notice that we had to choose d parallel output neurons, since the labels (HMOs) have d components. It is convenient to collect the weights and thresholds into a matrix $\mathbf{W} = \{w_{il}\}_{i=1,l=0}^{d,d}$

For extensive accounts of the notions used above we refer to any text on neural nets, for instance Haykin (1994), or to the brief but useful introduction by Boddy *et al.* (1990).

Learning by Minimizing Crossentropy

The process of learning means that the parameters in the matrix $\mathbf{W}$ are adjusted so that the network's output $\mathbf{u}(\mathbf{x}^{(p)})$ resembles as closely as possible the label $t^{(p)}$ for all $p = 1, \ldots, n$. Since the outputs are strictly between zero and one, the final labels are obtained from the network's output by the rounding off rule as in (1).

As the measure of closeness we choose the crossentropy defined by

$$C_n(\mathbf{W}) := \sum_{p=1}^{n} \sum_{i=1}^{d} C\left[t_i^{(p)}, u_i(\mathbf{x}^{(p)})\right] \qquad (3)$$

where $C(t, u) := -[(1-t)\log(1-u) + t \log u]$.

Using (2) and taking the partial derivative of $C_n(\mathbf{W})$ with respect to each of w_{il} and setting these derivatives equal to zero one obtains

$$T \cdot X^* = U(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}; \mathbf{W}) \cdot X^* \qquad (4)$$

for $l \neq 0$ and

$$\mathbf{t} = U(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}; \mathbf{W}) \qquad (5)$$

for $l = 0$, where we have used the matrix notation

$$T = \begin{pmatrix} t_1^{(1)} & \ldots & t_1^{(n)} \\ t_2^{(1)} & \ldots & t_2^{(n)} \\ \vdots & & \\ t_d^{(1)} & \ldots & t_d^{(n)} \end{pmatrix}, X = \begin{pmatrix} x_1^{(1)} & \ldots & x_1^{(n)} \\ x_2^{(1)} & \ldots & x_2^{(n)} \\ \vdots & & \\ x_d^{(1)} & \ldots & x_d^{(n)} \end{pmatrix}, \mathbf{t} = \begin{pmatrix} \sum_{p=1}^{n} t_1^{(p)} \\ \sum_{p=1}^{n} t_2^{(p)} \\ \vdots \\ \sum_{p=1}^{n} t_d^{(p)} \end{pmatrix}$$

and where $U(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}; \mathbf{W})$ denotes the $d \times 1$ -vector with $\sum_{p=1}^{n} u_i(\mathbf{x}^{(p)})$ as entries. * denotes transposition of a matrix.

A necessary condition for $C_n(\mathbf{W})$ to attain its minimum is that $\mathbf{W}$ satisfies the system (4), (5) of matrix equations.

The matrix $T \cdot X^*$ is, up to standardization with $1/n$, the *correlation matrix memory* of the early theory of association memories, as discussed by Kohonen (1989). Thus the network measures the correlations between the training vectors and their HMOs. The idea of correlating the feature vectors with the HMOs in bacterial identification

appears (in a slightly different form) already in the paper by H.G. Gyllenberg & Niemelä (1975). Correlation is, of course, a standard idea of matching in numerical taxonomy, see Pankhurst (1991).

The Association Rule

Once the network has been trained it can be used to identify new items as follows: To each input vector the memory associates the output of the network with each component rounded off to 0 or 1 as in (1). If the associated vector is one of the HMOs, then the vector is identified as belonging to the class corresponding to that HMO. If the associated vector is not one of the HMOs, the input vector is *rejected,* that is, it is left unidentified.

It is possible to incorporate a rejection threshold in the sense of H.G. Gyllenberg & Niemelä (1975) [see also Willcox *et al.* (1980)] by allowing identification with the ith class if the associated vector differs from the HMO a_i up to some prescribed number of bits. This however requires a two-layer network.

Training the network by backpropagation

As pointed out above, supervised learning amounts to finding the parameters in the matrix $\mathbf{W}$ such that the crossentropy (3) is minimized. Training the network by backpropagation can thus be viewed as an algorithm for solving the equations (4) and (5) for $\mathbf{W}$.

Computationally we implemented backpropagation for minimization of crossentropy in the environment provided by the *MATLAB™ Neural Networks Toolbox* manual by Demuth & Beale (1992). In addition, the datahandling and matrix manipulation commands of MATLAB™ are well suited for expansions of memory.

The updating algorithm for adjusting the weights and thresholds is, just as in the standard backpropagation,

$$w_{il}(r+1) = w_{il}(r) - \mu\,(\partial/\partial w_{il})\,C_n(\mathbf{W}) \qquad (6)$$

where $(\partial/\partial w_{il})\,C_n(\mathbf{W})$ is the partial derivative of the crossentropy with respect to the weight w_{il} evaluated at the values of the weights and thresholds at *training epoch* number r. The positive parameter μ is the *learning rate.*

A numerical experiment

To investigate the performance of the associative memory defined above we trained a single layer feedforward network by minimization of crossentropy using the training set χ^{12} from Gower (1974) (see right).

$$\chi^{12} = \begin{pmatrix} (0011111011), a_1 \\ (1101111110), a_2 \\ (0000000000), a_3 \\ (0011101011), a_1 \\ (0000001101), a_4 \\ (0011110011), a_1 \\ (1101111100), a_2 \\ (1101101000), a_2 \\ (0000000101), a_4 \\ (1101110000), a_2 \\ (0011111001), a_1 \\ (1101110100), a_2 \end{pmatrix}$$

Here each horizontal row is a feature vector $\mathbf{x}^{(p)}$ with $p = 1,\ldots,12$ running from the top of the array down. We have $d = 10$ attributes and the labels (that is, the HMOs) are

a_1 = (0011111011),

a_2 = (1101111100),

a_3 = (0000000000),

a_4 = (0000000101).

In fact, this labelling is the result of an unsupervised clustering obtained by Gower (1974) on basis of $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(12)}$ above by computationally minimizing

$$\frac{1}{12}\sum_{p=1}^{12}\min_{1\le j\le 4} d_H(\mathbf{x}^{(p)}, a_j)$$

where

$$d_H\left(\mathbf{x}^{(p)}, a_j\right) = \sum_{i=1}^{10}\left|\mathbf{x}_i^{(p)} - a_{ij}\right| \qquad (7)$$

is the *Hamming* metric between feature vectors, a standard measure of distance between binary vectors.

After 5000 training epochs using the backpropagation (6) with the learning rate $\mu = 0.9$ we obtained the following weight matrix with the entries w_{il}, where i indicates rows and l columns. The entries are rounded off, for reason of space, to two decimals.

$$V_1 = \begin{pmatrix} 3.32 & 3.6 & -1.59 & 1.32 & 1.06 & 0.51 & 0.24 & 0.50 & -0.34 & -2.60 \\ 2.90 & 3.50 & -1.89 & 1.48 & 1.48 & 0.47 & 0.20 & 0.67 & -0.30 & -2.82 \\ -1.81 & -2.49 & 3.60 & 1.21 & 1.34 & 1.10 & 0.54 & -2.85 & 1.56 & 1.73 \\ 1.66 & 1.36 & 1.74 & 3.29 & 2.57 & 2.15 & 1.07 & -1.11 & 1.83 & -0.75 \\ 2.09 & 1.23 & 2.12 & 2.94 & 3.01 & 1.73 & 0.72 & -1.02 & 1.60 & -0.55 \\ 1.55 & 1.59 & 2.19 & 2.69 & 3.21 & 1.97 & 0.90 & -1.09 & 1.31 & -0.64 \\ 1.57 & 1.73 & 1.99 & 3.26 & 2.76 & 1.81 & 0.78 & -1.25 & 1.28 & -0.37 \\ 4.41 & 4.17 & -3.62 & 0.42 & -0.07 & -0.14 & 0.36 & 6.26 & -1.27 & 2.34 \\ -2.72 & -1.83 & 3.60 & 1.55 & 1.07 & 1.13 & 0.74 & -2.79 & 1.59 & 1.41 \\ -2.82 & -2.02 & 2.88 & -0.08 & -0.37 & 0.21 & 0.82 & 2.59 & 0.67 & 6.12 \end{pmatrix}$$

For the thresholds w_{i0} we found the values given in the column vector

$$V_0 = (-4.41\ -4.40\ -4.57\ -4.38\ -4.39\ -4.38\ -4.39\ -4.04\ -4.54\ -4.16)^*$$

Let $\mathbf{W}_1 = (V_0\ V_1)$. It can be checked that storing of the training data in $\mathbf{W}_1$ the memory correctly associates the vectors in the training set to their HMOs.

Next we tried the association rule for

$\mathbf{x}^{(13)}$ = (0000000001).

Using $\mathbf{W}_1$ above it could easily be calculated, actually without the aid of any software specialized to neural networks, that the memory outputs $\mathbf{x}^{(13)}$ itself, which is not one of the HMOs $a_1,\ldots,a_4$. The feature vector $\mathbf{x}^{(13)}$ is therefore rejected.

It is in fact very reasonable to leave $\mathbf{x}^{(13)}$ unidentified. To see why, let us note that

$d_H(\mathbf{x}^{(13)}, a_3) = d_H(\mathbf{x}^{(13)}, a_4) = 1$,

where and a_3 and a_4 are the HMOs closest to $\mathbf{x}^{(13)}$ in the sense of the Hamming metric (7). In other words, $\mathbf{x}^{(13)}$ lies precisely *in between* these two HMOs, since $d_H(a_3, a_4) = 2$, that is, the Hamming metric is halved. Hence a tie would occur in identification by means of the Hamming

metric.

We can, however, easily expand the memory to resolve the tie. Let us for example augment χ^{12} simultaneously with both $(\mathbf{x}^{(13)}, a_3)$ and $(\mathbf{x}^{(14)}, a_2)$ and denote this data set by χ^{14}. Running the backpropagation algorithm this time with $\mathbf{W}_1$ as initial value and with the same learning rate gives after 5000 epochs of training a new $\mathbf{W}$-matrix which we denote by $\mathbf{W}_2$.

Using the new weights we consider the vector

$$\bar{\mathbf{x}}^t = (1110000110).$$

Note that $\bar{\mathbf{x}}^t$ is the binary complement of the vector $\mathbf{x}^t$, which best predicts the feature vectors $\{\mathbf{x}^{(p)}\}_{p=1}^{14}$ treated as a single group. In fact, $d_H(\bar{\mathbf{x}}^t\ a_3) = 5$ is the smallest Hamming distance to the HMOs stored in the memory. Calculating the neural net's output using the weights obtained for χ^{14}, we get that $\bar{\mathbf{x}}^t$ is associated (identified) with a_2. Clearly $d_H(\bar{\mathbf{x}}^t, a_2) = 6$. But a straightforward computation shows that the crossentropy between a_2 and $\mathbf{x}^{(14)}$ using $\mathbf{W}_2$ is = 1.15, whereas the rest of the crossentropies are larger than 20.0.

The discussion above shows that the memories trained here are not implementations of the Hamming metric, which was initially used for establishing χ^{12}. Instead the memory makes an association that minimizes crossentropy, at least locally, in some sphere around a feature vector.

The vector $\mathbf{x}^t$ is on the other hand associated by the same memory to a_1, which is the HMO closest to $\mathbf{x}^t$ both in the sense of the Hamming metric and of the crossentropy that assumes the value 1.52.

Picking 1000 binary vectors of length d at random we obtained a set of vectors that were on the average three bits away from the HMOs. Processing these through the memory with the weights $\mathbf{W}_2$ gave us on average the distance of one bit from the HMOs. In fact, 465 were strictly identified and 207 were associated with a vector one bit away from the HMOs.

However, we found in this sample the curious phenomenon of 11 binary vectors that were associated with vectors that lie, in the sense of Hamming distance, *further away from* the HMOs in the storage than the feature vector itself.

Most likely difficulties of this type are due to the training process converging to a local minimum over some special Hamming spheres. In particular, the existence of a group consisting of one element (= a_3) in the initial training set is also probably a source of trouble. Or the phenomenon could indicate some hidden characteristics of the data set with regard to the imposed crossentropies.

Of course, when $d = 10$ there are only $2^{10} = 1024$ different feature vectors and the training set is relatively small, too. Baum (1988) gave estimates for the complexity of the network needed to represent an arbitrary map on the set of feature vectors with a given number of feature vectors by a neural network with one hidden layer. In the example above we are initially dealing with 12 feature vectors by a net with more than a hundred weights, which is more than enough. On the other hand, there is a need for computerized numerical identification methods with small, or slowly growing sets of items, as argued by M. Gyllenberg *et al.* (1993). In Pankhurst (1991) this is called the cycle of *continuous classification and identification*.

The other way round, there are rules of thumb indicating the number of training vectors, needed for generalization, as a function of the number of of weights in the network. These would indicate that we need all the 1024 feature vectors in the training set, which makes no sense. As the binary data in the example above are structured (by maximal predictivity), the rules referred to above are probably too pessimistic.

More extensive empirical studies of the associative memory discussed in this paper as well as analyses of its mathematical properties are in progress for a conclusive evaluation of the memory's performance.

Interpretation of crossentropy

In this section we shall interpret crossentropy both in information theoretic and probabilistic terms and discuss its relevance for identification of bacteria. Finally we discuss crossentropy in terms of the learning process itself.

Crossentropy and Information Content

According to Pankhurst (1991) and Sneath (1995) maximization of the information content is an important aspect of taxonomy. There is an intimate relation between crossentropy and information content that goes back at least as far as Lewis (1959). An exact definition of the suggestive term 'information content' is beyond the scope of this presentation but the main implications of its relationship to crossentropy for our associative memory can easily be explained. For a mathematically rigorous discussion of information content in connection with numerical taxonomy we refer the reader to M. Gyllenberg *et al.* (1994).

M. Gyllenberg & Koski (1994*b*) showed that a large value of the crossentropy means that by changing $u_i^{(p)}$ to become more like $t_i^{(p)}$ a lot of information is gained. We may therefore say that the crossentropy minimizing network is trained to produce outputs that *as closely as possible approximate the information content of the stored HMOs*. Hence any feature vector, whose identification is rejected by the memory, contains new information to be added to the storage.

Probability of Belonging

In view of (2) we can interpret $u_i(\mathbf{x}^{(p)})$ as the conditional probability of $t_i^{(p)} = 1$ given $\mathbf{x}^{(p)}$. This is essentially the probabilistic interpretation of network output by Bridle (1990). In the context of our network for identification of bacteria $u_i(\mathbf{x}^{(p)})$ is the probability that the given feature vector $\mathbf{x}^{(p)}$ to be identified belongs to a class whose HMO has a 1 as its ith component.

Probabilistic ideas of the above type have since the appearance of the paper by Dybowski & Franklin (1968) played a prominent role in numerical identification of

bacteria. H.G. Gyllenberg & Niemelä (1975) introduced the notion of *probability of belonging* to a class and used as identification criterion that an unknown item should be placed into the group that maximizes the probability of belonging. M. Gyllenberg & Koski (1994*b*) showed that minimizing crossentropy is equivalent to maximizing the probability of belonging. Our method is therefore a version of probabilistic identification of bacteria.

Statistical Sufficiency

In 1922 R. A. Fisher introduced the notion of *sufficient statistic*, see Andersen (1991). Roughly speaking, a sufficient statistic is a function of the observations in a sample that in some sense summarizes all the information in the sample and thus makes precise knowledge of the individual observations irrelevant. In the associative memory under consideration the pairs of feature vectors with the corresponding labels in the training set correspond to the 'observations'. If the training sequence is very long, the data stored in it may be computationally difficult to handle. It would therefore be very desirable to find a sufficient statistic independent of the length of the training sequence.

M. Gyllenberg & Koski (1994*b*) proved that the correlation matrix memory $T \cdot X^*$ and $\mathbf{t}$ together form a *sufficient statistic* for W. This means that $T \cdot X^*$ and $\mathbf{t}$ compress the training data without losing any information. Note that the dimension of the sufficient statistic is fixed: it does not depend on the number of vectors in the training set. This is important for the expansion capabilities of the associative memory.

Bayesian Learning and Networks

We shall now interpret the results of the preceding subsection in terms of learning. As explained above, when training the associative memory the feature vectors in the training set are compressed in the correlation matrix memory $T \cdot X^*$ and in $\mathbf{t}$. It is through these quantities that each *individual* feature vector influences the learning process. Moreover, when computing the correlation matrix memory the associative network also takes into account the possible interactions between different attributes in the training set.

On the other hand, the relative frequency θ_j is, for a given classification, the maximum likelihood estimate of the corresponding marginal probability of the *multivariate Bernoulli distribution* attached to the group c_j, see Duda & Hart (1973). The maximum likelihood estimate is also a sufficient statistic and as such compresses the data optimally and hence eliminates the individual feature vector in supervised Bayesian learning without any loss of information. This summarizes the difference between supervised Bayesian learning and supervised training of our associative memory.

Concluding remarks

The taxonomic associative memory introduced in the present paper is well founded on some of the basic principles of bacterial identification. After the learning phase the memory associates new inputs by minimizing the crossentropy between the item and the most typical organisms stored in the memory and does this using the simple parallel neural computations. These computations involve measuring the correlations between the training vectors and their hypothetical mean organisms. The association can be interpreted as maximizing the 'probability of belonging to a class' and also as maximizing the information content of the classification. In addition, the memory has a natural capability of extending the memory and various possibilities for rejection of identification, as is desired in continuous classification and identification. Thus this associative memory of most typical organisms has the basic properties required in computer-assisted identification of micro-organisms, see H.G. Gyllenberg & Niemelä (1975) and Sneath (1995).

Acknowledgement

The research was partially funded by The Bank of Sweden Tercentenary Foundation.

References

ANDERSEN, E.B. (1991). *The Statistical Analysis of Categorical Data Second, Revised and Enlarged Edition*. Springer Verlag, Berlin, Heidelberg, New York, Tokyo.

BAUM, E.B. (1988). On the capabilities of multilayer perceptrons. *Journal of Complexity*, 193- 215.

BODDY, L., MORRIS, C.W. & WIMPENNY, J.W.T. (1990). Introduction to neural networks, *Binary* **2**, 139- 144.

BRIDLE, J. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. *NATO ASI Series F*, **68**, 227- 236. Heidelberg: Springer.

BUNGAY, H. & BUNGAY, M.L. (1990). Identification of micro-organisms with a neural network. *Binary* **2**, 51 - 52.

CHUN, J., ATALAN, E., WARD, A.C. & GOODFELLOW, M. (1993). Artificial neural network analysis of pyrolysis mass-spectrometric data in the identification of streptomyces strains, *FEMS Microbiological Letters* **107**, 321-325.

DEMUTH, H. & BEALE, M. (1992). *MATLAB™ Neural Network Toolbox*. Natick, Mass.: The MATHWORKS Inc.

DUDA, R.O. & HART, P.E. (1973). *Pattern Classification and Scene Analysis*. Wiley: New York.

DYBOWSKI, W. & FRANKLIN, D.A. (1968). Conditional probability and the identification of bacteria. *Journal of General Microbiology* **54**, 215-229.

GOWER, J.C. (1974): Maximal predictive classification, *Biometrics* **30**, 643-654.

GYLLENBERG, H.G. (1963): A general method for deriving determination schemes for random collections of microbial isolates, *Annales Academiae Scientarum Fennicae, Series A. IV. Biologica* **69**, 1-23.

GYLLENBERG, H.G. (1965): A model for computer identification of bacteria. *Journal of General Microbiology* **39**, 401- 405.

GYLLENBERG, H.G. (1976). Development of reference systems for automatic identification of clinical isolates of bacteria. *Archivum Immunologiae et Therapiae Experimentalis* **24**, 1-19.

GYLLENBERG, H.G. & NIEMELÄ, T.K. (1975). Basic principles in computer-assisted identification of microorganisms. In *New Approaches to the Identification of Microorganisms*, Ch. 13. Edited by C-G. Héden & T.Illéni. New York: John Wiley and Sons.

GYLLENBERG, M., GYLLENBERG, H.G., KOSKI, T. & SCHINDLER, J. (1993). Nonuniqueness of numerical taxonomic structures. *Binary* **5**, 138-144.

GYLLENBERG, M. & KOSKI, T. (1994*a*). Numerical taxonomy and the principle of maximum entropy. *Journal of Classification*, in press.

GYLLENBERG, M. & KOSKI, T. (1994*b*). A taxonomic associative memory of most typical organisms based on neural computation (preprint).

GYLLENBERG, M., KOSKI, T. & VERLAAN, M. (1994). Classification of binary vectors by stochastic complexity, submitted, also available as preprint: *Research Report A5* (1994), University of Turku, Institute for Applied Mathematics.

HAYKIN, S. (1994). *Neural Networks. A Comprehensive Foundation.* New York: IEEE Press, Macmillan College Publishing Company.

HILL, L.R. (1974). Theoretical aspects of numerical identification. *International Journal of Systematic Bacteriology* **24**, 494-499.

KANERVA, P. (1990). *Sparse Distributed Memory.* 2nd Printing. London: A Bradford Book, MIT Press, Cambridge Mass.,

KOHONEN, T. (1989). *Self-Organization and Associative Memory.* 3rd Edition. New York: Springer Verlag.

LAPAGE, S.P., BASCOMB, S., WILLCOX, W.R. & CURTIS, M.A. (1970). Computer identification of bacteria. In *Automation, Mechanization and Data Handling in Microbiology,* pp. 1-22. Edited by A. Baillie & R.J. Baillie. New York: Academic Press.

LAPAGE, S.P., BASCOMB, S., WILLCOX, W.R., CURTIS, M.A. (1973). Identification of bacteria by computer: general aspects and perspectives. *Journal of General Microbiology* **77**, 291- 315.

LEWIS, P.M. (1959). Approximating probability distributions to reduce storage requirements. *Information and Control* **2**, 214-225.

LIPPMAN, R.P. (1988). Pattern classification using neural networks. *IEEE Communications Magazine,* pp. 47-64.

PANKHURST, R.J. (1991). *Practical Taxonomic Computing.* Cambridge: Cambridge University Press.

RATAJ, T. & SCHINDLER, J. (1991). Identification of bacteria by a multilayer neural network. *Binary* **3**, 159-164.

RICHARD, M.D. & LIPPMAN, R.P. (1991). Neural network classifiers estimate Bayesian posteriori probabilities. *Neural Computation* **3**, 461-483.

SHORE, J.E & JOHNSON, R.W. (1980). Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy. *IEEE Transactions on Information Theory* **IT-26**, 26-37.

SHORE, J.E & JOHNSON, R.W. (1981). Properties of cross-entropy minimization. *IEEE Transactions on Information Theory* **IT-27**, 472-482.

SHORE, J.E & GRAY, R.M. (1982) Minimum cross-entropy pattern classification and cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-4**, 11-17.

SNEATH, P.H.A. (1957*a*). Some thoughts on bacterial classification. *Journal of General Microbiology,* 17, 201- 226.

SNEATH, P.H.A. (1957*b*). The application of computers to taxonomy. *Journal of General Microbiology* **17**, 261- 268.

SNEATH, P.H.A. (1974). Test reproducibility in relation to identification. *International Journal of Systematic Bacteriology,* 24, 508 - 523.

SNEATH, P.H.A. (1979*a*). BASIC Program for identification of an unknown with presence-absence data against an identification matrix of percent positive characteristics. *Computers and Geoscience* **5**, 195-213.

SNEATH, P.H.A. (1979*b*). BASIC Program for determining the best identification scores possible from the most typical examples when compared with an identification matrix of percent positive characteristics. *Computers and Geoscience* **6**, 27-34.

SNEATH, P.H.A. (1995) The history and future potential of numerical concepts in systematics: the contributions of H.G. Gyllenberg. *Binary,* this issue.

WILLCOX, W.R., LAPAGE, S.P. & HOLMES, B. (1980). A review of numerical methods in bacterial identification. *Antonie van Leeuwenhoek* **46**, *233 - 299.*