Elucidation of Chemical Reactivity Using an Associative Memory System

Klaus-Peter Schulz and Johann Gasteiger*

Organisch-Chemisches Institut, Technische Universität München,
Lichtenbergstrasse 4, D-8046 Garching, Germany

Received July 6, 1992

An associative memory system, a model of the information processing in the human cerebellum, is used to investigate chemical reactivity in terms of polar bond breaking. The information which bonds in an organic molecule break or do not break, an information discernible from reaction databases, is generalized. An optimization procedure was introduced to find the associative memory system with the highest predictive power. In this way, the electronic and energy effects that have the greatest influence on chemical reactivity can be determined. The results are compared with those obtained from various pattern recognition methods. An associative memory system is able to represent the relationship between structure and reactivity in an implicit manner without the user having to specify an explicit mathematical relationship. Once trained it can be successfully applied to the prediction of chemical reactivity for a wide range of organic chemistry. This is demonstrated with some examples.

INTRODUCTION

Quite often, the course and outcome of a chemical reaction cannot be predicted with reliability. The purely theoretical approach is still too complex, and therefore asks for assumptions and simplifications that introduce errors that are often larger than the accuracy required by experiment.

In this situation, chemists have developed models and concepts for ordering their observations and for drawing conclusions by analogy. A powerful approach is to look for functional groups and then assume that a reaction course is followed that is analogous to one that has been observed in molecules having the same functional group. Problems, however, arise when several functional groups are present that interfere and compete with one another. Then, the various possible reaction courses have to be evaluated in order to decide which is the most favored one. For this endeavor, chemists have segregated various effects of energetic, electronic, and steric nature and have developed a feeling for their relative importance in a given situation. Limited success has been achieved in quantifying some of these effects largely through linear free enthalpy relationships.

Over many years we have developed empirical methods that allow us to calculate the magnitude of electronic and energy effects. Calculations and correlations of physical data from these parameters have established their value.[1] This laid the foundation for their use in correlating and predicting chemical data and in proposing details of the mechanism of organic reactions.[2,3] All this rested on deriving equations for calculating data on chemical reactivity; equations that have been obtained through statistical and pattern recognition techniques.[4] However, the basic assumption of such an approach that the dependence of chemical reactivity on structure can be expressed in a simple—mostly linear—mathematical equation might be too straightforward. It is clear that this is not the way a chemist derives his knowledge about chemical reactivity.

We have therefore looked into approaches that store the relationship between two properties in an implicit way rather than expressing it explicitly by an equation. This can be accomplished by neural networks that have recently gained prominence.[5] Another related approach is an associative memory. Both stem from formal models of the brain or the cerebellum. In contrast to neural networks, the formalism of distributed associative memories deviates more from the biological origin.[6] This paper describes the application of a distributed associative memory system to the study of the relationship between structure and chemical reactivity. A preliminary communication of this work has already appeared.[7]

The reactivity information used in this study, whether a bond breaks or does not break, is rather elementary. However such information can automatically be deduced from reaction databases. Thus, this approach outlined in this study offers a potential for machine learning of chemical reactivity.

ASSOCIATIVE MEMORY SYSTEM

The distributed associative memory (AMS) used in this investigation is based on the CMAC (cerebellar model articulation controller; later called cerebellar model arithmetic computer) algorithm of J. S. Albus.[8–10] The cerebellum is ultimately involved in the control of the movements of the limbs and the eyes. Detailed knowledge of the structure and function of the cerebellar cortex has been accumulated through anatomical and neurophysiological studies.

On the basis of these data a theory of cerebellar function has been developed by Albus.[11] Input to the cerebellum comes as sensory and proprioceptive feedback signals from the muscles, joints, and skin together with commands from higher level control centers (cerebral cortex) telling what movement is to be performed. According to Albus' theory, the input constitutes an address to a neuron (neural memory cell), the contents of which are the appropriate muscle actuator signals required to carry out the desired movement. At each point in time, the input addresses an output that drives the muscle movement. The movement produces a new input, and the process is repeated. The basic principles of how the cerebellum organizes input data, how it generates the addresses of where to store the control signals, and how the output control signals are generated have gone into the design of the CMAC algorithm.

The CMAC makes use of a table look-up technique to obtain the value of an arbitrary function or relationship. Unlike a

0095-2338/93/1633-0395$04.00/0

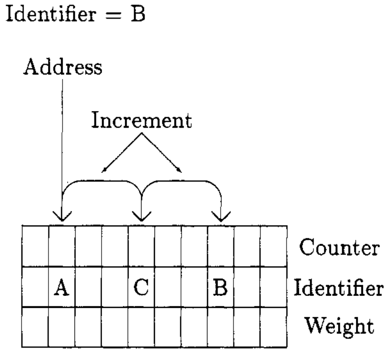


Figure 1. Organization of the memory and the mechanism for storing information. If the identifier of the input is different from the identifier of the cell being addressed, the address is incremented until both identifiers are equal, or until an empty cell is found.

simple table reference it is *able to generalize*, an important feature in a learning process. In the case of a function which projects into an *n*-dimensional space the AMS automatically performs an interpolation or an extrapolation to compute the result. In addition, it has the capacity of dynamically adapting to a changing environment.

The associated memory system (AMS) that we used comprises two main components, a memory and a series of hashcodes to address the memory. The memory consists of a row of cells. The hashcodes map every input to a set of cells in the memory. They are used in two fashions, in a learning mode (training) and a prediction mode (test). In both modes, the hashcodes project a distinct input onto the same set of cells. To accomplish this and to avoid unwanted effects due to hashing collisions, the memory cells are composed of three storage fields (triplets) (Figure 1). The first field contains a *counter*, which records the number of accesses in the training mode to this particular cell. The second field stores an *identifier*. This identifier is another hashcode generated from the input and provides a mechanism for handling unwanted hashing collisions. Information is stored in the memory cell being addressed only if it is empty or if the identifier from the input is equal to the identifier of the memory cell. Otherwise a new address is obtained by increasing the address with the increment. Then, the check on the identifiers is made again, and the process might be repeated. The use of an identifier reduces the amount of memory needed to store a sparsely filled matrix. The value of the function (*weight*) is the last part of the memory cell.

By using a hashcode, similar inputs should be put together in the same memory cell. The hashcoder generates a series of triplets of addresses with their associated identifiers and increments to avoid hashing collisions as much as possible and to better handle similarities in the input data. In the illustrative example of Figure 2, five such triplets of address, identifier, and increment are generated. In the application to chemical reactivity (vide infra) thirteen such triplets are used. Spreading the input information onto several blocks of addresses, identifiers, and increments allows the AMS to use the information it was trained on for generalization, to make predictions. Therefore, the number of triplets is called the degree of generalization.

Figure 2 illustrates the steps made to store the input information in the memory (training) or to extract information from the memory. It should be noted that the values (weights) in the memory are equal to 100 only if the cells are addressed the first time. In later steps of training and when predictions are made, the addressed cells might contain different values (weights).

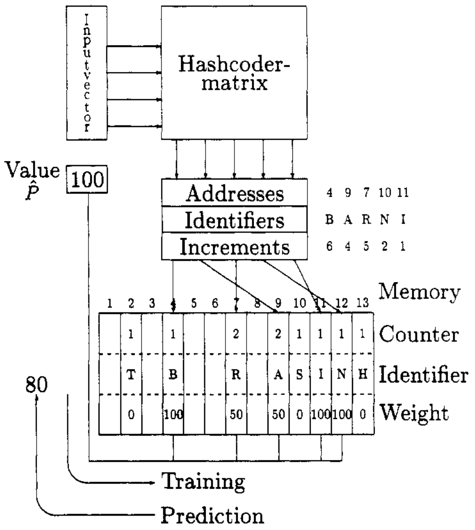


Figure 2. Training of an AMS with a second data point and predictions by an AMS. The first triplet (4, B, 6) generated by the hashcoder points to the fourth memory cell which is still empty and thus the information can be directly stored. The second triplet (9, A, 4) addresses the ninth cell which already contains information. The identifier of the addressing triplet is equal to the identifier of the cell, and therefore the information can be stored. It is set to the arithmetic mean of the weight already present (0) and that of the input (100), and the counter is increased by 1. The third triplet (7, R, 5) is preprocessed analogously. The fourth triplet (10, N, 2) points to a cell that already contains information. However, the identifier in the cell is different from the identifier of the input triplet. Therefore, the address is increased by the increment 2. The newly addressed cell (12) is empty, and the information can be stored. Thus, the identifier and the weight can be directly stored. If the contents of the memory as indicated would be used for prediction, a value of (100 + 50 + 50 + 100 + 100)/5 = 80 would be obtained.

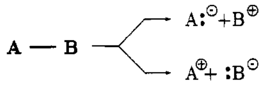


Figure 3. Two alternatives for the polar breaking of a bond.

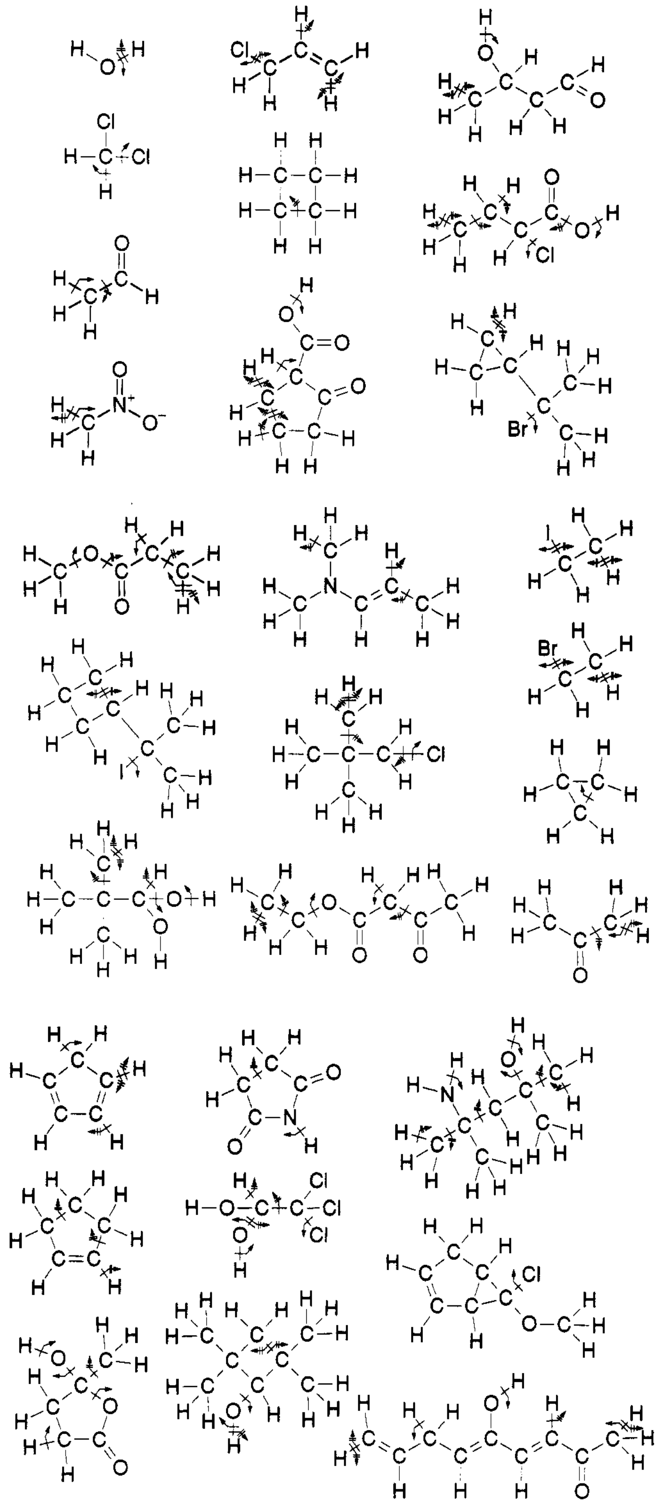
DATA SET FOR CHEMICAL REACTIVITY

Many important organic reactions proceed through heterolytic cleavage of bonds. The aim of this study was to predict the reactivity of single bonds in aliphatic compounds. We have selected a data set of 29 molecules that covers the range of polar reactivity in aliphatic compounds as broadly as possible.

This data set contains 362 single bonds. As each bond has two possibilities for breaking in a polar manner generating a positively and negatively charged species (Figure 3), 724 bond breakings have to be considered. However, quite a few are constitutionally equivalent. Thus, altogether 382 different types of polar breakings of single bonds can be found in this data set of 29 molecules. From these, 116 bond breakings were selected and put into the category reactive or nonreactive, respectively (see Scheme I). Altogether 42 reactive bonds were present, while 74 bonds were considered nonreactive.

This data set has been previously studied with various statistical and pattern recognition methods.[4,12] It contains an intentional misclassification so that the response of the AMS to this bad data can be studied. This misclassification comprises the heterolytic ring opening of unsubstituted cyclopropane that was taken as being reactive, although it is difficult to achieve and therefore should be put in the nonreactive category. A variety of physicochemical parameters were calculated for each bond by empirical methods.[13-17] These factors include the difference in total charge, Δq_{tot}, the

Scheme I. Data Set of Molecules with Reactive (Bent Arrows) and Nonreactive Bonds (Crossed Arrows) Indicated



σ-polarity, Q_σ, the difference in σ-electronegativity, $\Delta\chi_\sigma$, the difference in π-charge, Δq_π, the resonance stabilization of charges generated by heterolysis, R, the polarizability of the bond, α_b, and the bond dissociation energy, BDE. These parameters allow a detailed description of the structural features of a molecule in terms of energy and electronic effects.

IMPLEMENTATION OF AN AMS

The aim of an AMS is to obtain a response, $\hat{P}$, for a specific input, S. In order to achieve this, the memory must be trained. This is accomplished by a series of mappings

$$\mathbf{S} \rightarrow \mathbf{M^*} \rightarrow \mathbf{M} \rightarrow \mathbf{A}^{(0)} \rightarrow \mathbf{A^*}$$

These mappings convert the input information, **S**, into matrices **M*** and **M**, which are further transformed into preliminary addresses $\mathbf{A}^{(0)}$ and the final addresses **A*** and the contents of the memory cells.

In this general outline, our AMS follows the cerebellar model arithmetic computer (CMAC) algorithm of Albus.[8-10] The specific implementation is, however, different. One novel feature of our implementation of an AMS is its combination with an optimization phase for preprocessing the input data. Details are given in Appendix A.

Training Phase. In the training of an AMS the expectation value, $\hat{P}$, of the response or output is stored. However, a memory cell will be trained several times, and the more a cell is trained the more reliable its contents will be. Thus, the weight, W, in a cell is made dependent on the number of writing accesses to the cell as expressed in the counter, C.

For empty cells, the weight in the cell is set equal to the expectation value of the output.

$$W(1) = \hat{P} \qquad (1a)$$

If a cell already contains information, ($W(C)$), the new weight, ($W(C+1)$), is calculated by eq 1b.

$$W(C+1) = W(C) + \frac{\hat{P} - W(C)}{C+1} \qquad (1b)$$

Prediction Phase. The result of the prediction, the output, P, for an input vector is obtained by combining the weights of the r^* individual searches, W_i, resulting from the various triplets. In the simplest case, this can be the arithmetic mean (eq 2), with f being the no. of addressed filled cells ($f \leq r^*$).

$$P = \frac{1}{f}\sum_{i=1}^{f} W_i \qquad (2)$$

Preprocessing of Input Data. The original input data were preprocessed in two different ways, by straightforward *scaling* or by weighting factors obtained by an optimization procedure. The combination of an AMS with an optimization phase for preprocessing the input data is a novel feature of our AMS implementation.

The original input data comprise quantities with quite different units (e.g., bond dissociation energies in kilojoules per mole, charges in electron units). To allow for a proper comparison, the different quantities were scaled so as to bring the data into the range of integers 0–255.

To yield good results, the associative memory system asks for training with a grid of data points that are approximately evenly distributed over the input space. The input space contains all possible input vectors. For an input space of relatively high dimensionality a large number of data points are needed to obtain a well-trained memory.

To reduce the number of data points, the input space can be transformed by compressing the various coordinates of the input space by different amounts. In other words, a transformation is made that weights the components of the input vector according to their importance. This is done to train an AMS that should extract a maximum predictive capability for a given set of data. To accomplish this goal, the AMS is incorporated into a system that optimizes the predictive capability of the AMS for the chosen data set. By choosing an appropriately defined error index, the task is identical with the problem of finding the minimum in an n-dimensional space.

Two properties can assist in the evaluation of a preprocessing scheme. Firstly, an AMS can be judged on the basis of how well it can recognize the information used as an input in the training of the AMS (recognition test). Secondly, the quality of predicting unknown data, not used in the training of the AMS, can also serve as an indicator for the evaluation (prediction test).

Recognition Test. The recognition test was performed as follows: The AMS is trained with the entire information, i.e., all 116 bond breakings and their classification as either reactive (42) or nonreactive (74). Then, each individual polar breaking of a bond (heterolysis) from the data set of 116 was predicted with the fully trained AMS to give a value for its reactivity between 0 (nonreactive) and 100 (reactive). The deviations between the values of the classification given, $\hat{P}_j$, and the values obtained through prediction, P_j, were used to determine an error index, EI, from the deviations of all n data points by weighing and standardizing according to eq 3.

$$\mathrm{EI} = \frac{1}{n}\sum_{j=1}^{n} \frac{|\hat{P}_j - P_j|}{1 + \exp(5 - 0.1|\hat{P}_j - P_j|)} \qquad (3)$$

The form of the function is chosen so as to ensure that small deviations are penalized less than larger ones, for small deviations are indeed wanted as they change the reactivity from a logistic value (reactive or nonreactive) to a more continuous function. On the other hand, deviations of 50% and more change the classification of the bond into the wrong category. This should be avoided.

Prediction Test. A jackknife test[18] was used to determine the performance of the various AMS in a prediction test. The AMS was used to determine the reactivity of bonds not included in the data set used for the training of the AMS. In the jackknife test one data point (bond breaking) is deleted from the data set and the AMS is trained with all the other data points (115). The AMS thus obtained is used to predict the classification (reactivity) of the bond not considered in the training of the AMS.

This procedure is performed 116 times by deleting each data point once. The error index for the prediction, EI_e, is calculated by eq 3. In this case, each prediction provides one value P_j, which is compared with the classification, $\hat{P}_j$, (either 0 or 100). The summation of eq 3 runs over the prediction test of each individual data point in a separate jackknife test.

In contrast to the recognition test, the prediction of an untrained bond breaking may result in an empty set of active memory cells; i.e., every addressing triplet points either to an empty memory cell or to a memory cell whose identifier differs from the identifier of the addressing triplet. Then, the associative memory system cannot predict a value for the reactivity of such a bond breaking. This may occur when the investigated bond breaking is rather different from all the trained bond breakings.

The deviations resulting from bond breakings for which no predictions can be made should add less penalty to the error index than the deviations of predicted data. Therefore separate error indices are computed—according to eq 3—for predictable and unpredictable bond breakings.

Optimization. The input data stored in the vector S are multiplied with weighting factors, **B**, to obtain the preprocessed input vector (eq 4).

$$v_i = \text{round } (b_i{*}s_i) \qquad (4)$$

$$i = 1, ..., \text{no. of input data}$$

The weight factors for preprocessing are obtained either by scaling or by an optimization procedure. The scaling of the elements of the input vector is submitted to the round function to obtain integer data in the range from 0 to 255.

Table I. Combination of Parameters Selected for Describing the Polar Bond Breaking in the Input Vector

		parameter combination			
effect	symbol	1	2	3	4
difference in σ-electronegativity	$\Delta\chi_\sigma$	×	×		×
difference in π-charge	Δq_π	×			
resonance stabilization	R	×		×	
bond polarizability	α_b	×			
difference in total charge	Δq_{tot}	×	×		×
σ-bond polarity	Q_σ	×		×	
bond dissociation energy	BDE	×	×		

The factors, b_i, for scaling the input data, s_i (eq 4), are determined by minimizing a linear combination of error indices (eq 3) called the global error index $\mathrm{EI_T}$ (cf. eq 10). The global error index forms a hypersurface with many local minima. Since analytical derivatives are not available, we have implemented the Brent–Powell algorithm that uses the method of conjugated directions.[19] Details of the optimization of scaling factors are given in Appendix B.

RESULTS WITH THE AMS

The hashcoder matrix was initially chosen to generate 13 triplets for addressing the memory (in contrast to the five chosen in the illustrative example of Figure 2). This is to say that the degree of generalization of the AMS was thirteen. In later studies this value was changed (see Appendix C).

The description of the AMS pointed out that in the training phase the address is changed through incrementation if the identifier of an addressing triplet and that in the memory cell are different. This incrementation is repeated until the identifiers match, or a preset number of incrementations has been performed. When this number is reached, the contents of the memory cell are overwritten with the contents of the addressing triplet. Thus, the information in the memory cell is lost when such a collision occurs. To avoid such collisions in the AMS, the system was programmed to set the size of the storage space to an amount that leaves 35–40% of the cells empty.

Parameters and Preprocessing. First, different combinations of variables and different transformations were tested by preprocessing. For each bond breaking, either the entire set of seven electronic or energy parameters, or only a selection thereof, were investigated. Table I shows the different types of combinations of variables of the input vector. Preprocessing was performed and optimized when only a subset of variables was used (combinations 2, 3, and 4). The combinations of parameters 2, 3, and 4 have been selected because they had led to good results in previous studies with pattern recognition methods[4,12] or in a preliminary investigation with an AMS.[7] The correlation between the parameters in the data set on chemical reactivity is given in Table II with the correlation matrix. Only the pairs Q_σ, Δq_{tot} (0.87); Q_σ, $\Delta\chi_\sigma$ (−0.74); and α_b, BDE (−0.75) have a significant correlation.

Recognition. The first combination of parameters (containing all seven; see Table I) reproduces the selected classification very well (see Table III, set 1 of recognition). Only a few and slight deviations from this classification occur: 110 objects were within 0%, and all 116 bonds were within 10%. Of the reactive bonds two have a reactivity

Table II. Correlation Matrix of the Physicochemical Parameters in the Data Set

parameter	$\Delta\chi_\sigma$	Δq_π	R	α_b	Δq_{tot}	Q_σ	BDE
$\Delta\chi_\sigma$	1.00	0.25	0.00	0.04	−0.33	−0.74	−0.05
Δq_π	0.25	1.00	−0.21	0.10	−0.23	−0.37	0.01
R	0.00	−0.21	1.00	0.38	0.11	0.06	−0.29
α_b	0.04	0.10	0.38	1.00	−0.03	−0.06	−0.75
Δq_{tot}	−0.33	−0.23	0.11	−0.03	1.00	0.87	0.04
Q_σ	−0.74	−0.37	0.06	−0.06	0.87	1.00	0.06
BDE	−0.05	0.01	−0.29	−0.75	0.04	0.06	1.00
reactivity classification	−0.66	−0.30	0.28	−0.03	0.62	0.77	−0.02
Fisher quotient	1.76	0.18	0.17	0.00	1.21	2.92	0.00

Table III. Results of the Recognition and Predictions of the Reactivity of Bonds by the Various AMS

	recognition								prediction							
	1		2		3		4		1		2		3		4	
reactivity	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*
0	70	0	46	0	26	0	45	0	37	0	43	0	19	1	40	1
1–10	4	0	27	1	41	0	28	1	0	11	26	1	40	0	26	0
11–20	0	0	1	0	3	1	1	0	8	0	1	0	2	1	1	0
21–30	0	0	0	0	3	1	0	0	1	0	1	0	2	1	3	0
31–40	0	0	0	0	1	0	0	0	3	1	0	0	3	1	1	0
41–50	0	0	0	0	0	0	0	0	5	0	0	0	1	0	0	0
51–60	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	0
61–70	0	0	0	0	0	0	0	0	1	3	0	0	0	1	0	0
71–80	0	0	0	0	0	3	0	0	0	0	0	0	0	4	0	0
81–90	0	0	0	1	0	5	0	1	0	1	0	3	0	3	0	2
91–99	0	2	0	6	0	1	0	10	0	0	0	5	0	0	0	9
100	0	40	0	34	0	31	0	30	9	12	0	33	0	27	0	30
no prediction	0	0	0	0	0	0	0	0	12	12	3	0	5	3	3	0

between 91 and 99%; 40 bonds, a reactivity of 100%. Of the nonreactive ones, 70 are completely nonreactive (0) and 4 bonds have a reactivity between 1 and 10%. With other words, the reactivity is kept close to either very high or very low. At first glance, this may constitute a good result, although it is a little disappointing that the misclassification of the cyclopropane bond (see above) is not recognized. The weakness of this AMS will show up in the prediction test.

The second combination of parameters uses for each bond the values of the difference in σ-electronegativity, the difference in the total charge, and the bond dissociation energy and spreads the reactivity range a little further apart (see Table III, set 2 of recognition). From the set of reactive bonds one is perceived as nonreactive with a reactivity value between 1 and 10%, one bond has a reactivity between 81 and 90%, six bonds have a value between 91 and 99%, and 34 bonds have the highest reactivity value of 100%. From the set of nonreactive bonds 46 are perceived as completely nonreactive (0), 27 bonds have a reactivity between 1 and 10%, and one bond has a reactivity between 10 and 20%. The classification of one bond is changed from reactive to nonreactive. It turns out that this bond is the bond in cyclopropane that was intentionally set to a wrong classification. Thus, the second combination of parameters with its preprocessing gives only correct answers, in spite of a partially incorrect classification in the input.

The third combination of parameters uses only a two-dimensional space for the AMS spanned by the parameters for the resonance stabilization and the σ-polarity. It gives larger differentiation in reactivity (see Table III, set 3 of recognition). The classification of two bonds is changed in comparison to the initial assignment. This includes the correction of the intentionally introduced error of classifying the C–C bond in cyclopropane as reactive. In addition, the deprotonation of dichloromethane, that was classified as a feasible reaction, is given the low reactivity value of 23% by the AMS. Indeed, it is doubtful whether dichloromethane should be considered as acidic or not.

The last preprocessing gives results that are quite similar to the second combination of parameters (Table III, set 4 of recognition). This is not too surprising, as both preprocessing schemes use the same two parameters, the difference in σ-electronegativity and the difference in total charge, with the bond dissociation energy as an extra parameter in the second preprocessing. However, BDE has only a small weight in this combination. Only the differentiation of the reactivity of the bonds is slightly higher in the fourth parameter combination than in the second preprocessing. Again, the bond breaking that is put in the wrong category is the ring opening of cyclopropane, a satisfying result as it was intentionally put in the wrong category.

Prediction. The results of the jackknife test for the four preprocessing schemes are shown in Table III under the heading "prediction". The first parameter combination performs rather badly in the prediction test (set 1). From the set of reactive bonds, 12 cannot be predicted at all, another 12 are perceived as nonreactive (11 with a reactivity between 1 and 10%, 1 with a reactivity between 31 and 40%), 2 bonds have a reactivity of 51–60%, 3 of 61–70%, 1 of 81–90%, and 12 of 100%. From the set of nonreactive bonds 12 cannot be predicted, 10 are put in the wrong reactive class (9 with a reactivity of 100%, 1 with 61–70%); only the rest are predicted as nonreactive (5 in the range of 41–50%, 3 in 31–40%, 1 in 21–30%, 8 in 11–20%, and only half of the bonds (37) with a reactivity of 0). Only about three-fifths of the cases (70) are classified correctly and for one-fifth of the cases no information can be obtained. Furthermore, the high number of 22 wrong classifications clearly shows that this AMS has only low predictive power.

On the other hand, the picture of the jackknife test for the second combination (Table III, set 2 of prediction) is only slightly different from the picture of the recognition test (set 2 of recognition). Thus, by using these parameters and their corresponding weights an AMS with good predictive power is available. As expected, the predicted reactivity spreads in the prediction test more than in the recognition test. It is also

Table IV. Number of Misclassifications in KNN Analyses with Various Combinations of Parameters

combination	parameters used	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
1	all seven	5	5	4	4	6	6	6
2	$\Delta\chi_\sigma$, Δq_{tot}, BDE	4	4	7	5	4	4	5
3	R, Q_σ	6	6	4	4	5	5	5
4	$\Delta\chi_\sigma$, Δq_{tot}	3	3	3	3	2	2	3

gratifying that the intentional misclassification is corrected. Only three bond breakings cannot be predicted.

In the third combination of parameters larger differences between prediction and recognition test are observed (Table III, set 3 of recognition). Again the reactivity is more spread than in the recognition test. The same changes in classification occur; however, four additional false classifications are made. The reactivity of eight bonds cannot be predicted; three of them are reactive bonds.

The last combination of parameters has produced the results that are marginally different from those of the second one (Table III, set 4 of recognition). Only the reactivity of bonds is spread more than in the second preprocessing scheme.

COMPARISON WITH PATTERN RECOGNITION METHODS

When an AMS is compared with pattern recognition methods, k-nearest neighbor analysis (KNN) and cluster analysis will be first choices as all of them use similarity measures in their processing of data.

All four combinations of parameters of Table I, all seven, one three-, and both two-parameter combinations, have been analyzed by KNN analyses. Table IV shows the results of these investigations.

All misclassifications—including the one of the bond in cyclopropane, which should not be perceived as a misclassification since it was intentionally put in the wrong class—which are obtained with $k = 5$ with combinations 2, 3, and 4 are also classified incorrectly by the AMS or are either perceived by the AMS as being nonpredictable or are classified correctly with a rather low reliability (<30%). In this sense, the AMS has advantages as it does not simply predict the reactivity of bonds but determines that bond breakings that are too far outside the range of the other data (in terms of electronic and energy parameters) cannot be predicted or can only be predicted with low reliability.

Another advantage of the AMS lies in the weighting of parameters by the optimization procedure. This shows up most clearly in the comparison of the combinations 2 and 4. In adding another parameter, BDE, to combination 4 and thus generating the combination 2, the results of the KNN analysis get worse (the misclassifications increase from 2 to 4). The AMS, on the other hand, can marginally improve the results as the BDE parameter obtains a low weight in the optimization procedure but is used as an additional parameter to help improve the classifications.

At the same time, a series of hierarchical cluster analyses were performed. The comparison of the groupings resulting from the cluster analyses with the classifications given by a chemist shows the greatest correspondence when using a Mahalanobis distance and the linkage method of Ward. The most favorable combination of two physicochemical effects consists of the resonance effect, R, and the σ-bond polarity, Q_σ.

Taking the partitioning into thirteen clusters leads to four misclassifications—including the heterolytical opening of the cyclopropane ring. Even less misclassifications can be obtained when the difference in total charge, Δq_{tot}, is taken as an additional parameter. The separation into sixteen clusters yields only two misclassifications: the polar breaking of a C–C bond in cyclopropane and the loss of a methyl cation in acetone. The high correlation between the Q_σ-bond polarity, Q_σ, and the difference in total charge, Δq_{tot}, (c.f. Table II) does not depreciate this result, because of the use of the Mahalanobis distance which takes the correlation of parameters into account. More details on the results of KNN and cluster analyses are given in ref 12.

One great advantage of the AMS in comparison with nearly all other statistical or pattern recognition methods lies in the information on the reliability for each prediction. Another disadvantage of KNN analysis or cluster analysis—if they use statistical distances or standard cluster analysis techniques—as opposed to the AMS is that the training and prediction phases are not separated. In such a KNN or cluster analysis, the prediction of a new data point (bond breaking) asks for its inclusion in the entire data set and then the performance of an analysis on the new, complete data set. Thus, the computation on a full data set has to be redone for each prediction, and computation times increase with the number of data points.

Even if a KNN or a cluster analysis method is used that does not require a full recomputation on the extended set, the best performance that can be achieved by an KNN is a linear proportionality to the number of data points (N), while the performance of a cluster analysis is proportional to N^2.

The training of an AMS can ask for quite large computation times when an optimization of the weights of the parameters is performed and a sizeable training set is used. In our example, depending on the number of parameters used, on the average 80 CPU min were required on a mVAX 3800. However, an AMS has to be trained only once. Therefore, predictions are very fast with an AMS (up to 440 predictions per CPU second on a MicroVAX 3800) as only the new data point has to be evaluated and compared with the AMS trained before and then is available for predictions on any additional data point.

CHEMICAL SIGNIFICANCE OF THE RESULTS

The information on chemical reactivity which is contained in the classification of reactive and nonreactive bonds for the molecules of Scheme I has been condensed into 3.8 kB of storage space. This small part of memory contains the relationship between structure and chemical reactivity in an implicit manner. This alleviates the problem of defining an exact explicit mathematical relationship between structure and reactivity which is usually an oversimplification of the problem.

Furthermore, this memory has predictive power; i.e., conclusions on the reactivity of bonds not used in the training can be drawn. The predictive power of the AMS has been investigated with the jackknife test. To further explore the scope of the predictions on chemical reactivity that can be made with this associative memory system, various organic structures containing a variety of functional groups in different settings were investigated with the AMS.

These investigations were made with the AMS which showed the greatest predictive power. Studies with preprocessing Scheme 2 of Table I lead to an AMS capable of predicting reactivity values for all heterolyses and which corrects the

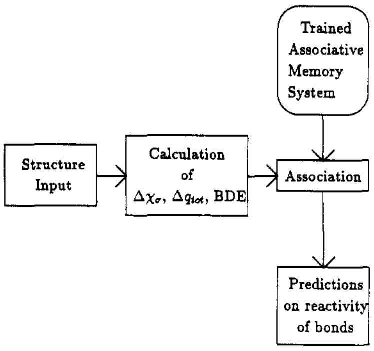

Trained Associative Memory System
Structure Input
Calculation of $\Delta\chi_\sigma$, Δq_{tot}, BDE
Association
Predictions on reactivity of bonds

Figure 4. Function flow for the prediction of the reactivity of bonds in an organic molecule.

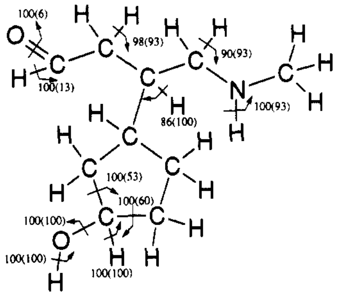


Figure 5. Reactive bonds as predicted by the AMS. The values give the probability of bond breaking in percent; the values in parentheses are the reliabilities of the predictions in percent. The arrows indicate the shift of the electron pair on heterolysis.

intentional misclassification without introducing a new one. These studies are outlined in Appendix C.

All of the following studies on the reactivity of bonds in organic molecules (Figures 5–9) were made with an AMS that uses the parameters σ-electronegativity difference, $\Delta\chi_\sigma$, difference in total charge, Δq_{tot}, and bond dissociation energy, BDE. The weights of these parameters were optimized (preprocessing Scheme 2 of Table I).

The number, r^*, of active memory cells for each input vector, i.e., the degree of generalization, was set to 15 since the previous study had shown that this gave optimum performance (cf. Figures 11 and 12).

After the input of a structure, the charge distribution is calculated by the PEOE method.[13] This gives the values for the total charge difference and the σ-electronegativity difference for each bond of the molecule. In addition, the BDE is calculated for each bond by an additivity scheme.[17] Using these three parameters for each bond as an input, the trained AMS is accessed to make predictions on the probability of polar bond breaking (Figure 4).

Four different molecules were chosen to study the predictions made by the AMS. The first example only consists of functional groups contained in the training set (cf. Scheme I). The aim was to test the influence of a slightly modified chemical environment on the predictions and to observe the predictions of bonds which were not contained as a preclassified data point in the training set. The next two molecules test atoms—sulfur and phosphorus—bonds and functional groups that differ from those of the training set, while the last example studies the influence of two or more functional groups on each other.

In the example of Figure 5 three different functional groups (secondary alcohol, aldehyde, secondary amine) were set into different molecular environments but well separated from each other.

All bonds that were predicted to be reactive are indicated in the figure. The discussion is started at the OH site. Deprotonation of the OH group and loss of OH^- are reactions of high probability. These predictions have high reliability (100%). This is not too surprising as these types of bond breakings were also marked reactive in the molecules of the training set (cf. Scheme I). Deprotonation at the α-carbon of the OH group is found to be a facile reaction. Apparently, the resulting carbanion is recognized to be stabilized by the inductive effect of the OH group, an effect that is overestimated. This conclusion is supported by the fact that the two C–C bonds adjacent to the OH group are predicted to break into a direction that generates a carbanion at the α-carbon. However, this information is less reliable (53 and 60%, respectively).

We now turn our attention to the bonds around the nitrogen atom: Deprotonation of the nitrogen atom is found to be an easy reaction. Of the two types of C–H bonds at the two carbon atoms next to the nitrogen, only the one at the methylene group is found to be reactive (90% reactivity with 93% reliability), whereas the one of the methyl group is predicted to be nonreactive (1% reactivity with 100% reliability). Of course, in reality there is not such a marked reactivity difference between these two C–H bonds. The AMS distinguishes the two reactions by their electronic properties: $\Delta\chi_\sigma$ and Δq_{tot}. In the case of the methyl group there is an equivalent nonreactive example of a methyl group adjacent to a nitrogen of an amine function (cf. Scheme I) in the training set. This explains the prediction for the methyl group. However, the properties considered in the input vector of the methylene group acidity are very similar to those of the deprotonation reaction of the methyl group in ethanal. This last reaction has to be regarded as reactive (cf. Scheme I), because of the resonance stabilization of the negative charge, an effect that cannot take place in the case of the methylene group. The implicit estimation of the effect of resonance stabilization from the three parameters $\Delta\chi_\sigma$, Δq_{tot}, and BDE fails in the case of the methylene group. Obviously, the data set does not contain any pair of reactions that differ only in the values of the resonance effect and the reactivity classification. Therefore, AMS optimization (and other methods) lead to the assumption that the effect of the resonance stabilization on the reactivity can be expressed using the combination of $\Delta\chi_\sigma$, Δq_{tot}, and BDE. This assumption has to be reconsidered if other examples—different from the one reported—are found which are incompatible with it.

The reactivity of the tertiary C–H bond must be attributed to an overestimation of the stabilization of a carbonion by three carbon atoms as neighbors analogously to the deprotonation reaction of the methylene group adjacent to the nitrogen atom.

The AMS was trained for the recognition of reactivity of single bonds only. Nevertheless, it comes up with the prediction of reactivity for the CO double bond of the aldehyde group in a direction that shifts the negative charge to the oxygen atom. However, the reliability of the information is rather low (6%). Apparently, only one of the fifteen memory cells that had been addressed contained information. On the other hand, no prediction can be made for the breaking of the CO double bond in the direction that shifts the negative charge to the carbon atom. Deprotonation of the aldehyde is found reactive, but again this prediction is not very reliable (13%, corresponding to two cells with information). This deprotonation does seem to be a viable reaction if it is not hidden by a competition of nucleophilic attack at the carbonyl group. Deprotonation at the carbon atom α to the carbonyl group is correctly predicted to be an easy reaction (98% reactivity,

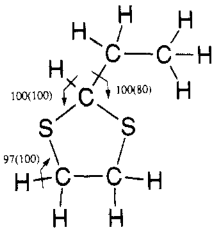


Figure 6. Reactive bonds in 2-ethyl-1,3-dithiole as predicted by the AMS.

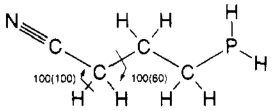


Figure 7. Reactive bonds in 4-phosphinobutyronitrile as predicted by the AMS.

estimated with 93% reliability, coming from fourteen of the fifteen cells being addressed).

All reactive bond breakings of the molecule of Figure 5 are indicated in the figure; those heterolyses for which no prediction could be made have been mentioned in the text. All other bond breakings were predicted to be nonreactive (with a reactivity between 0 and 6%).

The treatment of atoms and bonds not contained in the training set is investigated with the second example, 2-ethyl-1,3-dithiole, as no sulfur atom is in the set of 29 molecules used in the training of the AMS (cf. Scheme I).

The identity of atoms and bonds in a molecule is condensed into electronic and energy parameters and thereby generalized. This allows the AMS to make inferences also on atom and bond types, as well as on functional groups, not contained in the training set. The acidity of the proton at carbon atom 2 of 2-ethyl-1,3-dithiole is correctly predicted with high reliability (Figure 6). The AMS draws this conclusion from the inductive effect of the two sulfur atoms stabilizing the carbanion and the low BDE of this CH bond. The very same factors make the CC bond between the ethyl group and the ring carbon atom (C2) reactive. In reality, there is a large reactivity difference between the C–H and the C–C bond at this carbon atom. The reason lies in the stabilization of the proton by the solvent, an effect not being taken care of by our parameters.

The C–H bond at carbon atom 4 is also predicted to be reactive, although slightly less so as the carbanion can only be stabilized by one sulfur atom. Clearly, the decrease in acidity is underestimated. This, again, shows that the AMS is not able to convert the binary information of whether a bond is reactive or not into a fully developed reactivity scale (at least not with the limited data set used in this study).

In the molecule of Figure 6, reactivity predictions for all bond breakings could be made, all reactive ones being indicated in the figure. All the other heterolyses were predicted to be nonreactive. The next example, 4-phosphinobutyronitrile (Figure 7) was chosen because it contains two functional groups, the phosphino and the nitrile groups, that are not contained in the training set.

Only two bonds are classified as reactive, the removal of a proton from a CH bond in α position to the nitrile group and the heterolysis of a CC bond adjacent to the nitrile group.

Both bond breakings generate a negative charge which is stabilized by the nitrile group. Furthermore, it is correctly perceived that the phosphine group does not induce any reactivity in an alkyl chain and that the PH and PC bonds do not easily break in a polar manner.

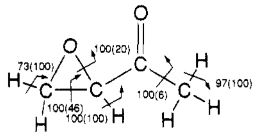


Figure 8. Reactive bonds as predicted by the AMS.

The example in Figure 8, 3,4-epoxybutan-2-one, shows a high concentration of functional groups, three adjacent carbon atoms bearing heteroatoms. No oxirane ring was contained in the training set, nor was any classification given for an ether bond.

The AMS predicts a high reactivity (with high reliability) for the deprotonation of the methyl group adjacent to the carbonyl group. Thus, the polarization and stabilization effect for a carbon anion by the carbonyl group is correctly perceived. Furthermore, it is found that a methyl cation can be lost from the acetyl group. Apparently, the oxirane ring induces an extra polarization in this bond that gives a reactivity beyond that of the corresponding bond in acetone. However, the conclusion that this bond is reactive should be taken with much caution as the reliability of this information is very low; only one memory cell from fifteen contained this information. In accordance with this, in reality, this bond should be considered as nonreactive.

The CH bonds at both positions of the oxirane ring are considered as prone to deprotonation, with the CH bond a to the carbonyl group having the higher acidity. The three-membered ring and the oxygen atom in this ring do indeed induce some degree of acidity in these bonds, although the extent of this reactivity at carbon atom 4 seems somewhat overestimated. The high acidity at carbon atom 3 is reasonable as a carbanion at this position can be stabilized by the carbonyl group.

The breaking of the oxirane ring is of high reactivity; a finding that corresponds to experience. The breaking of the CC bond of the oxirane ring—to give a 1,3-dipole—is an observed reaction although it usually needs more stabilization than the stabilization provided by one carbonyl group.

In summary, most of the bonds found reactive in this molecule do have some pronounced degree of reactivity.

Clearly, the results presented in this reaction show that a proper selection of the data set used for training is of critical importance. The predictions of bond reactivity agree in most cases quite well with chemical experience. However, the data set seems to be somehow biased in overestimating the importance of the inductive effect and underestimating that of the resonance effect.

CONCLUSIONS

It could be shown that an associative memory system (AMS) can be trained to store the relationship between structure and chemical reactivity in a small amount of memory. The quality of predicting the reactivity of bonds is, however, strongly dependent on the parameters chosen for representing the nature of a bond. All important electronic and energy effects have to be included. Their selection should be based on the analysis by statistical or pattern recognition methods. Furthermore, the importance of the various parameters should be determined by carefully optimizing their weights.

A very helpful feature of an AMS is that it also provides information on the *reliability* of its estimates and tells when it can make no predictions based on the information presented in the training.

The training of an AMS can ask for quite long computation times when an optimization of the weights of the parameters is involved. However, this procedure has to be done only once; predictions with such a trained AMS are fast.

It must be realized that the information on chemical reactivity used in the present study—whether a bond is reactive or not—is of a rather crude nature. Thus, prediction from an AMS trained with such information must be rather elementary. Although it manages to spread reactivity to a certain extent on a scale because a range of different molecules and bonds was presented to it, to a large part it still behaves like a classifier putting the bonds either at 0 or 100% reactivity.

An inherent drawbak in the primary information is that molecules of quite different reactivity are compared and, nevertheless, some of their bonds are fixed as reactive without giving any means for comparison of reactive bonds between different molecules.

Clearly, quantitative information on reactivity is by far to be preferred as a foundation for determining the relationship between structure and reactivity. However, we have shown that even such crude information as the classification of reactive and nonreactive bonds in a rather limited data set can be beneficially used to arrive at novel predictions on reactivity.

ACKNOWLEDGMENT

We thank Dr. Paul Hofmann for the initial implementation of the AMS in our group and Dr. Peter Löw for setting up the data set, indicating the reactive and nonreactive bonds and performing extensive statistical studies on these data. We gratefully acknowledge support of our work by the Bundesminister für Forschung und Technologie. Fruitful discussions with Prof. Dr. J. Zupan, Ljubljana, helped to improve the presentation.

APPENDIX A: DETAILS OF THE AMS ALGORITHM

The S → M⁺ Mapping. The input data are split into two parts according to eq 5, which is in essence a modulo-function.

$$v_i = q_i r^* + r_i \quad \text{with} \quad 0 \le r_i \le r^* \tag{5}$$

In this equation, r^* is the chosen number of active memory cells (in the previous example (Figure 2) this value was 5), q_i is an integer quotient, and r_i gives the positive remainder of the division. The matrix elements $m^*_{i,l}$ used for the calculation of the triplet of address, identifier, and increment are calculated by eq 6.

$$m^*_{i,l} = \begin{cases} q_i \bmod 2^8 & l = 1, ..., r_i \\ (q_i - 1) \bmod 2^8 & l = r_{l+1}, ..., r^* \end{cases} \tag{6}$$

In the case of slightly different input vectors for two data points, eqs 5 and 6 will produce matrices **M*** that have some identical row vectors because similar input vectors give mostly equivalent quotients q_i. If the q_i differ by 1, the shifting property of eq 6 will produce equal row vectors.

Consider, for example, a one-dimensional problem with two input "vectors" $v^{(1)}$ and $v^{(2)}$. Let $v^{(1)} = 25$, $v^{(2)} = 27$ and $r^* = 13$. Then $q^{(1)} = 1$, $q^{(2)} = 2$, $r^{(1)} = 12$, and $r^{(2)} = 1$. This gives the following "matrices":

$$\mathbf{M}^{*(1)} = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0)$$

$$\mathbf{M}^{*(2)} = (2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

Thus, there are only $|v^{(1)} - v^{(2)}|$ different entries in the matrices **M***.

Equal rows in the **M*** matrices that have the same position, lead to the same cell in the memory. This means that the number of identical cells addressed by different input vectors increases with their similarity.

The M* → M Mapping. The procedure is continued by the addition of 2^8 to all those elements of **M*** that have a negative value. This ensures that the resulting matrix **M** has only nonnegative elements.

The M → A⁽⁰⁾ Mapping. The elements of **M** can each be stored in 1 byte. Four such elements in a row can be taken together to one 32 bit word.

If less than 4 bytes are available, trailing bits are set to zero. In our example, a maximum of seven parameters was needed to describe the nature of a bond. The first word is constructed from the first four elements (bytes) $m_{i,l}$ ($l = 1, ..., 4$) and the second word from the last three elements $m_{i,l}$ ($l = 5, ..., 7$) and the last eight bits being equal to zero. This transformation is done for all $i = 1$–r^* rows of **M**. The words are treated by two scrambling functions MIX1 and MIX2 that contain SWAP, ROTATE, EXCHANGE, and EXCLUSIVEOR commands. Both words are separately processed by the scrambling functions and then split into two 16 bit parts. The first part is used for the calculation of the addresses; the second, for the determination of the identifiers and increments. All parts are multiplied by coefficients (c_{i1}, c_{i2}, d_{i1}, d_{i2}, e_{i1}, e_{i2}) that have previously been obtained by a random number generator and stored in a matrix. The coefficients are read from this matrix so as to make an unambiguous mapping of the input vector transformed into r^* triplets for addressing.

The products of the address part and the combined identifier and increment part are summed separately. Splitting the second 16 bit part gives the identifier and the increment. The addressing part is mapped into the range of addresses actually used for the memory cells. By application of this mapping onto all r^* columns of **M** the r^* triplets of address, identifier, and increment, called $\mathbf{A}^{(0)}$, are obtained.

The A⁽⁰⁾ → A* Mapping. Locating the addresses in the memory with this triplet depends on whether one is in the *training phase* or in the prediction phase. In the training phase, the AMS searches the memory for a cell that has the same identifier as the addressing triplet, or for a cell that is empty (i.e., that has no identifier).

The search starts with the cell that is accessed by the primary address. The address is increased in steps by the increment until the appropriate memory cell—same identifier or empty—is found. Then, the information to be learned (identifier and weight) is stored, and the counter is increased by 1. To keep the search within a reasonable boundary in a densely populated memory, it is stopped after a preset number of incrementation steps (in our case after 20 incrementations). In this latter case, the information in the memory cell last addressed is overwritten. The preference for the latest information ensures that each AMS is capable of learning new relations.

In the *prediction phase*, the AMS searches the memory in the same manner as in the training phase. The only difference is that the search is not stopped when an empty cell is found but rather continued to find a cell that contains information. If after the preset number of incrementations the identifier of the cell accessed is different from the one in the addressing triplet, no information can be derived.

APPENDIX B: OPTIMIZATION OF SCALING FACTORS

The following scheme gives an outline of the optimization algorithm for the error indices:

step 1 give a starting point $B^{(0)}$ for the scaling vector

step 2 calculate the global error index for this scaling vector

(1) train an AMS with all data points S, $\hat{P}$

(2) calculate a response, P, for all data points, S, using the AMS of step 1

(3) calculate the error index, $\mathrm{EI_R}$, for this recognition test according to eq 3

(4) do for each data point k the following steps: (a) train a new AMS with all data points except data point k (jackknife test); (b) predict a value, P_k, for this data point k using the AMS of step 4a

(5) calculate, analogous to eq 3, an error index, EP_p, for this jackknife test for all n_p predicted data points

(6) calculate with eq 3 an error index, EI_N, for this jackknife test for all n_n data points that cannot be predicted

(7) calculate the total error index, $\mathrm{EI_{PT}}$, for the jackknife test with eq 7

$$\mathrm{EI_{PT}} = \frac{1}{n}(n_p\mathrm{EI}_P + n_n\mathrm{EI}_N) \quad (7)$$

(8) calculate the global error index, $\mathrm{EI}_T{}^{(t)}$, for this iteration t according to

$$\mathrm{EI}_T{}^{(t)} = c_R\mathrm{EI_R} + c_\mathrm{A}\mathrm{EI_{JA}} + c_\mathrm{PT}\mathrm{EI_{PT}} \quad (8)$$

(In our algorithm, the following weights were used: $c_R = 0.1$; $c_\mathrm{A} = 0.3$; $c_\mathrm{PT} = 0.6$)

step 3 check the termination criterion for ($\mathrm{EI}_T{}^{(t)}$, $B^{(t)}$)

if $2\|B_\mathrm{opt}{}^{(t-1)} - B_\mathrm{opt}{}^{(t)}\|_2 < \epsilon^{1/2}\|B_\mathrm{opt}{}^{(t)}\|_2 + T$, then stop

(ϵ = machine precision $1 + \epsilon > 1$; T = tolerance)

step 4 do an optimization step with the Brent–Powell $\Theta(X^{(t)}, Y^{(t)}, X^{(t-1)}, Y^{(t-1)}, ...)$ algorithm by changing B

$$B^{(t+1)} = \Theta(B^{(t)}, \mathrm{EI}_T{}^{(t)}, B^{(t-1)}, \mathrm{EI}_T{}^{(t-i)}, ...)$$

step 5 go to step 2

APPENDIX C: VARYING THE DEGREE OF GENERALIZATION, R^*

To complete the optimization studies, the performance of the AMS under different degrees of generalization was investigated. In this study, the number of memory cells that obtain information from one set of input data (one polar bond breaking) is changed. With a rather low degree of generalization the AMS works more and more like a simple hashtable. In other words, with decreasing r^* the recognition gradually becomes a simple recall. In the extreme, each data point is described by a single value of the hashcode. This is fine for finding the data point in the recognition test. However, such an AMS has no predictive value at all.

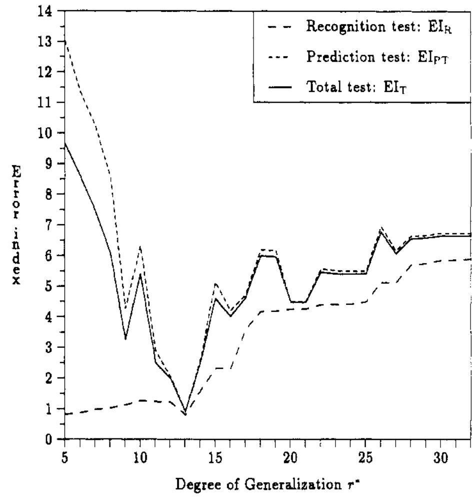


Figure 9. Dependence of the error indices EI on the degree of generalization: EI_R, recognition test; $\mathrm{EI_{PT}}$, prediction test for all data points (eq 9); $\mathrm{EI_T}$, total error index (eq 10). The weights b_i are kept constant.

All investigations were made with the second parameter combination of Table I using the difference of σ-electronegativity of the atoms of a bond, the difference in their total charge, and its bond dissociation energy. The weights b_i for the various parameters were determined by optimization taking $r^* = 13$.

This combination of parameters performed well both in the recognition test and in the prediction test. The amount of cells containing information was kept at 60–65% in all cases. The first question to be answered is: what happens if the degree of generalization is varied, and the weights, b_i, are held constant?

Figure 9 shows that all error indices have a minimum at $r^* = 13$, at a generalization degree of thirteen. This is not too surprising as the weights of the input parameters were optimized at this degree of generalization.

The decrease of the error index in the recognition test (EI_R) at degrees of generalization lower than nine has to be ascribed to the isolation of the individual data points. Here we have the case that the AMS is good for recognition but has no predictive power anymore as indicated by the high values of $\mathrm{EI_{PT}}$. The large difference between EI_R and $\mathrm{EI_{PT}}$ at low degrees of generalization shows that the AMS is only good as a recall system.

Figure 10 shows that the mean value of the counter, C, increases and the number of data points that cannot be predicted decreases with the degree of generalization, r^*.

The area of generalization of one input vector contains all those input vectors that are similar to the considered vector and can therefore influence it. Similarity in this context means that two similar input vectors must have at least one addressing triplet and therefore one activated memory cell in common. For an explanation, the notion of the area of generalization is introduced. The area of generalization of an input vector is a multidimensional prism around the tip of an input vector. The edges of this hyperprism have a length g_i that can be calculated by eq 9 from the degree of generalization, r^*, and the weights of the component of the scaling factors, b_i. With

$$g_i = 2r^*/b_i \quad (9)$$

increasing r, the area of generalization spreads and more and more data influence each other. The result is that the average counter of the memory cells that contain information in the

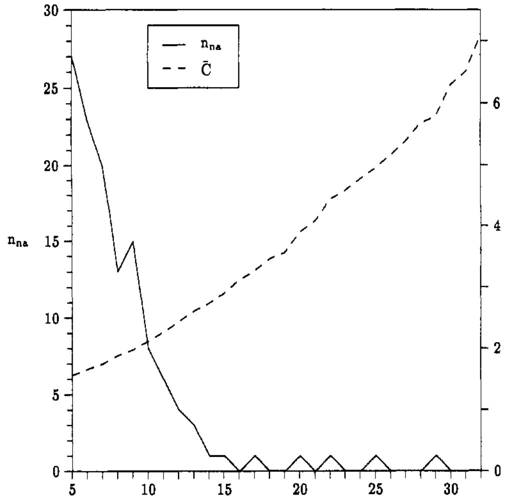


Figure 10. Dependence of the number of data points that cannot be associated, n_{na}, and the mean value of the counter, C, on the degree of generalization. Again, the weights b_i were held constant.

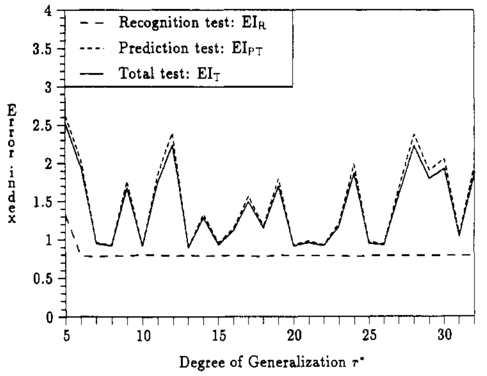


Figure 11. Dependence of the error indices on the degree of generalization at constant area of generalization: EI_R, recognition test; $\mathrm{EI}_{\mathrm{PT}}$, prediction test; EI_T, total error index.

reproduction test increases. Simultaneously, the number of data points (bonds) that cannot be associated decreases.

With decreasing r^*, the area of generalization shrinks (cf. eq 9) and therefore fewer bonds are similar to the bond under consideration and can be used for the prediction of chemical reactivity. The area of generalization has been optimized for the degree $r^* = 13$. Thus, any change of r^* leads to a deterioration of the predictive capability of the AMS. This can be clearly seen in Figure 9 from the values of the error indices, $\mathrm{EI}_{\mathrm{PT}}$ and EI_T. The error index EI_R in the recognition test essentially shows the same trend. However, for values of $r^* < 9$ the individual data points become more and more isolated in their areas of generalization. Thus, with decreasing r^* the AMS works more like a hashtable and the recognition becomes a recall.

The next question that emerges from the preceding discussion is what happens if the degree of generalization is varied while the area of generalization, g_i, is held constant?

The error indices show a much smaller variation when the area of generalization is kept constant (Figure 11). This can be achieved by changing the scaling factors, b_i, simultaneously with r^* so as to keep g_i constant (cf. eq 9). The error index for recognition, EI_R, stays constant except for very small values of the degree of generalization. The error index for the jackknife test, $\mathrm{EI}_{\mathrm{PT}}$, exhibits irregular deviations that are small in comparison to those of Figure 9. Analogous to Figure 9 there is an optimum at $r^* = 13$.

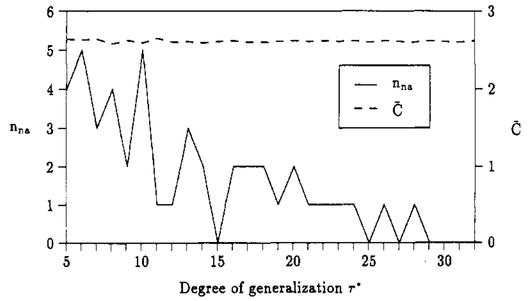


Figure 12. Dependence of the mean counter, C, and the number of data points that cannot be associated, n_{n}, on the degree of generalization, r^*, in a constant area of generalization.

Basically the same picture emerges when the average counter, C, and the number of data points that cannot be associated in a constant area of generalization (Figure 12) are considered. The mean counter, C, stays constant, as expected. The number of data points, n_{na}, that cannot be predicted shows fluctuations that are small compared to the results of Figure 10. Overall, n_{na} decreases with increasing degree of generalization. This decrease is due to a statistical effect on the unwanted hashing collisions: an increase in the degree of generalization, r^*, raises the probability of a hashing collision in at least one of the r^* addresses.

From Figures 11 and 12 two optimum associative memory systems can be derived: The AMS with the degrees of generalization of 15 and 25 find the intentional misclassification (as do all other AMS of this series), generate no wrong classifications and can make predictions for each data point (bond).

The results of Figures 11 and 12 show that the notion of the area of generalization is able to explain the main effects of a variation of the degree of generalization but not all. Other effects may stem from the discretization of input vectors (eq 4) and the hashing functions.

The influence of the hashing functions becomes clear when the global error index, EI_T, and the number of bonds for which no prediction can be made, n_{na}, are plotted against the number of memory cells. Holding all other parameters constant the variation of the global error index, EI_T, is based solely on effects of the hashing algorithm. Figure 13 shows the results for three different degrees of generalization: 6, 13, and 26. The starting point for the number of memory cells was taken from the tests of Figure 11 (the values are not shown in Figure 11). The subtraction of 10 from these values yields the lower bounds for the number of memory cells in Figure 13, adding 10 yields the upper bounds.

Moreover, the high values of the global error index correspond to a low number of bonds that cannot be predicted. This correspondence allows the conclusion that the fluctuations are mainly due to hashing collisions and that these hashing collisions also explain the fluctuations of the error indices in Figure 11 and of n_{na} in Figure 12. Since the hashing collisions lead to wrong predictions the reliability of such a prediction will be very small. Thus, if only taking predictions with high reliability are considered, no problems from hashing collisions will be encountered.

Altogether, Figures 11 and 12 make evident that it is possible to create an optimal AMS for most degrees of generalization. To minimize the number of hashing collisions and to maximize the speed of the associations, the degree of generalization should be chosen as low as possible. To reduce the effect of hashing collisions and of the discretization of the data the degree of generalization should be as large as possible. Thus, the optimum choice lies in a degree of generalization between

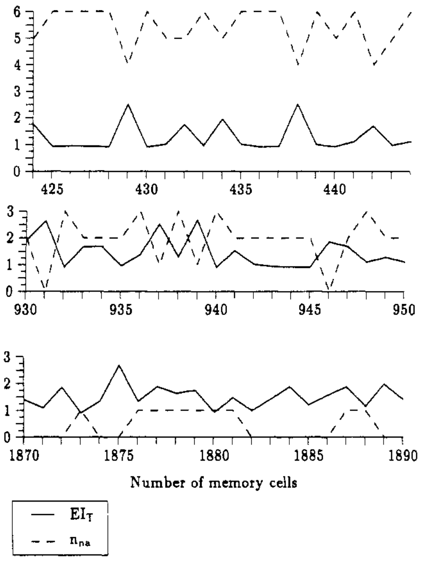


Figure 13. Dependence of the global error index, EI_T, and the number of reactions that cannot be predicted, n_{na}, on the number of memory cells the AMS consists of. This is shown for three different values of the degree of generalization, r^*. For $r^* = 6$ (a, top), $r^* = 13$ (b, middle), and $r^* = 26$ (c, bottom).

the values of 10 and 20. As an example, the AMS with $r^* = 15$ from Figures 11 and 12 forms a good basis for additional investigations.

REFERENCES AND NOTES

(1) Gasteiger, J. In *Physical Property Prediction*; Jochum, C., Hicks, M. G., Sunkel, J., Eds.; Springer: Heidelberg, 1988; pp 119–138.

(2) Gasteiger, J.; Hutchings, M. G.; Christoph, B.; Gann, L.; Hiller, C.; Löw, P.; Marsili, M.; Saller, H.; Yuki, K. A New Treatment of Chemical Reactivity. Development of EROS, an Expert System for Reaction Prediction and Synthesis Design. *Top. Curr. Chem.* **1987**, *137*, 19–73.

(3) Gasteiger, J.; Marsili, M.; Hutchings, M. G.; Saller, H.; Löw, P.; Rafeiner, K.; Röse, P. Models for the Representation of Knowledge about Chemical Reactions. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 467–476.

(4) Gasteiger, J.; Saller, H.; Löw, P. Elucidating Chemical Reactivity by Pattern Recognition Methods. *Anal. Chim. Acta* **1986**, *191*, 111–123.

(5) Zupan, J.; Gasteiger, J. Neural Networks: A New Method for Solving Chemical Problems or Just a Passing Phase?. *Anal. Chim. Acta* **1991**, *248*, 1–30.

(6) Kohonen, T. *Self-Organization and Associative Memory*, 2nd ed.; Springer: New York, 1988; pp 68–184.

(7) Schulz, K.-P.; Hofmann, P.; Gasteiger, J. In *Software-Entwicklung in der Chemie 2*; Gasteiger, J., Ed.; Springer: Berlin, 1988; pp 181–196.

(8) Albus, J. S. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). *Dyn. Syst., Meas. Control* **1975**, *97*, 220–227.

(9) Albus, J. S. Data Storage in the Cerebellar Model Articulation Controller (CMAC). *Dyn. Syst., Meas. Control* **1975**, *97*, 228–233.

(10) Albus, J. S. In *Brains, Behaviour, and Robotics*; BYTE Books; Petersborough: NH; 1981; pp 139–180.

(11) Albus, J. S. A Theory of Cerebellar Functions. *Math. Biosci.* **1971**, *10*, 25–61.

(12) Gasteiger, J.; Schulz, K.-P.; Kredler, C. The Analysis of the Reactivity of Single Bonds in Aliphatic Molecules by a Battery of Pattern Recognition Methods. *J. Chem. Inf. Comput. Sci.*, preceding paper in this issue.

(13) Gasteiger, J.; Marsili, M. Iterative Partial Equalization of Orbital Electronegativity-A Rapid Access to Atomic Charges. *Tetrahedron* **1980**, *37*, 3219–3228.

(14) Gasteiger, J.; Saller, H. Calculation of the Charge Distribution in Conjugated Systems by a Quantification of the Resonance Concept. *Angew. Chem.* **1985**, *97*, 699–701; *Angew. Chem., Int. Ed. Engl.* **1985**, *24*, 687–689.

(15) Hutchings, M. G.; Gasteiger, J. Residual Electronegativity-An Empirical Quantification of Polar Influences and Its Application to the Proton Affinity of Amines. *Tetrahedron Lett.* **1983**, *24*, 2541–2544.

(16) Gasteiger, J.; Hutchings, M. G. Quantification of Effective Polarisability. Applications to Studies of X-Ray Photoelectron Spectroscopy and Alkylamine Protonation. *J. Chem. Soc., Perkin Trans. 2* **1984**, 559–564.

(17) Gasteiger, J. Automatic Estimation of Heats of Atomization and Heats of Reaction. *Tetrahedron* **1979**, *35*, 1419–1426. Gasteiger, J.; Rafeiner, K. Unpublished results.

(18) Massart, D. L.; Vandeginste, B. G. M.; Deming, S. N.; Michotte, Y.; Kaufmann, L. *Chemometrics*: *A Textbook*; Elsevier, Amsterdam, 1988.

(19) Brent, R. P. In *Algorithms for Minimization without Calculating Derivatives*; Prentice-Hall: Englewood Cliffs, NJ, 1973; pp 116–167.