THE LEARNING MACHINE IN QUANTITATIVE CHEMICAL ANALYSIS Part 1. Anodic Stripping Voltammetry of Cadmium, Lead and Thallium

M. BOS* and G. JASINK

*Department of Chemical Technology, Twente University of Technology, Enschede
(The Netherlands)*

(Received 6th March 1978)

SUMMARY

The linear learning machine method was applied to the determination of cadmium, lead and thallium down to 10^{-8} M by anodic stripping voltammetry at a hanging mercury drop electrode. With a total of three trained multicategory classifiers, concentrations of Cd, Pb and Tl could be predicted with an accuracy of $\pm 10\%$. The classifiers were trained with the use of least-squares minimization. Numerical problems in the data matrix inversion were overcome by using singular value decomposition.

Pattern recognition has been applied in chemical analysis in various fields, i.e. infrared spectroscopy [1], mass spectrometry [2], nuclear magnetic resonance [3] and stationary electrode polarography [4]. Most of the applications described so far refer to qualitative analytical aspects such as identification problems, the presence or absence of a second component, multiplicity of peaks, etc. Some quantitative problems with a restricted number of classes were solved by the use of sets of binary classifiers e.g. in the number of carbon atoms in a given compound from its mass spectrum [5]. When a concentration has to be classified, this method becomes impractical because of the large number of classifiers needed if a reasonable accuracy is required over even a moderate concentration range.

Anodic stripping voltammetry (a.s.v.) is a very sensitive technique which is used extensively for determination of the heavy metal content of natural waters, body fluids, etc. Although the differential pulse mode makes this technique quite selective, cases with badly overlapping peaks remain in which classical evaluation of the results is impossible. Peak deconvolution could be used to resolve these problems, but the process is time-consuming and requires a knowledge of what to look for, otherwise serious errors will occur. Pattern recognition is fast — at least after training is complete — and requires no knowledge about the unknown samples other than that they should resemble samples of the training set. Moreover, in the fields where this type of analysis is being used, often only a go/no go decision is required from the result of the analysis.

Accordingly, the learning machine method developed by Kowalski et al. [6] where the classifier operating directly on the data produces the quantitative measure of interest, is the best to apply. As Tunnicliff and Wadsworth [7] have pointed out, numerical problems can arise in handling the data matrix. In the present work, singular value decomposition of the data matrix and a modified Ho—Kashyap algorithm [8] were used to circumvent this problem.

THEORY

A short review on the theory of linear learning machines is given here. Anodic stripping voltammograms, like any other set of measurements, can be represented as patterns $X_1, X_2, \dots, X_i, \dots, X_d$, where, for a.s.v., X_i denotes the current measured at point i on the voltage axis and d is the number of measurements taken along that axis. The vector X (pattern $X_1, X_2, \dots, X_d$) can be thought of as a point in the d -dimensional euclidian vector space E_d . A given phenomenon is associated with pattern X . This phenomenon belongs to one out of R classes. The learning machine problem is to find from a set of patterns with known classification (the training set) a classifier (or a set of classifiers) that correctly classifies an unknown pattern. The euclidian space E_d can be divided into R parts, each of which corresponds to one of the R classes. Classes are separated by so-called decision planes. These planes are defined implicitly by a set of functions $g_1(X), g_2(X), \dots, g_R(X)$ of the pattern vector X . The decision functions are chosen in such a manner that if X belongs to class i , then

$$g_i(X) > g_j(X) \text{ for } j = 1, 2, \dots, R \text{ with } j \neq i \quad (1)$$

If the functions $g_1(X), g_2(X), \dots, g_R(X)$ are known, the pattern X can be classified by determining the maximum value. A simple case is that of only two classes ($R = 2$), in which a single decision function suffices:

$$g(X) = g_1(X) - g_2(X) \quad (2)$$

if $g(X) > 0$ then X belongs to class 1, and if $g(X) < 0$ then X belongs to class 2. The decision surface can then be represented by

$$g(X) = 0 \quad (3)$$

In the linear learning machine, the decision plane is given by

$$g(X) = W_1X_1 + W_2X_2 + \dots + W_dX_d + W_{d+1} = 0 \quad (4)$$

This is a hyperplane with the dimension $(d - 1)$ separating hyperspace E_d into two parts.

During the training, the parameters $W_1, W_2, \dots, W_d, W_{d+1}$ are adjusted in such a way that the hyperplane $g(X) = 0$ separates the pattern points of the training set correctly. The hyperplane $g(X) = 0$ does not pass through the origin of the space E_d . As it is easier to work with a decision plane through the origin, a constant element c is added to the vector X :

$$Y = (Y_1, Y_2, \dots, Y_d, Y_{d+1}) = (X_1, X_2, \dots, X_d, c) = (X, c) \quad (5)$$

Then $g(X)$ can be replaced by

$$g(y) = W_1 Y_1 + W_2 Y_2 + \dots + W_d Y_d + W_{d+1} Y_{d+1} \quad (6)$$

which can be written as

$$g(Y) = (W \cdot Y) \quad (7)$$

In the following treatment, $n = d + 1$ and the vectors W and Y are vectors in euclidian space E_n . If the sign of the patterns belonging to class 2 is changed, eqns. (2) and (7) can be combined:

$$(W \cdot Y_j) > 0 \quad j = 1, 2, \dots, m \quad (8)$$

with m the number of patterns in the training set. This inequality can be replaced [8] by $(W \cdot Y_j) = b_j$ with $b_j > 0$ and $j = 1, 2, \dots, m$, which can be re-written as

$$AW = b \quad (9)$$

Here A is the data matrix ($m \times n$) of which the rows are constituted by the patterns $Y_j, j = 1, 2, \dots, m$.

Least-squares training

In general, $m > n$, so that the set of equations (9) is overdetermined and has no exact solution. If the training set is linearly separable, $b > 0$ (notation $b > 0$ means $b_j > 0, j = 1, 2, \dots, m$) can be determined with the Ho—Kashyap [8] algorithm in such a way that a value of W can be found which satisfies $A \cdot W > 0$. This algorithm minimizes the function

$$J(W, b) = \|Aw - b\|^2 \quad (10)$$

for w and for b , hence the name least-squares training. The algorithm is iterative with k the number of the iteration step being processed: for $b^{(0)} > 0$, with initial arbitrary values of b ,

$$W^{(k)} = A^+ b^{(k)} \quad (11)$$

In this equation A^+ is a generalized inverse of A :

$$e^{(k)} = A W^{(k)} - b^{(k)} \quad (12)$$

if $e_j^{(k)} > 0$, then $b_j^{(k+1)} = b_j^{(k)} + 2\rho e_j^{(k)}$ with $0 < \rho < 1$

$$e_j^{(k)} < 0, \text{ then } b_j^{(k+1)} = b_j^{(k)} \quad (13)$$

The sequence (11) through (13) is repeated until

$$A W^{(k)} > 0 \quad (14)$$

This algorithm has the following properties: if $e_i^{(k)} < 0$ for all $j, 1 < j < m$, then the training set is not linearly separable; if $e_j^{(k)} > 0$ for all $j, 1 < j < m$ and for $k = 1, 2, \dots, k$, and eqn. (14) is not yet satisfied, then there may be

a solution. If this solution exists, it will be found in a finite number of iterations. If it does not exist, then $e^{(k)}$ approaches a limiting vector.

Least-squares training for (nearly) dependent columns in data matrix

The calculation of the generalized inverse A^+ from the data matrix A requires special attention. If the columns of A are linearly independent, A^+ can be replaced by $A^+ = (A^t \cdot A)^{-1} \cdot A^t$

If A is ranked much lower than n , the matrix $(A^t \cdot A)$ is singular and A^+ cannot be calculated from this equation.

To restate the problem: the set of m equations with n unknowns of eqn. (9), $Aw = b$, must be solved for $m > n$. Matrix A has n columns each consisting of m elements. These columns of A can be represented as the (column) vectors $a_1, a_2, \dots, a_n$ in space E_m , which is the m -dimensional euclidian vector space. The vector b also lies in E_m . The vector W has n elements $(W_1, W_2, \dots, W_n)$ and can be represented in E_n .

The n columns of the matrix A span a subspace of E_m . This subspace, denoted as $R(A)$, contains all vectors that are a linear combination of the columns of A . For all arbitrary values of $\alpha_i, i = 1, 2, \dots, n$, the vector d defined as $d = \alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_n a_n$ lies in $R(A)$. The vector d defined as $d = W_1 a_1 + W_2 a_2 + \dots + W_n a_n$, therefore also lies in $R(A)$ for all possible values of $W_i, i = 1, 2, \dots, n$. Thus for all vectors w from E_n , d lies in $R(A)$. If the vector b from E_m (cf. eqn. 9) lies in $R(A)$, then there is a vector w from E_n for which $d = Aw$ coincides with b and eqn. (9) can be satisfied.

If the vector b is not contained in $R(A)$, then there is no vector w from E_n for which $d = Aw$ coincides with b . For all vectors w from E_n , there is then a difference vector: $e = d - b = Aw - b$, which is different from the null vector. The distance between d and b is the length of the difference vector:

$$\|d - b\| = \|e\| = \left(\sum_{j=1}^m e_j^2 \right)^{\frac{1}{2}} \quad (15)$$

For the solution w in E_n of $Aw = b$ satisfying the least-squares criterion, the distance between d and b is minimized:

$$\|e\|^2 = \|Aw - b\|^2 = J(w) \quad (16)$$

The distance between d and b is minimal when d is the normal projection of b onto the subspace $R(A)$. The normal projection of a point (b) from the space E_m onto the subspace $R(A)$ of E_m is a unique point of the subspace $R(A)$ and is denoted as $d(\text{proj})$. If the point $d(\text{proj})$ of $R(A)$ has been found, then vector w can be found, satisfying $Aw = d(\text{proj})$.

The matrix A contains n columns. If only r of these columns are linearly independent, then the other $(n - r)$ columns are linear combinations of the first r columns. In that case, the rank of the matrix A is r and the set of equations $Aw = 0$ has $(n - r)$ independent solutions $w_1, w_2, \dots, w_{n-r}$. If $w(\text{spec})$ is a solution of $Aw = d(\text{proj})$, the general solution of $Aw = d(\text{proj})$ becomes

$$\mathbf{w} = \mathbf{w}(\text{spec}) + \sum_{i=1}^{n-r} \alpha_i \mathbf{w}_i \quad (17)$$

with α_i arbitrary. For one of these solutions $\mathbf{w}$ of eqn. (17), $\|\mathbf{w}\|$ is minimal. This solution can be found with a pseudo-inverse [9, 10]. If the rank of the matrix A equals r , the subspace $R(A)$ of E_m can be spanned by only r columns of A . If the columns are aligned in such a way that $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ are linearly independent, then $\mathbf{a}_{r+1}, \mathbf{a}_{r+2}, \dots, \mathbf{a}_n$ are linearly dependent on $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ with $\mathbf{a}_{r+l} = \sum_{j=1}^r \alpha_j \cdot \mathbf{a}_{j,l}$ and $\mathbf{a}_n = \sum_{j=1}^r w_j \mathbf{a}_{j,n}$. The vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ now form a base of the subspace $R(A)$; the dimension of the subspace is r .

Singular value decomposition to determine the rank of the data matrix

To determine the orthonormal projection $d(\text{proj})$ of vector $\mathbf{b}$ from E_m onto $R(A)$, a base for $R(A)$ must be known as well as the rank of A or the dimension of $R(A)$. This problem can be solved by calculation of the pseudo-inverse of A based on singular value decomposition (SVD) [9–14]. The pseudo-inverse of the $m \times n$ matrix A is the $n \times m$ matrix A^+ with the property that

$$\mathbf{w} = A^+ \mathbf{b} \quad (18)$$

is a solution of $A\mathbf{w} = d(\text{proj})$, for which $\|\mathbf{w}\|$ is minimal if $A\mathbf{w} = d(\text{proj})$ has more than one solution. In the SVD, an arbitrary matrix A can be decomposed to 3 matrices with special properties:

$$A = U \Sigma V^t \quad (19)$$

in which A is the $m \times n$ arbitrary matrix ($m \geq n$); U is the $m \times n$ orthonormal matrix (for columns), i.e. $U^t U = I_n$; V is the $n \times n$ orthonormal matrix (for columns and rows), i.e. $V^t V = I_n$ and $V V^t = I_n$; and Σ is the $n \times n$ diagonal matrix with diagonal elements $\delta_1 \geq \delta_2 \geq \delta_3 \geq \dots > \delta_n \geq 0$, the other elements being zero. The elements δ_i of the diagonal matrix Σ are called the singular values of A .

The pseudo-inverse of matrix A is

$$A^+ = V \Sigma^+ U^t \quad (20)$$

in which Σ^+ is the $n \times n$ diagonal matrix with elements $\delta_1^+, \delta_2^+, \dots, \delta_n^+$ with $\delta_i^+ = 1/\delta_i$ for $\delta_i > 0$, and $\delta_i^+ = 0$ for $\delta_i = 0$ [9, 10].

In theory, the rank of A equals the number of the singular values greater than zero. If this number is r , then a base for the subspace $R(A)$ is formed by the first r columns of matrix U . These are the only columns of U that are of importance in eqns. (19) and (20); the other $(n - r)$ columns of U are unused because the corresponding $(n - r)$ elements of Σ are zero.

In practice, A can contain a large number of nearly dependent columns. In this case, Σ has a number of elements δ_i that are very small. These are the higher numbered δ_i values in the above $\delta_1 \dots \delta_n$ sequence. If the first q singular values are relatively much larger than the others, A^+ can be calculated

from eqn. (20) with Σ^* replaced by $\Sigma_{(q)}^*$, which is the $n \times n$ diagonal matrix with elements

$$\delta_1^* = 1/\delta_1, \delta_2^* = 1/\delta_2, \dots, \delta_q^* = 1/\delta_q \text{ and } \delta_{q+1} = \delta_{q+2} = \dots = \delta_r = \dots = \delta_n = 0 \quad (21)$$

In fact, matrix A has been replaced by matrix $A(q)$ with rank q . The base of space $R(A_{(q)})$ is then formed by the first q columns of U . In this way, the value of the "norm" of the difference $A - A(q)$

$$\|A - A(q)\| = \left(\sum_{i=q+1}^n \delta_i^2 \right)^{\frac{1}{2}} \quad (22)$$

is smallest for a given q value (see [10]). $Aw = b$ can then be solved by means of eqns. (20) and (21).

The solution w_q with q non-zero singular values

The classifier w that is required has to satisfy only $d = Aw > 0$. SVD gives $A = U\Sigma V^t$, of which the elements are

$$a_{ik} = \sum_{l=1}^n U_{il} V_{kl} \delta_l \quad (23)$$

Here a_{ik} is the i -th row and k -th column element of the matrix A . U_{il} and V_{kl} are the elements of U and V defined in the same manner. From eqn. (23):

$$d = Aw = \sum_{k=1}^n \delta_k (V_k \cdot w) u_k \quad (24)$$

where $(v_k \cdot w)$ represents the inner product of the k -th column of V and the vector w . This inner product is defined in E_n . From eqn. (24), it follows that d is a linear combination of the columns u_k of U . If the rank of A equals r , this can be seen from $\delta_{k=0}$ for $k > r$. Equation (24) then reduces to

$$d = \sum_{k=1}^r \delta_k (v_k \cdot w) u_k \quad (25)$$

For $w = A^+b$ (eqn. 18), an expression in the elements of U , V and Σ can also be found (cf. eqn. 20).

If the rank of A equals r , then $\delta_i^* = 1/\delta_i$ for $i \leq r$ and $\delta_i^* = 0$ for $i > r$. From eqns. (18), then

$$w = \sum_{i=1}^r (1/\delta_i) (u_i \cdot b) v_i \quad (26)$$

where $(u_i \cdot b)$ is the inner product of the i -th column of U and the vector b , defined in E_m . The vectors d (cf. eqn. 25) and w (cf. eqn. 26) correspond to $w = A^+b$ and $d = A.w$ calculated with the full rank- r pseudo-inverse A^+ .

If w is calculated with only q terms ($q < r$), then

$$\mathbf{w}_q = \sum_{i=1}^q (1/\delta_i) (\mathbf{u}_i \cdot \mathbf{b}) \mathbf{v}_i \quad (27)$$

This equation indicates that $\mathbf{w}_q$ is a linear combination of the columns $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q$ of V . V is orthonormal (cf. $V^t V = I_n$ above), and so for the columns of V , $(\mathbf{v}_k \cdot \mathbf{v}_i) = 1$ (for $k = i$) or 0 for $k \neq i$.

The subspace of E_n spanned by the vectors $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q)$ is the orthogonal complement of the subspace spanned by $(\mathbf{v}_{q+1}, \mathbf{v}_{q+2}, \dots, \mathbf{v}_n)$. A vector from the one subspace is orthogonal to all vectors from the other orthogonally complementary subspace. Thus $\mathbf{w}_q$ is orthogonal to $(\mathbf{v}_{q+1}, \mathbf{v}_{q+2}, \dots, \mathbf{v}_n)$; i.e. $(\mathbf{v}_k \cdot \mathbf{w}_q)$ is an arbitrary value for $k = 1, 2, \dots, q$, or is zero for $k = q + 1, q + 2, \dots, n$. Thus

$$\mathbf{d}(\mathbf{w}_q) = A\mathbf{w}_q = \sum_{k=1}^q \delta_k (\mathbf{v}_k \cdot \mathbf{w}_q) \mathbf{u}_k + \sum_{k=q+1}^r \delta_k \mathbf{0} \mathbf{u}_k \quad (28)$$

or

$$\mathbf{d}(\mathbf{w}_q) = \sum_{k=1}^q \delta_k (\mathbf{v}_k \cdot \mathbf{w}_q) \mathbf{u}_k$$

where $\mathbf{d}(\mathbf{w}_q)$ is the vector $\mathbf{d}$ corresponding to a $\mathbf{w}$ value calculated from q terms only. Equation (27) gives

$$(\mathbf{v}_k \cdot \mathbf{w}_q) = \left(\mathbf{v}_k \cdot \left(\sum_{i=1}^q \frac{1}{\delta_i} (\mathbf{u}_i \cdot \mathbf{b}) \mathbf{v}_i \right) \right) = \sum_{i=1}^q \frac{1}{\delta_i} (\mathbf{u}_i \cdot \mathbf{b}) (\mathbf{v}_k \cdot \mathbf{v}_i) \quad (29)$$

Combination of eqn. (29) with the term $(\mathbf{v}_k \cdot \mathbf{v}_i) = 1$ (for $k = i$) gives

$$(\mathbf{v}_k \cdot \mathbf{w}_q) = \frac{1}{\delta_k} (\mathbf{u}_k \cdot \mathbf{b}) \quad (30)$$

and eqns. (28) and (30) finally give

$$\mathbf{d}(\mathbf{w}_q) = \sum_{k=1}^q (\mathbf{u}_k \cdot \mathbf{b}) \mathbf{u}_k \quad (31)$$

Thus for the calculation of $\mathbf{d}(\mathbf{w}_q)$, it is not necessary to calculate $\mathbf{w}_q$.

For a classifier $\mathbf{w}$, it is sufficient to satisfy $\mathbf{d} = A\mathbf{w} > 0$. With eqn. (31), a vector $\mathbf{b}$ can be found for a specific choice of q in such a way that $\mathbf{d}(\mathbf{w}_q) > 0$, so that $A\mathbf{w}_q > 0$ is also then valid. With this q value and this $\mathbf{b}$ vector, $\mathbf{w}_q$ can be calculated from eqn. (27).

In the search for a vector $\mathbf{b}$ to satisfy $\mathbf{d}(\mathbf{w}_q) > 0$, a modified Ho—Kashyap algorithm can be used: for arbitrary positive initial values of $\mathbf{b}$, i.e. $\mathbf{b}^{(0)} > 0$,

$$\mathbf{d}^{(k)} = \sum_{i=1}^q (\mathbf{u}_i \cdot \mathbf{b}^{(k)}) \mathbf{u}_i \quad (32)$$

$$\mathbf{e}^{(k)} = \mathbf{d}^{(k)} - \mathbf{b}^{(k)} \quad (33)$$

$$\text{If } e_j^{(k)} > 0, \text{ then } b_j^{(k+1)} = b_j^{(k)} + 2\rho e_j^{(k)} \text{ with } 0 < \rho < 1 \quad (34)$$

$$\text{If } e_j^{(k)} \leq 0, \text{ then } b_j^{(k+1)} = b_j^{(k)}$$

The calculations based on the relationships (32)–(34) must be repeated until $\mathbf{d}^{(k)} > 0$.

The properties of this modified Ho–Kashyap training algorithm regarding termination in the case of a linearly inseparable training set are the same as for the original algorithm.

U is orthonormal, i.e. $U^*U = I_n$, thus for the U columns, $(\mathbf{u}_k, \mathbf{u}_i) = 1$ for $k = i$, or 0 for $k \neq i$. The subspace spanned by the columns $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_q$ is denoted as $R(U_q)$. The orthogonal projection of the vector $\mathbf{b}$ from E_m onto the subspace $R(U_q)$ of E_m is given by

$$\mathbf{b}' = \sum_{k=1}^q (\mathbf{u}_k, \mathbf{b}) \mathbf{u}_k \quad (35)$$

Equation (35) is identical with eqn. (31), and so $\mathbf{d}(\mathbf{w}_q)$ is the projection of $\mathbf{b}$ onto $R(U_q)$.

Selection of the number of non-zero singular values, q

If the measurements comprising the data matrix A were without experimental error, the dimensionality of the subspace $R(A)$ would be low, which can be denoted by $q(\text{ideal})$. Young and Calvert [15] called this the intrinsic dimensionality of the training set. Because of random errors during measurements, the mathematical rank of the matrix A equals r . Thus $q(\text{ideal}) < r \leq n$.

The low numbered columns of matrix U form the most important vectors spanning $R(A)$. It is likely that the influence of the phenomena to be classified on the measurement will exceed the influence of random errors. Therefore the best choice for q will involve including the first q columns of U for which $\mathbf{d}(\mathbf{w}_q) > 0$ can be found.

Selection of the right-hand vector, $\mathbf{b}$

If a binary classifier is required, the vector $\mathbf{b}$ can be initialized with a set of arbitrary positive values. The modified Ho–Kashyap training algorithm (eqns. 32–34) can then be used to adjust this $\mathbf{b}$ vector until $\mathbf{d}^{(k)} > 0$; $\mathbf{w}_q$ can then be calculated.

If a multicategory classifier is required, the vector $\mathbf{b}$ is initialized with a measure of the property of the patterns. In the a.s.v. experiments, this can be done with a concentration measure for the compound of interest. Then $\mathbf{w}$ must be calculated directly from eqn. (27) with the lowest possible q value, giving a small value for

$$\|\mathbf{e}\| = \|\mathbf{Aw} - \mathbf{b}\| \quad (36)$$

EXPERIMENTAL

Chemicals

The supporting electrolyte used was 10^{-2} M potassium chloride (Merck, Suprapur) and 10^{-4} M hydrochloric acid (Merck, Suprapur). Ultrapure nitrogen was passed through an acidic chromium(II) solution to remove the last traces of oxygen down to 1 ppm [16], and then through an all-glass tubing system to the cell to remove oxygen from the solution; purging was continued for 15 min. The mercury for the hanging mercury drop electrode (HMDE) was purified by washing it with nitric acid and vacuum-distilling it twice. The 10^{-6} – 10^{-8} M solutions of lead, thallium and cadmium were prepared daily from 0.1 M stock solutions of the nitrates (Merck, reagent grade).

Apparatus

A PAR Model 174 polarograph was connected to a PDP-11/10 (Digital Equipment Corp.) online computer. The computer samples the cell current available at the recorder terminals of the PAR 174 with a 12-bit A/D-converter (range ± 5 V) contained in the Laboratory Peripheral System (Digital Equipment Corp.).

The electrodes were a HMDE (Radiometer P958 b), a saturated calomel reference electrode, and a platinum wire auxiliary electrode; the auxiliary electrode was connected to the measuring vessel by an agar–potassium chloride salt bridge. A Radiometer type SMP-1 stirring motor was used with a glass stirring rod. The cell was thermostatted to 20 ± 0.5 °C.

All glassware was degreased with dichromate/sulphuric acid mixture, and washed with (1 + 1) nitric acid and doubly distilled water. When not in use, the glassware was kept in doubly distilled water.

Procedure

Measurements were done in the differential pulse mode at a scan rate of 5 mV s^{-1} , with a measuring interval of 0.5 s and a pulse height of 25 mV. A typical run was performed as follows. Purified nitrogen was passed through the solution for 15 min. The metals were then concentrated on the HMDE by applying -0.9 V to the cell, with stirring, for 3 min. Stirring was then stopped, and after 30 s the mode was switched to differential pulse; after another 30 s, the anodic scan was started. The PDP-11/10 automatically senses the start of the scan and with a delay of 180 mV acquires the current data at 480-ms intervals in the range of -0.72 to -0.22 V.

Computer programs

Data acquisition program. The program for data acquisition is straightforward and runs on the PDP-11/10 online computer. The data acquired during the runs can be stored on disks, together with pertinent information. The information stored can be typed and punched in ASCII format for transference to another computer.

Training program. The training of the learning machine was done on a DEC-10 computer. Input to the training program was the ASCII paper-tape containing the results of the measurements in the form of integers between 2048 and 4096 (A/D-converter format for voltages between 0 and +5 V). These values are converted to real format and adjusted in accordance with the position of the current range switch on the PAR 174 used during the experiment. The dynamic range of the patterns becomes 42–420000. For the training of binary classifiers, the concentration range for a specific compound is divided into a number of intervals, and classifiers are trained that classify the pattern on the correct sides of the boundaries between intervals, by means of the singular value decomposition and modified Ho–Kashyap algorithm described above.

The singular value decomposition was done with the use of a subroutine from the Numerical Algorithms Group (Oxford, England). This routine is based on the method of Golub and Reinsch [17]. In calculations of the multicategory classifiers, the elements b_j of the right-hand vector are given the value of the concentrations of the metal ion of interest as the initial value. Matrix U is not adjusted in this case. In fact, the program for this case is a simplification of the former one in which calculation is stopped after the zero-th iteration of the modified Ho–Kashyap algorithm.

Program for evaluation of results. To test the performance of the multicategory classifiers, the products $(w \cdot y)$ must be calculated. The last element of the patterns is a constant factor c ; thus the last term from $(w \cdot y)$ is $w_n c$.

To simplify the calculations and the presentation this last term is omitted in the table entries, but is indicated separately. Under normal conditions, a.s.v. results for a mixture equal the sum of the results for the separate compounds. If $yp = (p, c)$ is a pattern recorded for metal ion P, and $yq = (q, c)$ is a pattern recorded for metal ion Q, then a mixture of P and Q gives a pattern:

$$y(p, q) = (p + q, c) \quad (37)$$

If the last element of the weight vector w is w_n then:

$$(w \cdot yp) = (w \cdot (p, c)) = (\bar{w} \cdot p) + w_n c \quad (38)$$

$$(w \cdot yq) = (w \cdot (q, c)) = (\bar{w} \cdot q) + w_n c \quad (39)$$

$$(w \cdot yp, q) = (\bar{w} \cdot (p + q)) + w_n c = (\bar{w} \cdot p) + (\bar{w} \cdot q) + w_n c \quad (40)$$

Thus, classification of a mixture of P and Q can be simulated by calculating $(\bar{w} \cdot p)$ and $(\bar{w} \cdot q)$, determining the sum and adding $w_n c$. In this way, presentation of the results can be restricted to the value of the $(\bar{w} \cdot p)$ for single-component samples of the various metal ions. Moreover, the value of $w_n c$ turned out to be very small in comparison to the other terms from $(\bar{w} \cdot p)$ and could be neglected.

RESULTS

The patterns recorded by the computer contained 209 points. From these points, 49 were selected at equally spaced voltage intervals of 7.2 mV between -0.642 and -0.297 V. The training set consisted of 7 patterns for each of the three metal ions (Pb, Cd and Tl). The concentrations for these 7 patterns were 10^{-8} , 2.5×10^{-8} , 5×10^{-8} , 10^{-7} , 2.5×10^{-7} , 5×10^{-7} and 10^{-6} M; the codes used to represent these concentrations were 10, 25, 50, 100, 250, 500 and 1000, respectively. The constant c for the 50th element of the pattern was taken as 10000.

Table 1 shows the peak currents in mA for the members of the training set. Peak positions were -0.380 ± 0.005 V for Pb, -0.580 ± 0.005 V for Cd, and -0.445 ± 0.005 V for Tl.

Separate classifiers were calculated for lead, thallium and cadmium by using 2, 3, 4, 5 and 16 non-zero singular values and the 3×7 training set. The calculations were carried out in the following sequence: singular value decomposition of the data matrix formed by the training set, selection of the required number of non-zero singular values, and calculation of the weight vector with the use of the coded concentrations as the right-hand vector b .

Table 2 shows the recognition performance of the weight vectors for lead, cadmium and thallium obtained with 4 non-zero singular values. In these calculations, patterns y belonging to the training set were classified by calculation of $w_i \cdot y_i$, omitting the term $w_{50} \cdot 10000$. The results are listed in the column 'Recognition with shift 0 mV'. The same calculations were performed with patterns from the computer measurements comprising the training set, in which 49 data points were selected which were shifted a fixed amount along the voltage axis with regard to the original 49 points used for training. These results are given in the columns 'Recognition with shift -5 , -10 , $+5$ and $+10$ mV'.

Patterns simulated by linear interpolation of the current values of the patterns comprising the training set were used to test the capability for continuous classification of the weight vectors. Table 3 shows the results of the classification of these simulated patterns.

TABLE 1

Peak currents during anodic stripping voltammetry of Pb, Cd and Tl

Concn. (M)	Peak current (mA)		
	Pb	Cd	Tl
10^{-8}	0.094	0.060	0.041
2.5×10^{-8}	0.240	0.184	0.090
5.0×10^{-8}	0.410	0.365	0.185
10^{-7}	0.847	0.790	0.353
2.5×10^{-7}	2.13	2.07	0.938
5.0×10^{-7}	4.35	4.11	1.853
10^{-6}	9.10	8.36	3.71

TABLE 2

Recognition performance of weight vectors for Cd, Pb and Tl

Sample recomposition ($\times 10^{-3}$ M)		Recognition with shift 0 mV			Recognition with shift -5 mV			Recognition with shift -10 mV			Recognition with shift +5 mV			Recognition with shift +10 mV			
Cd	Tl	Pb	Cd	Tl	Pb	Cd	Tl	Pb	Cd	Tl	Pb	Cd	Tl	Pb	Cd	Tl	Pb
—	—	10	0.3	1.4	10.6	0.3	1.4	10.6	0.3	1.4	10.6	0.3	1.4	10.6	0.3	1.4	10.6
—	—	25	0.2	1.3	26.8	0.3	1.2	27.5	0.2	2.0	27.5	0.2	1.4	25.6	0.2	2.1	24.0
—	—	50	0.3	0.5	46.3	0.4	0.8	47.4	0.3	2.4	47.2	0.3	1.0	44.6	0.2	2.3	41.8
—	—	100	0.8	0.4	95.2	0.9	1.3	96.6	0.8	4.4	97.0	0.7	1.2	91.7	0.6	4.1	86.2
—	—	250	0.4	-0.9	237.6	0.6	-0.3	242.4	0.6	5.6	244.3	0.2	2.3	227.2	-0.3	10.3	210.3
—	—	500	0.1	0.3	501.2	-0.3	19.9	494.8	-1.1	53.3	478.0	-0.1	-6.6	493.2	-0.7	-0.3	474.2
—	—	1000	-0.6	-0.2	1002.2	0.6	-5.9	1035.4	0.8	12.2	1050.3	-1.9	25.1	950.5	-3.4	55.8	881.9
10	—	—	7.2	1.4	1.9	7.1	1.3	2.0	6.9	1.3	2.1	7.2	1.4	1.8	7.0	1.7	1.7
25	—	—	22.0	1.5	1.8	22.0	0.9	2.0	21.3	0.6	2.1	21.5	1.0	1.5	21.0	2.7	1.3
50	—	—	43.8	1.1	2.9	43.1	0.3	3.2	41.6	-0.1	3.4	43.7	2.1	2.5	42.9	3.5	2.0
100	—	—	94.7	0.9	3.4	94.4	-0.9	4.0	92.1	-2.5	4.5	93.8	3.6	2.5	91.1	7.2	1.5
250	—	—	248.8	0.3	2.5	246.3	-5.1	4.0	239.1	-8.6	5.1	245.3	7.2	0.5	239.2	15.4	-1.6
500	—	—	489.8	-10.1	-0.2	475.6	-17.8	2.1	458.4	-23.7	4.0	492.8	0.8	-3.2	483.6	15.1	-6.7
1000	—	—	1005.8	4.5	-1.6	1001.6	-18.6	5.0	973.9	-35.3	9.9	987.7	33.4	-8.8	980.6	60.4	-17.7
—	10	—	0.9	11.2	3.3	0.9	11.5	3.1	0.9	11.7	2.9	0.8	10.9	3.5	0.8	10.5	3.7
—	25	—	0.8	25.7	1.3	1.0	26.0	0.7	1.2	26.2	0.2	0.7	26.2	1.8	0.6	24.5	2.5
—	50	—	0.1	49.8	2.0	0.3	51.2	0.5	0.6	52.0	-0.8	-0.2	48.0	3.8	-0.3	45.9	5.4
—	100	—	0.3	94.8	2.5	0.8	97.5	-0.5	1.4	99.1	-2.9	-0.0	91.7	5.8	-0.3	87.7	9.1
—	250	—	0.3	253.9	1.1	1.5	260.6	-6.4	3.3	264.5	-13.1	-0.7	245.9	10.1	-1.6	235.3	19.3
—	500	—	0.1	500.0	-0.8	2.9	510.9	-15.9	6.2	518.1	-28.7	-1.9	482.6	16.6	-3.4	462.3	34.5
—	1000	—	-0.7	999.1	-0.9	4.5	1025.8	-31.2	11.4	1041.0	-59.3	-4.7	986.5	33.6	-7.9	925.3	72.0

TABLE 3

Prediction performance of the weight vectors on interpolated numbers of training set

Sample composition ^a	Prediction			Sample composition ^b			Prediction			Sample composition ^c			Prediction		
	Cd	Tl	Pb	Tl	Pb	Cd	Tl	Pb	Cd	Tl	Pb	Pb	Cd	Tl	Pb
14.5	11.6	1.4	1.9	14.5		0.9	15.6	2.7	0.2	14.5		15.5	0.2	1.3	15.5
19	16.1	1.4	1.8	19		0.8	19.9	2.1	0.3	19		20.2	0.3	1.3	20.2
32.5	28.6	1.4	2.1	32.5		0.6	32.9	1.5	0.2	32.5		32.7	0.2	1.0	32.7
40	35.1	1.2	2.5	40		0.4	40.2	1.7	0.3	40		38.5	0.3	0.8	38.5
65	59.1	1.0	3.0	65		0.2	63.3	2.1	0.5	65		61.0	0.5	0.5	61.0
80	74.4	1.0	3.2	80		0.2	76.8	2.3	0.6	80		75.7	0.6	0.4	75.7
100	94.7	0.9	3.4	100		0.3	94.8	2.5	0.8	100		95.2	0.8	0.4	95.2
145	140.9	0.7	3.1	145		0.3	142.5	2.1	0.7	145		137.9	0.7	-0.0	137.9
190	187.1	0.5	2.8	190		0.3	190.3	1.7	0.6	190		180.7	0.6	-0.4	180.7
325	321.1	-2.8	1.6	325		0.2	327.7	0.5	0.3	325		316.7	0.3	-0.6	316.7
400	393.4	-6.0	0.8	400		0.2	401.6	-0.0	0.2	400		395.8	0.2	-0.2	395.8
650	644.6	-5.7	-0.6	650		-0.1	649.8	-0.8	-0.1	650		651.5	-0.1	0.1	651.5
800	799.4	-1.3	-1.0	800		-0.4	799.5	-0.9	-0.3	800		801.8	-0.3	-0.0	801.8
1000	1005.8	4.5	-1.6	1000		-0.7	999.1	-0.9	-0.1	1000		1002.2	-0.1	-0.2	1002.2

^a $\times 10^{-9}$ M; Tl, Pb absent. ^b $\times 10^{-9}$ M; Cd, Pb absent. ^c $\times 10^{-9}$ M; Cd, Tl absent.

The results of the classification of completely independent real samples of Cd—Pb—Tl mixtures are given in Table 4.

CONCLUSIONS

As can be seen from Table 4 the quantitative predictive ability of the multi-category classifiers for lead, cadmium and thallium is quite good. The determinations are accurate to $\pm 10\%$ over the concentration range 10^{-6} – 10^{-8} M, even in cases where the presence of a component cannot be detected visually in the anodic stripping voltammograms (Fig. 1). The method is rather insensitive to shifts of the patterns along the voltage axis such as may be caused by drift of the reference electrode (see Table 2).

The number of four singular values used in the calculations proved to be optimum. With a lower number, all results were poor, whereas the use of higher numbers of singular values decreased the performance in the recognition test, especially when a shift was applied.

TABLE 4

Prediction performance of the weight vectors on separate samples of mixtures of Cd, Pb and Tl

Sample composition ($\times 10^{-9}$ M)			Prediction		
Cd	Tl	Pb	Cd	Tl	Pb
25	25	25	22.2	25.1	22.9
25	25	25	22.5	25.2	23.7
1000	1000	1000	1044.2	1083.6	1044.7
50	1000	50	48.1	1051.8	65.4
50	1000	50	48.4	1051.4	59.1
100			92.4	0.1	1.7
100			90.8	-0.5	1.7
			0.4	1.8	2.3
			0.6	2.4	2.6
			0.6	1.6	3.0
		10	0.4	1.2	11.3
		100	0.7	0.6	93.7
	100		1.1	99.0	-2.9
	W50 C		0.3	0.1	0.2

Fig. 1. A.s.v. of 10^{-6} M Tl, 5×10^{-8} M Cd and 5×10^{-8} M Pb.

The authors are indebted to J. Nieukoop for preparing the manuscript and to Prof. E. A. M. F. Dahmen for his interest.

REFERENCES

- 1 B. R. Kowalski, P. C. Jurs, T. L. Isenhour and C. N. Reilley, *Anal. Chem.*, **41** (1969) 1945.
- 2 S. R. Lowry, T. L. Isenhour, J. B. Justice, F. W. McLafferty, H. E. Dayringer and R. Venkataraghavan, *Anal. Chem.*, **49** (1977) 1720.
- 3 C. L. Wilkins and T. R. Brunner, *Anal. Chem.*, **49** (1977) 2136.
- 4 Q. V. Thomas, R. A. De Palma and S. P. Perone, *Anal. Chem.*, **49** (1977) 1376.
- 5 P. C. Jurs, B. R. Kowalski and T. L. Isenhour, *Anal. Chem.*, **41** (1969) 21.
- 6 B. R. Kowalski, P. C. Jurs, T. L. Isenhour and C. N. Reilley, *Anal. Chem.*, **41** (1969) 695.
- 7 D. D. Tunnicliff and P. A. Wadsworth, *Anal. Chem.*, **45** (1973) 12.
- 8 Y. C. Ho and R. L. Kashyap, *IEEE Trans. Electron. Comput.*, EC-14 no. 5, (1965) 683.
- 9 A. Albert, *Regression and the Moore—Penrose Pseudo-inverse*, Academic Press, New York, 1972.
- 10 A. Ben Israel and T. N. E. Greville, *Generalized Inverses; Theory and Applications*, Wiley, New York, 1974.
- 11 P. A. Businger and G. H. Golub, *Comm. ACM* **12** (1969) 564.
- 12 G. H. Golub and W. Kahan, *SIAM J. Numer. Anal.*, **2** (1965) 205.
- 13 S. Söderström, *SIAM J. Numer. Anal.*, **11** (1974) 61.
- 14 J. H. Wilkinson and C. Reinsch, *Handbook for Automatic Computation*, Vol. II, Linear Algebra, Springer Verlag, Berlin, 1971, p. 134—151.
- 15 T. Y. Young and T. W. Calvert, *Classification, Estimation and Pattern Recognition*, Elsevier, New York, 1973.
- 16 J. Sinko and J. Doležal, *J. Electroanal. Chem.*, **25** (1970) 53.
- 17 G. H. Golub and C. Reinsch, *Numer. Math.*, **14** (1970) 403.