JOURNAL OF COMPUTATIONAL PHYSICS 30, 250–258 (1979)

An Algorithm for Finding the Distribution of Maximal Entropy*

N. AGMON,† Y. ALHASSID,† AND R. D. LEVINE†

Department of Chemistry; Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Received October 28, 1977; revised April 21, 1978

An algorithm for determining the distribution of maximal entropy subject to constraints is presented. The method provides an alternative to the conventional procedure which requires the numerical solution of a set of implicit nonlinear equations for the Lagrange multipliers. Here they are determined by seeking a minimum of a concave function, a procedure which readily lends itself to computational work. The program also incorporates two preliminary stages. The first verifies that the constraints are linearly independent and the second checks that a feasible solution exists.

1. INTRODUCTION

In applications of probability theory to the physical sciences [1] one is often faced with the problem of determining a distribution consistent with a given set of average values. For n distinct states one thus seeks a vector x (components x_i, $x_i \geqslant 0$, $i = 1,..., n$), characterized by

$$\sum_{i=1}^{n} x_i = 1, \tag{1}$$

$$\sum_{i=1}^{n} A_{ri} x_i = b_r, \qquad r = 1,..., m. \tag{2}$$

Here Eq. (1) is the normalization condition and (2) defines b_r as the average value of the property A_r, whose magnitude in the ith state is A_{ri}. Equations (1) and (2) represent $m + 1$ constraints on the vector x and hence, if $m < n - 1$, do not suffice to provide a unique characterization. The principle of maximal entropy [1] provides that when $m < n - 1$, the probability assignment be made by the additional condition that the entropy, $S[x]$ (or missing information [1, 2]) of the distribution,

$$S[x] = -\sum_{i=1}^{n} x_i \ln x_i, \tag{3}$$

* Work supported by the Air Force Office of Scientific Research under grant AFOSR 77-3135.

† Permanent address: Department of Physical Chemistry, The Hebrew University, Jerusalem, Israel.

0021-9991/79/020250-09$02.00/0

be maximal. The method of Lagrange parameters [1] shows that the particular distribution x which is of maximal entropy is of the form

$$p_i = \exp\left(-\lambda_0 - \sum_{r=1}^{m} \lambda_r A_{ri}\right). \tag{4}$$

The $m + 1$ Lagrange parameters $(\lambda_0, \lambda_1, ..., \lambda_m)$ in (4) are to be determined by the $m + 1$ conditions that the distribution be normalized (Eq. (1))

$$Z(\lambda_1, ..., \lambda_m) = \exp(\lambda_0) = \sum_i \exp\left(-\sum_{r=1}^{m} \lambda_r A_{ri}\right) \tag{5}$$

and satisfy the m additional constraints (Eq. (2))

$$\sum_i A_{ri} p_i = b_r \tag{6}$$

which can be written in matrix notation $Ap = b$, or, using (5), as

$$\sum_i (A_{ri} - b_r) \exp\left(-\sum_{s=1}^{m} \lambda_s A_{si}\right) = 0, \qquad r = 1, ..., m. \tag{7}$$

To simplify the subsequent manipulations we now define

$$B_{ri} = A_{ri} - b_r \tag{8}$$

so that, in matrix notation, Eq. (7) reads $Bp = 0$. It is also important to note that using the B_r's as the constraints leaves the magnitude of the m Lagrange parameters $(\lambda_1, ..., \lambda_m)$ unchanged, and that the only change is in the magnitude of λ_0, i.e.,

$$\sum_{s=1}^{m} \lambda_s A_{si} + \lambda_0 = \sum_{s=1}^{m} \lambda_s B_{si} + \lambda_0', \qquad \lambda_0' = \lambda_0 + \sum_{s=1}^{m} \lambda_s b_s .$$

Hence, one can rewrite (7) as

$$\sum_i B_{ri} \exp\left(-\sum_{s=1}^{m} \lambda_s B_{si}\right) = 0, \qquad r = 1, ..., m. \tag{7'}$$

The m equations (7) for the Lagrange parameters are implicit and nonlinear. Even for $m = 1$, a numerical procedure is required (except if A_{1i} has a particularly pleasing structure) and increasingly often one requires a solution for $m > 1$, [3]. The purpose of this paper is to document an efficient algorithm for the determination of the Lagrange parameters. To avoid the need to solve Eq. (7), the approach recasts the problem of determining the Lagrange parameters as a variational problem. A "potential" function which is concave for any trial set of Lagrange parameters [4] is intro-

duced. The values of the parameters are determined as the set which minimizes the potential.

Section 2 is a discussion of the conditions on the avarage values which insure that a unique solution for the Lagrange parameters is indeed feasible. The method of solution is introduced in Section 3 and the actual algorithm is described in Section 4. An example is provided. A flow chart and the actual program are available upon request.

2. A Feasible Solution

To obtain a distribution of maximal entropy with a unique set of Lagrange parameters it is necessary to restrict the matrix A and the vector b (cf. Eq. (6)) as follows.

2.1. *Linear Independence*

The rows of the matrix A need be linearly independent (i.e., A should be of maximal rank). This not only insures that $Ap = b$ has a solution but also implies that the set of Lagrange parameters is unique, i.e., that $\ln p_i$ can be resolved in only one way as

$$\ln p_i = -\lambda_0 - \sum_{r=1}^{m} \lambda_r A_{ri} . \tag{9}$$

In other words, if there exists a vector $c \neq 0$ such that, for every i, $\sum_{r=0}^{m} c_r A_{ri} = 0$ (where we defined $A_{0i} = 1$), we may add $\sum_r c_r A_{ri}$ to the left-hand side of (9) leading to a new set of Lagrange parameters, i.e., $\lambda_r + c_r$. When the rows of A are not linearly independent one can always eliminate one or more constraints until the resulting set of Lagrange parameters is unique.

In practice, almost linear dependence amongst the rows of A is also objectionable. We therefore orthogonalize the rows of A by the Gram–Schmidt procedure. A new matrix, say A',

$$A' = QA, \tag{10}$$

is thereby generated, where Q is a regular matrix. Given the set μ_r, $r = 0,\ldots,m$ of Lagrange parameters for A' one readily verifies that the original set of Lagrange parameters is given by

$$\lambda = \mu Q. \tag{11}$$

2.2. *A Feasible Solution*

Even when rank $A = m + 1$ there may still not be a solution to $Ax = b$ which is a probability vector, i.e., which satisfies $x > 0$. This condition of nonnegativity of probability imposes additional restrictions on the components of the vector b. For one constraint this restriction is seen to be [4] the inequality

$$\min_{i=1,\ldots,n} \{A_{1i}\} < b_1 < \max_{i=1,\ldots,n} \{A_{1i}\} \tag{12}$$

which is evident also on intuitive grounds (an average must be between its upper and lower bounds). If one of the inequalities is replaced by an equality one would have $|\lambda_1| \to \infty$. If both are replaced by equalities one would have that A_1 is a multiple of $A_0 \equiv (1,\ldots,1)$, in contradiction to the assumed linear independence. Equation (12) can be easily generalized to more than one constraint [4], yielding the inequality

$$\min_{i=1,\ldots,n}\left\{\sum_{r=1}^{m} c_r A_{ri}\right\} < \sum_{r=1}^{m} c_r b_r < \max_{i=1,\ldots,n}\left\{\sum_{r=1}^{m} c_r A_{ri}\right\} \tag{13}$$

for any vector $c \neq 0$ in R^m. Again equality on one side means that some of the λ_r's tend to infinity and equaltiy on both sides means that the vectors A_r are linearly dependent. Thus (13) is a necessary and sufficient condition for linear independence and for the existence of a feasible solution for (4). When a feasible solution exists, the convecity of S assures that a unique maximum under the constraints (4) exists [5]. Therefore (13) is a necessary and sufficient condition for the existence of a maximal entropy distribution p (with finite λ_r's).

How does one verify that (13) holds for any $c \neq 0$? From a practical point of view it seems better to break the problem into two parts: First perform a Gram–Schmidt orthogonalization thus verifying that the A_r's are linearly independent, and then check the existence of a feasible solution to (4), using phase I of the modified simplex method of linear programming [6], for which computer programs are available.

3. The Lagrange Parameters

Given that a feasible solution exists we turn next to the determination of the Lagrange parameters. Equations (7) are a set of coupled nonlinear equations which define the λ_r's implicitly. They may be solved by numerically searching for a zero of $Ap - b$. In practice, when more than two constraints are present one often obtains effective zeros for which the resulting p's are quite removed from the correct solution. In principle, such a brute-force method fails to invoke the special character of p as the distribution of maximal entropy.

The method here proposed is based on the following:

LEMMA. *Let $\Omega \subset R^m$ be a simply connected domain. Let f: $\Omega \to R^m$ be a continuously differentiable (vectorial) function. Denote its Jacobian by M, that is, $M_{ij} \equiv \partial f_i/\partial x_j$, and suppose it is a symmetric positive definite matrix. The problem of solving the set of nonlinear equations $f(x) = 0$ is equivalent to finding a minimum of a concave scalar potential function F.*

Proof. The properties of Ω and f and the symmetry of M assure that f is a conservative vector field in R^m. Therefore, there exists a potential function F: $R^m \to R$ such that $f_i = \partial F/\partial x_i$. Because $M_{ij} = \partial^2 F/\partial x_i \partial x_j$ is the Hessian of F, its definiteness is sufficient for F to be strictly concave. Suppose x_0 is a solution of $f(x) = 0$, then $\nabla F|_{x_0} = 0$, which proves that x_0 is a unique global minimum of F. Q.E.D.

To employ the lemma, we construct a potential function F, as follows. Consider a trial distribution

$$p_i{}^t = \exp\left[-\sum_{r=1}^{m} \lambda_r{}^t B_{ri}\right] \Big/ Z(\lambda^t), \tag{14}$$

where B is given by (8) and

$$Z(\lambda^t) = \sum_{i=1}^{n} \exp\left[-\sum_{r=1}^{m} \lambda_r{}^t B_{ri}\right]. \tag{15}$$

By construction $Z(\lambda^t)$ is a function of the m trial Lagrange parameters $\lambda_1{}^t,\ldots, \lambda_m{}^t$. We now take the function $f(\lambda^t)$ of the lemma ($x \equiv \lambda^t$) to be Bp^t. The set of λ^t's that satisfies $f(\lambda^t) = 0$ is clearly the set that is obtained by solving (7'). By direct differentiation one verifies that the potential for the problem is

$$F = \ln Z(\lambda_1{}^t,\ldots, \lambda_m{}^t). \tag{16}$$

Here $Z(\lambda^t)$ is the function of the m Lagrange parameters defined in Eq. (15).

The practical gain in going from Eq. (7) to (7') should now be obvious. Using the A_r's as constraints, F would be $\ln Z(\lambda^t) + \sum_{r=1}^{m} \lambda_r{}^t b_r$ with Z given by (5). Defining B requires $m \cdot n$ subtractions at the beginning of the calculation but saves m^2 operations each time F is calculated and m operations each time ∇F is calculated. Because we perform many function calculations during each iteration, this results in a net reduction in computer time.

We have previously proven [4] that F is strictly concave whenever the constraints are not linearly dependent (if there is a direction in R^m along which the constraints are linearly dependent, F would be constant along that direction). Whenever there is a feasible solution to (4) (whenever (13) holds) F has (by the lemma) a unique minimum at the point λ which solves (15), so that the problem of solving (15) is converted to finding a minimum for (16).

Beside being a concave function of the trial Lagrange parameters and hence a conveneient computational tool, F admits of a physical interpretation. First of all

$$\min_{\lambda^t \in R^m} F(\lambda^t) = S[p] \tag{17}$$

which also shows that F is an upper bound to the entropy of a distribution which is consistent with the same set of constraints [4]. Suppose that instead of the averages b we are given an "experimental distribution" q, to which we want to fit a trial distribution p^t (cf. (14)), then one has to minimize the function

$$W = \sum_{i=1}^{n} q_i \ln q_i/p_i{}^t. \tag{18}$$

An interpretation of W as a work function has been previously given [7]. The problem of minimizing W is strictly equivalent to minimizing F because these two functions differ by the constant $S[q]$.

4. The Computer Program

4.1. *Purpose*

Given an $m \cdot n$ matrix A and an m-dimensional vector b, we find λ which minimizes F (cf. (16)), after satisfying the consistency checks. The program can also find the minimum of W (cf. (18)) when q is given instead of b. In this case, b is calculated from q according to $b = Aq$. In addition the program calculates the maximal entropy distribution $p(\lambda)$, the partition function Z (cf. (5)), $\ln Z$, $S[p]$, and the "corrlation matrix" M (the Hessian of F).

The same computations can also be performed given a prior distribution p^o. Equations (4) and (7) are then replaced by

$$Z = \sum_i p_i^o \exp\left[-\sum_r \lambda_r A_{ri}\right], \tag{19}$$

$$p_i = p_i^o \exp\left[-\sum_r \lambda_r A_{ri}\right] \Big/ Z. \tag{20}$$

The program consists of three main stages: Checking for linear independence, verifying the existence of a feasible solution, and solving for λ.

4.2. *Linear Independence*

A Gram–Schmidt orthogonalization of A is performed. At each stage of the procedure the angle between the vector to be orthogonalized and its projection on the subspace of the vectors which are already orthogonal is calculated. This angle serves as a criterion for linear independence. If it is effectively zero, the constraints are linearly dependent and execution of the run is stopped. If it is smaller than some parameter, the constraints are "almost linearly dependent" and a warning is issued. The execution then proceeds and the matrix Q (cf. 10)) is stored for performing the reverse transformation (11).

4.3. *A Feasible Solution*

The existence of a feasible solution to (2) is verified using the modified simplex method of linear programming [6]. If no feasible solution exists, the execution of the present run is stopped and data for the next example is read.

4.4. *Solving for the Lagrange Parameters*

This is done by a modified Newton method, as follows:

(1) The (orthogonalized) matrix A is replaced by B (cf. (14) and (15)), this is done for the practical reason explained above.

(2) Because of the concavity of F the iterations converge for any initial guess fo λ^t. However, three possibilities are given to the user: (i) $\lambda^t = 0$; (ii) λ^t is equal to λ of the previous run. (In case there was no previous run, it is automatically set to zero.) This option is recommended in the case of a series of runs where the probability

distribution (and therefore λ) develop continuously as a function of som parameter (e.g., time). A considerable reduction in the number of iterations may thus be achieved. (iii) Read in λ^t determined arbitrarily.

(3) Choice of the direction $u = -M^{-1}\nabla F$ for the iteration process. This direction is usually better than the direction of the gradient because it gives an extremum to the second-order expansion of F to a Taylor series [8]. The difference is most significant when F has a long valley in some direction. Then $-\nabla F$ is in the direction of the valley, but not necessarily in the direction of the minimum λ. In the present case, u is readily computed because we know both M and ∇F analytically. (In practice we do not acutally invert M. Instead, we solve a single set of linear equations $Mu = -\nabla F$.)

Nevertheless, the situation is not that simple. As may be seen from (16)F is asymptotically (i.e., for $|\lambda| \to \infty$) linear (this point is discussed in [4]). If we happen to choose the initial guess in the asymptotic region, M becomes almost singular. Then we are limited by the round-off errors of the computer and it is not possible to solve for u. In such a case one can still set $u = -\nabla F$ with the intention that after an iteration or two one would be outside the asymptotic region.

(4) The line search. The object here is to find $\min F$ along the direction u. This is done by finding two additional points along u, one on each side of the minimum, and fitting a parabola through the three points. The minimum of the parabola is taken as the starting point for the next iteration.

Algorithms for a line search vary and are largely a matter of taste, but may have a large effect on the efficiency of the whole program. We have chosen for this task a subroutine by Dax, whose algorithm is described in [8]. Our experience shows that for $n < 20$ the number of iterations usually vaires between three to ten (depending on the number of constraints and on the initial guess).

(5) Stopping criterion. The iterations are stopped if one of the following conditions are met: (i) $|\nabla F|^2/\alpha^2 < \delta$ where δ is some small given number and $n\alpha^2 = (1/m)\sum_{ir} A_{ri}^2$ or (ii) the number of iterations is greater than 20.

Whenever a run is complete, data for the next run is read, unless a cord with an and of data parameter is encountered.

4.5. *Memory Requirements*

The maximal values allowed for n and m are 200 and 20, respectively. We need one more row for the normalization constraint and two additional rows for the linear programming giving altogether the dimension 200×23. In order that the program does not take such a large memory in every run (especially since most problems do not have as many as 200 states and 20 constraints) A (and only A) is defined in blank common where it is given a small dimension which may be optionally increased by the user.

The linear programming destroys the original A. Instead of retaining two such matrices, the original A is written on a tape. In this way it is possible to reduce the storage requirements to within normal operating conditions.

4.6. *Practical Tips*

Although F is a strictly concave function and therefore the iteration procedure should theoretically converge for any initial guess, there are several practical problems. One that had already been mentioned is that the initial guess may be in the asymptotic region, where the Hessian M is algorithmically not positive definite. The solution is to use the direction of the gradient. Another problem may arise when the constraints are almost linearly dependent (cf. Section 2.1).

To avoid this problem we always transform to an orthonormal set of constraints, using the Gram–Schmidt orthonormalization procedure. Nevertheless, if the constraints are almost linearly dependent, the orthogonalization procedure is liable to introduce a very large (round-off) error. Therefore, the program issues a warning and the user is advised to examine the choice of constraints. Yet another problem may arise if some of the probabilities are too small (it is not possible to express zero as an exponent with a finite argument), when some of the Lagrange parameters may tend to infinity. As a result, small changes in the probabilities or in the averages may cause wild fluctuations in the Lagrange multipliers and the round-off errors in the values of the averages may render them meaningless. Moreover, the execution may be stopped by the computer's system because of an error in the function EXP, namely, an argument which is too small or too large. We have inserted a recovery procedure from such an error which returns the control to the main program which in turn reads the data for a new run. In addition, the program issues a warning when some of the probabilities are too small. The user is advised to delete such probabilities, thus transforming to a sample space with fewer points, where the above-mentioned problems would not be encountered.

4.7. *A Numerical Example*

A numerical example is given in Table I. The system chosen is of $n = 20$ states with one, two, or three constraints. The constraints chosen were the moments of the state index i:

$$A_{ri} = i^r, \qquad r = 1, 2, 3; \quad i = 1,..., 20. \tag{21}$$

The initial guess in all cases was $\lambda_r{}^t = 0$. The fast convergence of the algorithm is seen from the small number of iterations needed.

4.8. *Technical Details*

The program is written in FORTRAN IV and was executed on the CDC 6400 and Cyber computers of the Hebrew University of Jerusalem. It was compiled by FTN, scope 3.4.4. Compilation time is $\sim$10 sec, whereas the execution time of the above example was 1.2 sec. The following routines from the IMSL library were used: ZX3LP for linear programming and LEQT2P for solving a set of linear equations where the matrix is symmetric and positive definite.

TABLE I

An application of the computer program for $m = 1$, 2 and 3 constraints (note the decrease of the entropy S as m increases).

	m	1	2	3
DATA	b_1	15	15	15
	b_2	—	250	250
	b_3	—	—	4300
RESULTS	λ_0	5.0092	4.3616	2.0027
	λ_1	−0.1560	−0.0266	1.1937
	λ_2	0.0000	−0.0052	−0.1342
	λ_3	0.0000	0.0000	0.0038
	S	2.6697	2.6609	2.5456
	ITER[a]	3	3	4

[a] ITER is the number of iterations needed for achieving an agreement of $\sim 10^{-5}\%$ between the calculated averages and the given b_i's. The third constraint has an angle of 0.12° with its projection on the subspace spanned by A_0, A_1, and A_2.

ACKNOWLEDGMENTS

We thank A. Dax for the use of his line search subroutine and Y. Gat for assistance with the computer work.

REFERENCES

1. (a) E. T. JAYNES, *Phys. Rev.* **106** (1957), 620. (b) M. TRIBUS, "Rational Descriptions, Decisions and Design," Pergamon, Oxford, 1971.
2. R. ASH, "Information Theory," Interscience, New York, 1965.
3. (a) R. D. LEVINE AND R. B. BERNSTEIN, in "Dynamics of Molecular Collisions" (W. H. Miller, Ed.), Part B, p. 323, Plenum, New York, 1976. (b) R. D. LEVINE AND A. BEN-SHAUL, in "Chemical and Biochemical Applications of Lasers" (C. B. Moore, Ed.), Part II, p. 145, Academic Press, New York, 1977.
4. Y. ALHASSID, N. AGMON, AND R. D. LEVINE, *Chem. Phys. Lett.* **53** (1978), 22.
5. E. T. JAYNES, in "Statistical Physics" (K. W. Ford, Ed.), Vol. 3, p. 181, Benjamin, New York, 1963.
6. S. GASS, "Linear Programming: Methods and Applications," 4th ed., McGraw–Hill, New York, 1975.
7. R. D. LEVINE, *J. Chem. Phys.* **65** (1976), 3302.
8. A. DAX AND S. KANIEL, *SIAM J. Numerical Analysis* **16**, XXX (1979).