JOURNAL OF ALGEBRAIC STATISTICS
Vol. 2, No. 1, 2011, 1-13
ISSN 1309-3452 – www.jalgstat.com

An Iterative Method Converging to a Positive Solution of Certain Systems of Polynomial Equations

Dustin Cartwright

Department of Mathematics, University of California, Berkeley, CA 94720, USA

Abstract. We present a numerical algorithm for finding real non-negative solutions to a certain class of polynomial equations. Our methods are based on the expectation maximization and iterative proportional fitting algorithms, which are used in statistics to find maximum likelihood parameters for certain classes of statistical models. Since our algorithm works by iteratively improving an approximate solution, we find approximate solutions in the cases when there are no exact solutions, such as overconstrained systems.

2000 Mathematics Subject Classifications: 65H10, 14Q99

Key Words and Phrases: Expectation-maximization, iterative proportional fitting, numerical algebraic geometry, positive solutions

We present an iterative numerical method for finding non-negative solutions and approximate solutions to systems of polynomial equations satisfying the following two assumptions. First, for each equation, all the coefficients other than the constant term must be non-negative. Second, there is a technical assumption on the exponents, described at the beginning of Section 1, which, for example, is satisfied if all non-constant terms have the same total degree. In Section 3, there is a discussion of the range of possible systems which can arise under these hypotheses.

Because of the assumption on signs, we can write our system of equations as,

$$\sum_{\alpha \in S} a_{i\alpha} x^\alpha = b_i \quad \text{for } i = 1, \ldots, \ell, \tag{1}$$

where the coefficients $a_{i\alpha}$ are non-negative and the b_i are positive, and $S \subset \mathbb{R}^n_{\geq 0}$ is a finite set of possibly non-integer multi-indices. Our algorithm works by iteratively decreasing the generalized Kullback-Leibler divergence of the left-hand side and right-hand side of (1). The generalized Kullback-Leibler divergence of two positive vectors a and b is defined to be

$$D\left(a \,\|\, b\right) := \sum_i \left(a_i \log\left(\frac{a_i}{b_i}\right) - a_i + b_i \right). \tag{2}$$

Email address: dustin@math.berkeley.edu

The standard Kullback-Leibler consists only of the first term and is defined only for probability distributions, i.e. when the sum of each vector is 1. The last two terms are necessary so that the generalized divergence has, for arbitrary positive vectors a and b, the property of being non-negative and zero exactly when a and b are equal (Proposition 2).

Our algorithm converges to local minima of the Kullback-Leibler divergence, including exact solutions to the system (1). In this paper, we define an *approximate solution* to be a local minima of the Kullback-Leibler divergence. In order to find multiple local minima, we can repeat the algorithm for randomly chosen starting points. For finding approximate solutions, this may be sufficient. However, there are no guarantees of completeness for the exact solutions obtained in this way. Nonetheless, we hope that in certain situations, our algorithm will find applications both for finding exact and approximate solutions.

Lee and Seung applied the EM algorithm to the problem of non-negative matrix factorization [7]. They introduced the generalized Kullback-Leibler divergence in (2) and used it to find approximate non-negative matrix factorizations. Since the product of two matrices can be expressed by polynomials in the entries of the matrices, matrix-factorization is a special case of the equations in (1).

In [4], Dedieu and Shub study a generalization of Newton's algorithm to overdetermined systems of equations within the context of Smale's α-theory. Their analogue of our apporximate solutions are local minima of the norm-squared error of the equations in (1):

$$\sum_{i=1}^{\ell} \left(\sum_{\alpha \in S} a_{i\alpha} x^{\alpha} - b_i \right)^2 ,$$

and, under certain conditions, they obtain effective bounds on convergence to these minima. While their approach has the advantage of using the more standard Euclidean norm instead of the Kullback-Leibler divergence, the convergence of Newton iterations depends on the choice of starting point. When our hypotheses on the system of equations are satisfied, we obtain convergence for any starting point, albeit without explicit bounds on the rate of convergence like in [4].

For finding exact solutions to arbitrary systems of polynomials, there are a variety of approaches which find all complex or all real solutions. Homotopy continuation methods find all complex roots of a system of equations [9]. Even to find only positive roots, these two methods finds all complex or all real solutions, respectively. Lasserre, Laurent, and Rostalski have applied semidefinite programming to find all real solutions to a system of equations and a slight modificiation of their algorithm will find all positive real solutions [5, 6]. Nonetheless, neither of these methods has any notion of approximate solutions.

For directly finding only positive real solutions, Bates and Sottile have proposed an algorithm based on fewnomials bounds on the number of real solutions [1]. However, their method is only effective when the number of monomials (the set S in our notation) is only slightly more than the number of variables. Our method only makes weak assumptions on the set of monomials, but stronger assumptions on the coefficients.

Our inspiration comes from tools for maximum likelihood estimation in statistics. Parameters which maximize the likelihood are exactly the parameters such that the model probabilities are closest to the empirical probabilities, in the sense of mimimizing Kullback-Leibler divergence. Expectation-Maximization [8, Sec. 1.3] and Iterative Proportional Fitting [3] are well-known iterative methods for maximum likelihood estimation. We re-interpret these algorithms as methods for approximating solutions to polynomial equations, in which case their applicability can be somewhat generalized.

The impetus behind the work in this paper was the need to find approximate positive solutions to systems of bilinear equations in [2]. In this case the variables represented expression levels, which only made sense as positive parameters. Moreover, in order to accomodate noise in the data, there were more equations than variables, so it was necessary to find approximate solutions. Thus, the algorithm described in this paper was the most appropriate tool. Here, we generalize beyond bilinear equations and present proofs.

An implementation of our algorithm in the C programming language is freely available at `http://math.berkeley.edu/~dustin/pos/`.

In Section 1, we describe the algorithm and the connection to maximum likelihood estimation. In Section 2, we prove that the necessary convergence for our algorithm. Finally, in Section 3, we show that even with our restrictions on the form of the equations, there can be exponentially positive real solutions.

1. Algorithm

In addition to the assumption on the non-negativity of the coefficients, we make a further assumption on the exponent set. We assume that we have an $s \times n$ non-negative matrix g, with no column identically zero, and positive real numbers d_j for $1 \leq j \leq s$ such that for each $\alpha \in S$ and each $j \leq s$, $\sum_{i=1}^n g_{ji}\alpha_i$ is either 0 or d_j. For example, if all the monomials x^α have the same total degree d_1, we can take $s = 1$ and $g_{1i} = 1$ for all i. The other case of particular interest is multilinear systems of equations, in which each α_i is at most one. In this case the variables can be partitioned into sets such that the equations are linear in each set of variables, so we can take $d_j = 1$ for all j. Note that because d_j is in the denominator in (4), convergence is fastest when the d_j are small, such as in the multilinear case. We also note that, for an arbitrary set of exponents S, there may not exist such a g.

The algorithm begins with a randomly chosen starting vector and iteratively improves it through two nested iterations:

- Initialize x with n randomly chosen positive real numbers.
- Loop until the vector x stablizes:
    - For all $\alpha \in S$, compute
$$w_\alpha := \sum_i b_i \frac{a_{i\alpha}x^\alpha}{\sum_\beta a_{i\beta}x^\beta}. \tag{3}$$
    - Loop until the vector x stablizes:
        - Loop for j from 1 to s:

- Simultaneously update all entries of x:

$$x_i \leftarrow x_i \left(\frac{\sum_\alpha \alpha_i g_{ji} w_\alpha}{\sum_\alpha \alpha_i g_{ji} a_\alpha x^\alpha} \right)^{g_{ji}/d_j} \qquad \text{where } a_\alpha = \sum_i a_{i\alpha}. \quad (4)$$

Because there is no subtraction, it is clear that the entries of x remain positive throughout this algorithm.

Since our algorithm is iterative, it is dependent on stopping criteria to determine when to stop the iterations. For the inner loop, we terminate when the relative change in the components of the vector x is less than the square of the relative change in the last iteration of the outer loop. In this way, the threshold becomes more stringent as the algorithm converges. For the outer loop, we use the quadratic Taylor expansion of the divergence (2) to estimate the distance between the current value of x and the local minimum of the divergence. When the estimated distance is less than a given threshold, the loop terminates. This criterion was chosen over more straightforward ones, which proved to be unreliable in practice.

Note that each iteration is quite fast. The computation in (3) is equivalent to a single evaluation of the system of polynomials, plus a division for each term. On the other hand, each update (4) is potentially faster than a single evaluation of the system of polynomials, especially if $d_j = 1$, because the iteration is only over S, the set of exponent vectors, and not over all the terms. Finally to gain additional speed, in our implementation, when the algorithm gets close to an actual solution, we use Newton's algorithm to quickly refine it.

Our method is inspired by interpreting the equations in (1) as a maxmimum likelihood problem for a statistical model and applying the well-known methods of Expectation-Maximization (EM) and Iterative Proportional Fitting (IPF). For simplicity, let us assume that all the monomials x^α have the same total degree. Our statistical model is that a hidden process generates an integer $i \leq \ell$ and an exponent vector α with joint probability $a_{i\alpha} x^\alpha$. The vector x contains n positive parameters for the model, restricted such that the total probability $\sum_{i,\alpha} a_{i\alpha} x^\alpha$ is 1. The empirical data consists of repeated observations of the integer i, but not the exponent α, and b_i is the proportion of observations of i. In this situation, the vector x which minimizes the divergence of (1) is the maximum likelihood parameters for the empirical distribution b_i. The inner loop of the algorithm consists of using IPF to solve the log-linear hidden model and the outer loop consists of using EM to estimate the distribution on the hidden states.

We give a simple example of our algorithm in action.

Example 1. The (reciprocal of the) golden ratio is the unique positive root of the equation

$$x^2 + x - 1 = 0.$$

The coefficients have the correct signs, but doesn't satisfy the condition on the exponent vectors. We can fix this by adding a dummy variable y, which will be force to be equal to 1:

$$\begin{aligned} x^2 + xy &= 1 \\ y^2 &= 1. \end{aligned} \qquad (5)$$

Inner loop iterations	x	y
5	0.6944381615	0.9327435254
4	0.6339861529	0.9862879982
3	0.6217014340	0.9968624369
5	0.6187745905	0.9993672667
6	0.6181817273	0.9998738089
8	0.6180632943	0.9999749698
9	0.6180397990	0.9999950375
11	0.6180351404	0.9999990163

Table 1: Eight iterations of the outer loop applied to the system of equations in (5).

A sample run took 8 iterations of the outer loop and a total of 51 iterations of the inner loop to converge to 5 digits of accuracy, as detailed in Table 1. □

Example 2. Next we give an example of a system with only approximate solutions. One can check that the system of equations

$$x^2 + y^2 = 1$$
$$x^2 + 2xy + y^2 = 4$$

has no real solutions. On this system, our algorithm converges to the approximate solution $(x, y) = (\sqrt{5/6}, \sqrt{5/6}) \approx (0.91287, 0.91287)$. With these values, $x^2 + y^2 = 5/3$, which is somewhat greater than 1, and $x^2 + 2xy + y^2 = 10/3$, which is somewhat less than 4. □

Although the Kullback-Leibler divergence is rarely used outside probability and statistics, it can be approximated as a weighted L^2-norm. In order to make this statement precise, we define the function $C(t)$ to be 1 if $t \leq 1$ and to be $\log(t)/(t-1)$ if $t > 1$. Note that $C(t)$ is approximately equal to 1 in a neighborhood fo $t = 1$, which will be the case when comparing vectors which are close to each other.

Proposition 1. *With a and b two positive real vectors, and $C(t)$ as defined above, then $D(a \| b)$ is bounded below by the square of the weighted L^2-norm*

$$\sum_{i=1}^{n} \frac{C(a_i/b_i)}{2b_i}(a_i - b_i)^2 \tag{6}$$

Proposition 1 implies that, at least for nearby vectors, the major difference between divergence and Euclidean distance is that divergence places more weight on components whose values are closer to zero.

Proof. [Proof of Proposition 1] Note that $D(a \| b) = \sum_{i=1}^{m} D(a_i \| b_i)$ and thus, it suffices to prove the statement in the case when a and b are scalars. We let $t = a/b$, and then,

$$D(a \| b) = a \log\left(\frac{a}{b}\right) - a + b = b(t \log t - t + 1) = b \int_1^t \log s \, ds.$$

Now we bound the integral using a linear approximation of the logarithm. For $s \leq 1$, $\log s \leq s - 1$. On the other hand, for $1 \leq s \leq t$, $\log s \geq (s-1)\log(t)/(t-1)$, since log is a convex function. Therefore, whether $t \leq 1$ or $t \geq 1$

$$\int_1^t \log s \, ds \geq C(t) \int_1^t (s-1)\, ds.$$

Combining this inequality with (1),

$$D\left(a \,\|b\right) \geq C(t) b \int_1^t (s-1)\, ds = \frac{C(a/b)b}{2}\left(\frac{a}{b} - 1\right)^2 = \frac{C(a/b)}{2b}(a-b)^2,$$

which is the desired result.

As a corollary, we get:

Proposition 2. *For a and b vectors of positive real numbers, the divergence $D\left(a \,\|b\right)$ is always non-negative with $D\left(a \,\|b\right) = 0$ if and only if $a = b$.*

Proof. Since the quantity $C(a_i/b_i)$ from Proposition 1 is always positive, each term of the summation in (6) is non-negative and zero if and only $a_i = b_i$.

2. Proof of convergence

In this section we prove our main theorem:

Theorem 3. *The Kullback-Leibler divergence*

$$\sum_{i=1}^{\ell} D\left(b_i \,\middle\|\, \sum_{a \in S} a_{i\alpha} x^{\alpha}\right). \tag{7}$$

is weakly decreasing during the algorithm in Section 1. Moreover, assuming that the set S contains a multiple of each unit vector e_i, i.e. some power of each x_i appears in the system of equations, then the vector x converges to a critical point of the function (7) or the boundary of the positive orthant.

Remark 4. *The condition on S is necessary to ensure that the vector x remains bounded during the algorithm.*

We begin by establishing several basic properties of the generalized Kullback-Leibler divergence in Lemmas 5 and 6. The proof of Theorem 3 itself is divided into two parts, corresponding to the two nested iterative loops. The first step is to prove that the updates (4) in the inner loop converge a local minimum of the divergence $D\left(w_\alpha \,\|a_\alpha x^\alpha\right)$. The second step is to show that this implies that the outer loop strictly decreases the divergence function (7) exact at a critical point.

Lemma 5. *Suppose that a and b are vectors of m positive real numbers. Let t be any positive real number, and then*

$$D\left(a\,\|tb\right) = D\left(a\,\|b\right) + (t-1)\sum_{i=1}^{m} b_i - \sum_{i=1}^{m} a_i \log t$$

Proof. As in the proof of Proposition 1, we can assume that a and b are scalars. In this case, it becomes a straightforward computation.

Lemma 6. *If a and b are vectors of m positive real numbers, then we can relate their divergence to the divergence of their sums by*

$$D\left(a\,\|b\right) = D\left(\textstyle\sum_{i=1}^m a_i \,\|\sum_{i=1}^m b_i\right) + D\left(a \,\middle\|\, \frac{\sum_{i=1}^m a_i}{\sum_{i=1}^m b_i} b\right).$$

Proof. We let $A = \sum_{i=1}^m a_i$ and $B = \sum_{i=1}^m b_i$, and apply Lemma 5 to the last term:

$$\begin{aligned} D\left(a \,\middle\|\, \frac{A}{B} b\right) &= D\left(a\,\|b\right) + \left(\frac{A}{B} - 1\right) B - A \log\frac{A}{B} \\ &= D\left(a\,\|b\right) - A\log\frac{A}{B} + A - B = D\left(a\,\|b\right) - D\left(A\,\|B\right). \end{aligned}$$

After rearranging, we get the desired expression.

Lemma 7. *The update rule (4) weakly decreases the divergence. If*

$$x_i' = x_i \left(\frac{\sum_\alpha \alpha_i g_{ji} w_\alpha}{\sum_\alpha \alpha_i g_{ji} a_\alpha x^\alpha}\right)^{g_{ji}/d_j},$$

then

$$\sum_\alpha D\left(w_\alpha \,\|a_\alpha (x')^\alpha\right) \le \sum_\alpha D\left(w_\alpha \,\|a_\alpha x^\alpha\right) - \frac{1}{d}\sum_{i=1}^n D\left(\sum_\alpha \alpha_i g_{ji} w_\alpha \,\middle\|\, \sum_\alpha \alpha_i g_{ji} a_\alpha x^\alpha\right). \quad (8)$$

Proof. First, since the statement only depends on the jth row of the matrix g, we can assume that g is a row vector and we drop j from future subscripts. Second, we can assume that $d = 1$ by replacing g_i with g_i/d.

Third, we reduce to the case when $g_i = 1$ for all i. We define a new set of exponents $\tilde\alpha$ and coefficients $\tilde a_{\tilde\alpha}$ by $\tilde\alpha_i = g_i \alpha_i$ and $\tilde a_\alpha = a_\alpha \prod x_i$, where the product is taken over all indices i such that $g_i = 0$. We take $\tilde x$ to be a vector indexed by those i such that $g_i \neq 0$. Then, under the change of coordinates $\tilde x_i = x_i^{1/g_i}$, we have $a_\alpha x^\alpha = \tilde a_\alpha \tilde x^{\tilde\alpha}$ and the update rule in (4) is the same for the new system with coefficients $\tilde a_{\tilde\alpha}$ and exponents $\tilde\alpha$. Furthermore, if all entries of $\tilde\alpha$ are zero, then $\tilde x^{\tilde\alpha} = 1$ for all vectors x and so we can drop $\tilde\alpha$ from our exponent set. Therefore, for the rest of the proof, we drop the tildes, and assume that $\sum_i \alpha_i = 1$ for all $\alpha \in S$ and $g_i = 1$ for all i, in which case g drops out of the equations.

To prove the desired inequality, we substitute the updated assignment x' into the definition of Kullback-Leibler divergence:

$$\begin{aligned} D\left(w_\alpha \,\middle\|\, a_\alpha (x')^\alpha\right) &= w_\alpha \log \left(\frac{w_\alpha}{a_\alpha x^\alpha \prod_{i=1}^n \left(\frac{\sum_\beta w_\beta}{\sum_\beta \beta_i a_\beta x^\beta} \right)^{\alpha_i}} \right) - w_\alpha + a_\alpha (x')^\alpha \\ &= w_\alpha \log \frac{w_\alpha}{a_\alpha x^\alpha} - \sum_{i=1}^n \alpha_i w_\alpha \log \frac{\sum_\beta \beta_i w_\beta}{\sum_\beta \beta_i a_\beta x^\beta} - w_\alpha + a_\alpha (x')^\alpha \\ &= D\left(w_\alpha \,\|\, a_\alpha x^\alpha\right) - \sum_{i=1}^n \alpha_i w_\alpha \log \frac{\sum_\beta \beta_i w_\beta}{\sum_\beta \beta_i a_\beta x^\beta} - a_\alpha x^\alpha + a_\alpha (x')^\alpha. \qquad (9) \end{aligned}$$

On the other hand, let C denote the last term of (8), which we can expand as,

$$\begin{aligned} C &= \sum_{i=1}^n D\left(\sum_\alpha \alpha_i w_\alpha \,\middle\|\, \sum_\alpha \alpha_i a_\alpha x^\alpha \right) \\ &= \sum_{i=1}^n \left(\left(\sum_\alpha \alpha_i w_\alpha \right) \log \frac{\sum_\alpha \alpha_i w_\alpha}{\sum_\alpha \alpha_i a_\alpha x^\alpha} - \sum_\alpha \alpha_i w_\alpha + \sum_\alpha \alpha_i a_\alpha x^\alpha \right) \\ &= \sum_{i=1}^n \sum_\alpha \left(\alpha_i w_\alpha \log \frac{\sum_\beta \beta_i w_\beta}{\sum_\beta \beta_i a_\beta x^\beta} - \alpha_i w_\alpha + \alpha_i a_\alpha x^\alpha \right) \\ &= \sum_\alpha \left(\sum_{i=1}^n \alpha_i w_\alpha \log \frac{\sum_\beta \beta_i w_\beta}{\sum_\beta \beta_i a_\beta x^\beta} \right) - w_\alpha + a_\alpha x^\alpha, \qquad (10) \end{aligned}$$

where the last step follows from the assumption that that $\sum_i \alpha_i = 1$ for all $\alpha \in S$. We take the sum of (9) over all $\alpha \in S$ and add it to (10) to get,

$$\sum_\alpha D\left(w_\alpha \,\middle\|\, a_\alpha (x')^\alpha\right) + C = \sum_\alpha D\left(w_\alpha \,\|\, a_\alpha x^\alpha\right) - \sum_\alpha b_\alpha + \sum_\alpha a_\alpha (x')^\alpha. \qquad (11)$$

Finally, we expand the last term of (10) using the definition of x' and apply the arithmetic-geometric mean inequality,

$$\begin{aligned} \sum_\alpha a_\alpha (x')^\alpha &= \sum_\alpha a_\alpha x^\alpha \prod_{i=1}^n \left(\frac{\sum_\beta \beta_i b_\beta}{\sum_\beta \beta_i a_\beta x^\beta} \right)^{\alpha_i} \\ &\leq \sum_\alpha a_\alpha x^\alpha \sum_{i=1}^n \alpha_i \frac{\sum_\beta \beta_i b_\beta}{\sum_\beta \beta_i a_\beta x^\beta} \\ &= \sum_{i=1}^n \left(\sum_\alpha \alpha_i a_\alpha x^\alpha \right) \frac{\sum_\beta \beta_i b_\beta}{\sum_\beta \beta_i a_\beta x^\beta} \\ &= \sum_{i=1}^n \sum_\beta \beta_i b_\beta = \sum_\beta b_\beta. \end{aligned}$$

Together with (11), this gives the desired inequality.

Proposition 8. *A positive vector x is a fixed point of the update rule (4) for all $1 \leq j \leq s$ if and only if x is a critical point of the divergence function $\sum_\alpha D\left(w_\alpha \,\|a_\alpha x^\alpha\right)$.*

Proof. For the update rule to be constant means that the numerator and denominator in (4) are equal, i.e.

$$\sum_\alpha \alpha_i g_{ji} a_\alpha x^\alpha = \sum_\alpha \alpha_i g_{ji} w_\alpha \quad \text{for all } i \text{ and } j. \tag{12}$$

By our assumption on g, for each i, some g_{ji} is non-zero, so (12) is equivalent to

$$\sum_\alpha \alpha_i a_\alpha x^\alpha = \sum_\alpha \alpha_i w_\alpha \quad \text{for all } i. \tag{13}$$

On the other hand, we compute the partial derivative

$$\frac{\partial}{\partial x_i} \sum_\alpha D\left(w_\alpha \,\|a_\alpha x^\alpha\right) = \sum_\alpha -w_\alpha \frac{\alpha_i}{x_i} + \alpha_i a_\alpha \frac{x^\alpha}{x_i}.$$

Since each x_i is assumed to be non-zero, it is clear that all partial derivatives being zero is equivalent to (13).

Lemma 9. *If we define w_α as in (3), then*

$$\sum_{i=1}^n D\left(b_i \,\middle\|\, \sum_\alpha a_{i\alpha}(x')^\alpha\right) - \sum_{i=1}^n D\left(b_i \,\middle\|\, \sum_\alpha a_{i\alpha} x^\alpha\right) \leq \sum_\alpha D\left(w_\alpha \,\|a_\alpha (x')^\alpha\right) - \sum_\alpha D\left(w_\alpha \,\|a_\alpha x^\alpha\right).$$

Moreover, a positive vector x is a fixed point if and only if x is a critical point for the divergence function.

Proof. We consider

$$\sum_{i,\alpha} D\left(w_{i\alpha} \,\|a_{i\alpha}(x')^\alpha\right) \qquad \text{where } w_{i\alpha} = \frac{b_i a_{i\alpha} x^\alpha}{\sum_\beta a_{i\beta} x^\beta}, \tag{14}$$

and apply Lemma 6 in two different ways. First, by applying Lemma 6 to each group of (14) with fixed α, we get

$$\sum_{i,\alpha} D\left(w_{i\alpha} \,\|a_{i\alpha}(x')^\alpha\right) = \sum_\alpha D\left(w_\alpha \,\|a_\alpha (x')^\alpha\right) + \sum_{i,\alpha} D\left(w_{i\alpha} \,\middle\|\, \frac{\sum_j w_{j\alpha}}{\sum_j a_\alpha (x')^\alpha} a_{i\alpha}(x')^\alpha\right).$$

In the last term, the monomials $(x')^\alpha$ cancel and so it is a constant independent of x' which we denote E. On the other hand, we can apply Lemma 6 to each group in (14) with fixed i,

$$\sum_{i,\alpha} D\left(w_{i\alpha} \left\| a_{i\alpha}(x')^\alpha\right.\right) = \sum_i D\left(b_i \left\| \sum_\alpha a_{i\alpha}(x')^\alpha\right.\right) + \sum_{i,\alpha} D\left(w_{i\alpha} \left\| \frac{b_i a_{i\alpha}(x')^\alpha}{\sum_\beta a_{i\beta}(x')^\beta}\right.\right).$$

We can combine these equations to get

$$\sum_i D\left(b_i \left\| \sum_\alpha a_{i\alpha}(x')^\alpha\right.\right) = \sum_\alpha D\left(w_{i\alpha} \left\| a_\alpha(x')^\alpha\right.\right) + E - \sum_{i,\alpha} D\left(w_{i\alpha} \left\| \frac{b_i a_{i\alpha}(x')^\alpha}{\sum_\beta a_{i\beta}(x')^\beta}\right.\right). \quad (15)$$

By Proposition 2, the last term of (15) is non-negative, and by the definition of $w_{i\alpha}$, it is zero for $x' = x$. Therefore, any value of x' which decreases the first term compared to x will also decrease the left hand side by at least as much, which is the desired inequality.

In order to prove the statement about the derivative, we consider the derivative of (15) at $x' = x$. Because the last term is mimimized at $x' = x$, its derivative is zero, so

$$\left.\frac{\partial}{\partial x'_j}\right|_{x'=x} \sum_i D\left(b_i \left\| \sum_\alpha a_{i\alpha}(x')^\alpha\right.\right) = \left.\frac{\partial}{\partial x'_j}\right|_{x'=x} \sum_{i,\alpha} D\left(w_{i\alpha} \left\| a_{i\alpha}(x')^\alpha\right.\right).$$

By Proposition 8, a positive vector x is a fixed point of the inner loop if and only if these partial derivatives on the right are zero for all indices j, which is the definition of a critical point.

Proof. [Proof of Theorem 3] The Kullback-Leibler divergence $\sum_\alpha D\left(w_\alpha \left\| a_\alpha x^\alpha\right.\right)$ decreases at each step of the inner loop by Lemma 7. Thus, by Lemma 9, the divergence

$$\sum_{i=1}^n D\left(b_i \left\| \sum_\alpha a_{i\alpha} x^\alpha\right.\right) \quad (16)$$

decreases at least as much. However, the divergence (16) is non-negative according to Proposition 2. Therefore, the magnitude of the decreases in divergence must approach zero over the course of the algorithm. By Lemma 7, this means that the quantity C in that theorem must approach zero. By Proposition 2, this means that the quantities in that divergence approach each other. However, up to a power, these are the numerator and denominator of the factor in the update rule (4), so the difference between consecutive vectors x approaches zero.

Thus, we just need to show that x remains bounded. However, since some power of each variable x_i occurs in some equation, as x_i gets large, the divergence for that equation also gets arbitrarily large. Therefore, each x_i must remain bounded, so the vector x must have a limit as the algorithm is iterated. If this limit is in the interior of the positive orthant, then it must be a fixed point. By Lemma 9 and Proposition 8, this fixed point must be a critical point of the divergence (7).

3. Universality

Although the restriction on the exponents and especially the positivity of the coefficients seem like strong conditions, such systems can nonetheless be quite complex. In this section, we investigate the breadth of such equations.

Proposition 10. *For any system of ℓ real polynomials in n variables, there exists a system of $\ell + 1$ equations in $n + 1$ variables, in the form (1), such that the positive solutions $(x_1, \dots, x_n)$ to the former system are in bijection with the positive solutions $(x'_1, \dots, x'_{n+1})$ of the latter, with $x'_i = x_i/x_{n+1}$.*

Proof. We write our system of equations as $\sum_{\alpha\in S} a_{i\alpha}x^\alpha = 0$ for $1 \leq i \leq \ell$, where $S \subset \mathbb{N}^n$ is an arbitrary finite set of exponents and $a_{i\alpha}$ are any real numbers. We let d be the maximum degree of any monomial x^α for $\alpha \in S$. We homogenize the equations with a new variable x_{n+1}. Explicitly, define $S' \subset \mathbb{N}^{n+1}$ to consist of $\alpha' = (\alpha, d - \sum_i \alpha_i)$ for all α in S and we write $a_{i\alpha'} = a_{i\alpha}$. We add a new equation with coefficients $a_{\ell+1,\alpha} = 1$ for all $\alpha \in S'$ and $b_{\ell+1} = 1$. For this system, we can clearly take $g_{1i} = 1$ and $d_1 = d$ to satisfy the condition on exponents. Furthermore, for any positive solution $(x_1, \dots, x_n)$ to the original system of equations, $(x'_1, \dots, x'_{n+1})$ with $x'_i = x_i/\big(\sum_\alpha x^\alpha\big)^{1/d}$ and $x'_{n+1} = 1/\big(\sum_\alpha x^\alpha\big)^{1/d}$ is a solution to the homogenized system of equations.

Next, we add a multiple of the last equation to each of the others in order to make all the coefficients positive. For each $1 \leq i \leq \ell$, choose a positive $b_i > -\min_\alpha\{a_{i\alpha} \mid \alpha \in S'\}$, and define $a'_{i\alpha} = a_{i\alpha} + b_i$. By construction, the resulting system has all positive coefficients, and since the equations are formed from the previous equations by elementary linear transformations, the set of solutions are the same.

The practical use of the construction in the proof of Proposition 10 is mixed. The first step, of homogenizing to deal with arbitrary sets of exponents, is a straightforward way of guaranteeing the existence of the matrix g. However, for large systems, the second step tends to produce an ill-conditioned coefficient matrix. In these cases, our algorithm converges very slowly. Nonetheless, Proposition 10 shows that in the worst case, systems satisfying our hypotheses can be as complicated as arbitrary polynomial systems.

Proposition 11. *There exist bilinear equations in $2m$ variables with $\binom{2m-2}{m-1}$ positive real solutions.*

Proof. We use a variation on the technique used to prove Proposition 10.

First, we pick $2m-2$ generic homogeneous linear functions $b_1, \dots, b_{2m-2}$ on m variables. By generic, we mean for any m of the b_k, the only simultaneous solution of all m linear equations is the trivial one. This genericity implies that any $m-1$ of the b_k define a point in $\mathbb{P}^{m-1}$ By taking a linear changes of coordinates in each set of variables, we can assume that all of these points are positive, i.e. have a representative consisting of all positive real numbers.

Then we consider the system of equations

$$b_k(x_1, \dots, x_m) \cdot b_k(x_{m+1}, \dots, x_n) = 0, \text{ for } 1 \leq k \leq 2m-2 \tag{17}$$

$$(x_1 + \ldots + x_m)(x_{m+1} + \ldots + x_{2m}) = 1 \tag{18}$$
$$x_1 + \ldots + x_m = 1. \tag{19}$$

The equations (17) are bihomogeneous and so we can think of their solutions in $\mathbb{P}^{m-1} \times \mathbb{P}^{m-1}$. There are exactly $\binom{2m-2}{m-1}$ positive real solutions, corresponding to the subsets $A \subset [2m-2]$ of size $m-1$. For any such A, there is a unique, distinct solution satisfying $b_k(x_1, \ldots, x_m) = 0$ for all k in A and $b_k(x_{m+1}, \ldots, x_{2m}) = 0$ for all k not in A. By assumption, for each solution, all the coordinates can be chosen to be positive. The last two equations (18) and (19) dehomogenize the system in a way such that there are $\binom{2m-2}{m-1}$ positive real solutions. Finally, as in the last paragraph of the proof of Proposition 10, we can add multiples of (18) to the equations (17) in order to make all the coefficients positive.

Example 3. We illustrate the construction in Proposition 11 in the simplest case, when $m = 2$, and $n = 2m = 4$. We take the two homogenous functions $b_1 = x_1 - x_2$ and $b_2 = x_1 - 2x_2$, which each have a positive real solution, as desired. Following (17), we have the bilinear equations equations:

$$(x_1 - x_2)(x_3 - x_4) = x_1x_3 - x_1x_4 - x_2x_3 + x_2x_4 = 0$$
$$(x_1 - 2x_2)(x_3 - 2x_4) = x_1x_3 - 2x_1x_4 - 2x_2x_3 + 4x_2x_4 = 0$$

We appropriate multiples of $(x_1 + x_2)(x_3 + x_4) = 1$ to these two equations, to get the system with non-negative coefficients:

$$\begin{aligned} 2x_1x_3 + 2x_2x_4 &= 1 \\ 3x_1x_3 + 6x_2x_4 &= 2 \\ x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 &= 1 \\ x_1 + x_2 &= 1. \end{aligned}$$

The two solutions to these equations are $(1/2, 1/2, 2/3, 1/3)$ and $(1/3, 2/3, 1/2, 1/2)$. Because of the symmetry, our algorithm will converge to each half of the time. □

Acknowledgments

We thank Bernd Sturmfels for reading a draft of this manuscript. This work was supported by the Defense Advanced Research Projects Agency project "Microstates to Macrodynamics: A New Mathematics of Biology" and the U.S. National Science Foundation (DMS-0456960).

References

[1] Dan Bates and Frank Sottile. Khovanskii-Rolle continuation for real solutions. arXiv:0908.4579, 2009.

[2] Dustin A. Cartwright, Siobhan M. Brady, David A. Orlando, Bernd Sturmfels, and Philip N. Benfey. Reconstruction spatiotemporal gene expression data from partial observations. *Bioinformatics*, 25(19):2581–2587, 2009.

[3] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Math. Stat.*, 43(5):1470–1480, 1972.

[4] Jean-Pierre Dedieu and Mike Shub. Newton's method for overdetermined systems of equations. *Mathematics of Computation*, 69(231):1099–1115, July 2000.

[5] Jean B. Lasserre, Monique Laurent, and Philipp Rostalski. Semidefinite characterization and computation of real radical ideals. *Foundations of Computational Mathematics*, 8(5):607–647, 2008.

[6] Jean B. Lasserre, Monique Laurent, and Philipp Rostalski. A prolongation-projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27–29):2685–2700, 2009.

[7] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. *Adv. Neural Info. Proc. Syst.*, 13:556–562, 2001.

[8] Lior Pachter and Bernd Sturmfels. *Algebraic Statistics for Computational Biology*. Cambridge University Press, 2005.

[9] Jan Verschelde. Algorithm 795: PHCPACK: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, June 1999.