Maximum Likelihood Estimation and Polynomial System Solving

Kim Batselier
Department of Electrical Engineering (ESAT), SCD
Katholieke Universiteit Leuven
3001 Leuven, Belgium
Email: kim.batselier@esat.kuleuven.be

Bart De Moor
Department of Electrical Engineering (ESAT), SCD
Katholieke Universiteit Leuven
3001 Leuven, Belgium
Email: bart.demoor@esat.kuleuven.be

ABSTRACT

Discrete statistical models are probably one of the most important tools in bioinformatics. Learning the model parameters for these models from observations is commonly done via a maximum likelihood principle. In most cases however there are many solutions to this problem and only a local maximum can be found. The Expectation Maximization algorithm is the method of choice to tackle this problem. This paper presents a method that allows to find the global maximum likelihood estimate. The focus is limited on a specific class of discrete statistical models. For these models it is shown that the maximum likelihood estimates correspond with the roots of a multivariate polynomial system. Then, a new algorithm is presented, set in a linear algebra framework, which allows to find all these roots by solving a generalized eigenvalue problem. An illustrative example is worked out in which DNA is modeled in order to identify CpG islands.

I. INTRODUCTION

The term maximum likelihood was first coined by Fisher in 1922 [1], [2]. Since then, the use of maximum likelihood estimation has become extremely popular in a vast number of fields. Statistical methods are paramount for analysing biological data and maximum likelihood estimation therefore is the dominant framework in the different fields of computational biology. Hidden Markov Models, for example, are statistical models for discrete data in which the system being modeled is assumed to be a Markov process with unobserved states. They can be considered as the simplest dynamic Bayesian networks and were first used for speech recognition in the mid-1970's [3]. In the second half of the 1980's they were first used for modeling biological sequences [4] and have since then become ubiquitous in the field of bioinformatics. Some of their applications are in DNA sequence alignment, gene finding [5], phylogenetics [6], [7] and much more. The two most common methods for finding maximum likelihood estimates are Expectation Maximization (EM) [8] and Markov Chain Monte Carlo (MCMC) [9], [10]. EM is an iterative hill climbing algorithm. Starting from some initial guess, model parameters are updated consecutively such that the likelihood increases until convergence has occurred. This dependence of the solution on the initial guess means that for the case of many solutions only a local maximum is obtained. MCMC methods are typically used in a Bayesian Learning setting where one is usually more interested in posterior distributions than in point estimates. These methods allow to generate samples from unknown distributions which then can be used to calculate point estimates (such as the mode or mean). Although this method is commonly used to sample the posterior distribution it can be also utilized to obtain maximum likelihood estimates [11]. A more recent method has come from the field of algebraic statistics which seeks to mix algebraic geometry and commutative algebra with statistics [12]. This method relies on Buchberger's algorithm that transforms the polynomial system to an equivalent one which is easier to solve. Buchberger's algorithm is a symbolic method and has therefore inherent difficulties with dealing with real numbers.

This paper seeks out to establish a numerical method for finding maximum likelihood estimates which is guaranteed to find the global maximum. This is achieved by first showing that for algebraic statistical models maximum likelihood estimation of the model parameters corresponds with solving a polynomial system. Then an algorithm is presented which allows to find all solutions of polynomial systems by solving a generalized eigenvalue problem.

II. MAXIMUM LIKELIHOOD AND POLYNOMIAL SYSTEMS

The models considered in this paper are algebraic statistical models on a discrete state space. We first provide the definition of a probability distribution on a m-dimensional state space $[m]$. The definitions and notation in this section are adopted from [12].

Definition 1: A probability distribution on the set $[m]$ is a point in the probability simplex

$$\Delta_{m-1} := \{p \in \mathbb{R}^m : \sum_{i=1}^{m} p_i = 1 \text{ and } p_j \geq 0 \ \forall j\}.$$

The dimension of the simplex Δ_{m-1} is $m-1$. A statistical model for discrete data is a subset of the simplex Δ_{m-1}. Each coordinate p_i represents the probability of observing the state i and must therefore be a non-negative real number. We consider the class of statistical models where each probability distribution is parametrized in a polynomial fashion.

Definition 2: An algebraic statistical model is the image of a polynomial mapping

$$\mathbf{p} : \mathbb{R}^n \to \mathbb{R}^m, \theta = (\theta_1, \ldots, \theta_n) \to (p_1(\theta), \ldots, p_m(\theta))$$

where θ are the model parameters and each p_i a polynomial in n unknowns. This means p_i has the form

$$p_i(\theta) = \sum_{a \in \mathbb{Z}^n_{\geq 0}} c_a.\theta^a = \sum_{a \in \mathbb{Z}^n_{\geq 0}} c_a.\theta_1^{a_1}\theta_2^{a_2}\ldots\theta_n^{a_n}$$

where $c_a \in \mathbb{R}$ are called the coefficients and $\theta^a = \theta_1^{a_1}\theta_2^{a_2}\ldots\theta_n^{a_n}$ monomials. All of the exponents $a_1, \ldots, a_n$ are nonnegative integers, that is, $a_1, \ldots, a_n \in \mathbb{N} = \{0, 1, 2, 3, \ldots\}$.

Data are typically given as a sequence of observations $\{y_1, y_2, \ldots, y_N\}$ where each observation is an element from the state space $[m]$. The integer N is called the sample size. When all observations are independent and identically distributed, the data can then be summarized in a data vector $u = (u_1, \ldots, u_m)$. u_k is the number of indices j such that $y_j = k$ and therefore $u_1 + u_2 + \ldots + u_m = N$. We can now define the likelihood function.

Definition 3: Given a statistical model $\mathbf{p}$ and a sequence y_i of N independent and identical distributed samples then the likelihood function $L(\theta)$ is given by

$$L(\theta) = p_{y_1}(\theta)p_{y_2}(\theta)\ldots p_{y_N}(\theta) = \prod_{i=1}^{m} p_i(\theta)^{u_i}. \quad (1)$$

This function depends on the parameter vector θ and u and is hence called the likelihood function. Note that it is the assumption of independent and identical distributed observations that allows us to factorize the likelihood. Any reordering of the observations leads to the same data vector u and has therefore no effect. Multiplying probabilities leads to very small numbers which could lead on a computer to numerical underflow. By taking the logarithm of (1) the expression is reduced to

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{m} u_i \log p_i(\theta) \quad (2)$$

which effectively transforms the product of probabilities into a sum. This takes care of the numerical underflow problem. The maximum log-likelihood estimate $\hat{\theta}$ is the solution of the following optimization problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ l(\theta) \quad (3)$$

which is equivalent with maximizing L since the logarithm is a monotonic function. (3) is solved by taking the partial derivatives of $l(\theta)$ to each θ_i and equating these to zero. This results in the following multivariate polynomial system

$$\begin{cases} \frac{\partial l(\theta)}{\partial \theta_1} = \sum_i \frac{u_i}{p_i}\frac{\partial p_i}{\partial \theta_1} = 0 \\ \vdots \\ \frac{\partial l(\theta)}{\partial \theta_n} = \sum_i \frac{u_i}{p_i}\frac{\partial p_i}{\partial \theta_n} = 0 \end{cases} \quad (4)$$

of n equations in n unknowns. The highest degree that occurs in the polynomial system (4) is denoted by d_0. Note that the dependencies of p_i on θ are dropped in the notation. A polynomial system like this typically has many solutions.

III. Solving Polynomial Systems as Eigenvalue Problems

This section sets out to describe the algorithm we have developed which finds all solutions of (4), including the global maximum. In what follows we suppose the solution set of (4) is zero-dimensional. This means that the solution set consists solely of isolated points. A simple example of a non-zero-dimensional solution set is the intersection of two planes. The first step is to write the polynomial system in matrix form and in order to do this an ordering of monomials needs to be defined. Note that we can reconstruct the monomial $\theta^a = \theta_1^{a_1} \ldots \theta_n^{a_n}$ from the n-tuple of exponents $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$. Furthermore, any ordering $>$ we establish on the space $\mathbb{N}^n$ will give us an ordering on monomials: if $a > b$ according to this ordering, we will also say that $\theta^a > \theta^b$.

Definition 4: A monomial ordering $>$ is any relation $>$ on $\mathbb{N}^n$, or equivalently, any relation on the set of monomials $\theta^a, a \in \mathbb{N}^n$, satisfying:

1) $>$ is a total (or linear) ordering on $\mathbb{N}^n$.
2) if $a > b$ and $c \in \mathbb{N}^n$, then $a + c > b + c$.
3) $>$ is a well-ordering on $\mathbb{N}^n$. This means that every nonempty subset of $\mathbb{N}^n$ has a smallest element under $>$.

This definition can also be found in [13], along with a number of monomial orderings which are relevant for the field of algebraic geometry. For our algorithm however we need to define our own monomial orderings.

Definition 5: Xelicographic Order. Let a and $b \in \mathbb{N}^n$. We say $a >_{xel} b$ if, in the vector difference $a - b \in \mathbb{Z}^n$, the

leftmost nonzero entry is negative. We will write $\theta^a >_{xel} \theta^b$ if $a >_{xel} b$.

Definition 6: Graded Xel Order. Let a and $b \in \mathbb{N}^n$. We say $a >_{grxel} b$ if

$$|a| = \sum_{i=1}^{n} a_i > |b| = \sum_{i=1}^{n} b_i, \text{ or } |a| = |b| \; and \; a >_{xel} b.$$

For example, $(2,0,0) >_{grxel} (0,0,1)$ because $|(2,0,0)| > |(0,0,1)|$. Likewise, $(0,1,1) >_{grxel} (2,0,0)$ because $(0,1,1) >_{xel} (2,0,0)$. It can be easily verified that the xelicographic and graded xel orderings satisfy the 3 conditions of definition 4. We can now define a monomial basis vector.

Definition 7: A monomial basis vector k in n unknowns of degree d is the column vector of all monomials in n unknowns, graded xel ordered from degree 0 up to d.
A monomial basis in 2 unknowns of degree 3 has, for example, the following structure

$$\begin{bmatrix} 1 \\ \theta_1 \\ \theta_2 \\ \theta_1^2 \\ \theta_1\theta_2 \\ \theta_2^2 \\ \theta_1^3 \\ \theta_1^2\theta_2 \\ \theta_1\theta_2^2 \\ \theta_2^3 \end{bmatrix}. \tag{5}$$

We can now write the polynomial system in matrix form

$$M \; K = 0 \tag{6}$$

where $M \in \mathbb{R}^{nxk}$ is called the coefficient matrix and $K \in \mathbb{C}^{kxs}$ the kernel. $k = \binom{n+d_0}{n}$ equals the number of monomials in n unknowns up to a degree d_0. This number is also the dimension of $\mathbb{R}[x_1, \ldots, x_n]_{\leq d_0}$, the vector space that consists of all polynomials with real coefficients in n unknowns up to a degree d_0. s denotes the total number of solutions of the polynomial system. Each column of K is a monomial basis vector, evaluated in a specific solution of the polynomial system. Each row of M then contains the coefficients of a polynomial, according to the graded xel ordering of K. The row space of M consists of all polynomials of a degree maximal equal to d_0 which are a linear combination of the polynomials of (4). This vector space is denoted by $I_{\leq d_0}$. Note that all polynomials of $I_{\leq d_0}$ also vanish on K. The idea now is to to add polynomials of a higher degree to M that still vanish on K. This is achieved by taking the original polynomials of M and multiplying them with all possible monomials of degree 1. Matrix equation (6) then becomes

$$\begin{bmatrix} M \\ M_{d_0+1} \end{bmatrix} \begin{bmatrix} K \\ K_{d_0+1} \end{bmatrix} = 0 \tag{7}$$

from which it is clear that both the rows and columns of M are extendend while for K only the number of rows is increased (since we assume the number of solutions does not change). K_{d_0+1} contains all monomials of degree d_{0+1} and n unknowns. M_{d_0+1} contains the same coefficients as M but because of the multiplication they occupy columns which are more to the right. This procedure can be repeated indefinitely such that the polynomial system M is extended with polynomials of any degree d all vanishing on the same kernel K. For each of these extended polynomial systems we can again define a corresponding vector space $I_{\leq d}$. Now we define the Hilbert function of $I_{\leq d}$ as in [13].

Definition 8: The Hilbert function of $I_{\leq d}$ is the function on the nonnegative integers d defined by

$$\begin{aligned} HF(d) &= \dim \mathbb{R}[x_1, \ldots, x_n]_{\leq d} \,/\, I_{\leq d} \\ &= \dim \mathbb{R}[x_1, \ldots, x_n]_{\leq d} \; - \dim I_{\leq d} \\ &= k - \text{rank}(M) \end{aligned}$$

which is equal, by the rank-nullity theorem in linear algebra, to the nullity of M. The importance of the Hilbert function lies in the domain of algebraic geometry where it is linked to the dimension of the solution set of polynomial systems. Its use in this paper lies in the well-known theorem that from a certain degree $HF(d)$ converges to a fixed value s [13]. This implies in our linear algebra setting that the dimension of the kernel, and hence the number of solutions, will increase when the degree of M is extended as in (7). There is a certain degree however from where the number of solutions of (4) will stabilize at a certain value s. At this point it is then not necessary anymore to further increase the degree since all information on the solutions is captured in the M matrix. This 'sufficient' degree is called the index of regularity d_{reg} and has an upper bound d_{max} [14]

$$d_{reg} \leq n(d_0 + 1) - 1 \equiv d_{max}. \tag{8}$$

It is not possible to compute the kernel K directly but the right singular vectors Z corresponding to vanishing singular values of M span its null space and are therefore related to K by a linear transform V

$$K = Z \, V \tag{9}$$

where $V \in \mathbb{R}^{sxs}$ and of full rank. The rows of K obey a certain shift property. Multiplying rows of K with a specific monomial will map these rows to other rows of K. For example, multiplying the first three rows of (5) with θ_1

maps them to the $2^{\text{nd}}, 3^{\text{rd}}$ and 4th rows respectively. This shift property can be written, in general, as the following matrix equation

$$S_1\, K\, D \;=\; S_2\, K \tag{10}$$

where S_1 selects s independent rows of K, D is a diagonal matrix which contains the monomial by which $S_1 K$ is multiplied and S_2 selects the shifted rows of higher degree. Plugging (9) into (10) results in

$$S_1\, Z\, V\, D \;=\; S_2\, Z\, V. \tag{11}$$

This is a generalized eigenvalue problem where V are the eigenvectors and D contains the eigenvalues. Setting $S_1 Z = B$ and $S_2 Z = A$ we can write (11) as

$$B\, V\, D \;=\; A\, V.$$

Note that when B is invertible this problem can be reduced to the familiar eigenvalue decomposition

$$V\, D\, V^{-1} \;=\; B^{-1}\, A.$$

We now have all ingredients to formulate the algorithm.

Algorithm 1:

Input: polynomial system P

Output: kernel K

1: $M \leftarrow$ coefficient matrix of P up to degree d_{max}
2: $s \leftarrow$ nullity of M
3: $Z \leftarrow$ basis null space from SVD(M)
4: $S_1 \leftarrow$ row selection matrix for s linear independent rows of Z
5: $S_2 \leftarrow$ row selection matrix for shifted rows of $S_1\, Z$
6: $B \leftarrow S_1\, Z$
7: $A \leftarrow S_2\, Z$
8: $[V, D] \leftarrow$ solve eigenvalue problem $B\, V\, D \;=\; A\, V$
9: $K \leftarrow Z\, V$

Any monomial (or even a function of monomials) can be chosen to shift with in step 5. For convenience this is set to θ_1. Once K is reconstructed from Z in step 9, the solutions can be read off from the rows corresponding with monomials θ_1 up to θ_n.

IV. Example

In order to clarify the method an illustrative example is worked out in the following section. The following problem is tackled: given a sequence of DNA, determine whether it came from a CpG island or not. Cpg islands are genomic regions that contain a high frequency of sites where a cytosine (C) base in the DNA sequence is followed by a guanine (G) [15]. When in the human genome the CG dinucleotide (often written as CpG) occurs, the C nucleotide is typically chemically modified by methylation. This methyl-C in its turn has a high probability of turning into a T, with the consequence that CpG dinucleotides are more rare in the genome data than would be expected. Regions of DNA that contain a high frequency of CpG would therefore indicate that there is some selective pressure to keep them. This explains why Cpg islands are usually found around the promotors of many genes. They are typically a few hundred to a few thousand bases long.

In order to find a solution to the given problem we make the following simplification: instead of focusing specifically on the occurrence of CpG's we count the occurrences of C and G instead. Our state space in this case is four dimensional $[m] = \{\text{A}, \text{C}, \text{G}, \text{T}\}$. A mixture model of the DNA sequence is set up that mixes 3 distributions on $[m]$. Each one of these distributions represents a certain type of DNA: CG rich, CG poor and CG neutral. The first type, CG rich, stands for DNA that is rich in both the C and G bases. Therefore, the C and G bases sampled from this distribution will have higher probabilities to occur relative to those for A and G. In a similar fashion the CG poor and CG neutral type are characterized by specific probabilities. The complete model is summarized in table I and is obtained from [16] and [12].

In this example the following sequence of DNA is used

CTCACGTGATGAGAGCATTCTCAGA
CCGTGACGCGTGTAGCAGCGGCTCA.

which can be summarized in the data vector $u = (11, 14, 15, 10)$. As mentioned before, deciding whether this segment came from a GpG island is decided from the occurrences of both the C and G bases. This information is encoded in the mixing probabilities: θ_1, θ_2 and θ_3. These are the probabilities with which samples are drawn from either the CG rich, poor or neutral distribution respectively and comparing them relative to one another will allow us to answer the given problem. For any general mixture model of k mixtures, the probability of making a certain observation y is

$$p(y) \;=\; \sum_{i=1}^{k} \theta_i\, p(y|i)$$

where $p(y|i)$ is the probability to observe y given that it is sampled from distribution i. The probabilities of observing each of the bases A to T are therefore given by

$$\begin{aligned}
p(A) &= 0.15\,\theta_1 + 0.27\,\theta_2 + 0.25\,\theta_3 \\
p(C) &= 0.33\,\theta_1 + 0.24\,\theta_2 + 0.25\,\theta_3 \\
p(G) &= 0.36\,\theta_1 + 0.23\,\theta_2 + 0.25\,\theta_3 \\
p(T) &= 0.16\,\theta_1 + 0.26\,\theta_2 + 0.25\,\theta_3
\end{aligned}$$

Table I
PROBABILITIES FOR EACH OF THE DISTRIBUTIONS.

DNA Type	A	C	G	T
CG rich	0.15	0.33	0.36	0.16
CG poor	0.27	0.24	0.23	0.26
CG neutral	0.25	0.25	0.25	0.25

which can be reduced to

$$\begin{aligned} p(A) &= -0.10.\theta_1 + 0.02.\theta_2 + 0.25 \\ p(C) &= +0.08.\theta_1 - 0.01.\theta_2 + 0.25 \\ p(G) &= +0.11.\theta_1 - 0.02.\theta_2 + 0.25 \\ p(T) &= -0.09.\theta_1 + 0.01.\theta_2 + 0.25 \end{aligned}$$

since $\theta_1 + \theta_2 + \theta_3 = 1$. Our mixed model is therefore an algebraic statistical model. Each probability is described by a first order polynomial. The maximum likelihood estimators for θ_1, θ_2 and θ_3 are found from the following optimization problem

$$\underset{\theta_1,\theta_2,\theta_3}{\operatorname{argmax}}\, l(\theta)$$

where the log-likelihood is given by

$$l(\theta) = 11\log p(A) + 14\log p(C) + 15\log p(G) + 10\log p(T).$$

The polynomial system that corresponds with (4) for this problem is therefore

$$\begin{cases} \frac{\partial l(\theta)}{\partial \theta_1} &= \sum_{i=1}^{4} \frac{u_i}{p(i)} \frac{\partial p(i)}{\partial \theta_1} &= 0 \\ \frac{\partial l(\theta)}{\partial \theta_2} &= \sum_{i=1}^{4} \frac{u_i}{p(i)} \frac{\partial p(i)}{\partial \theta_2} &= 0. \end{cases}$$

Each of the $\frac{\partial p(i)}{\partial \theta_j}$ is a number in this example and clearing the denominators results in a polynomial system of degree 3 in 2 unknowns. The upper bound for the index of regularity is $d_{max} = n(d_0 - 1) + 1 = 5$ and M is a 12 by 21 matrix. The nullity of M is 9, hence there are 9 solutions. The 9 independent rows of the B matrix correspond with the monomials $1, \theta_1, \theta_2, \theta_1^2, \theta_1\theta_2, \theta_2^2, \theta_1^3, \theta_1^2\theta_2, \theta_1^4$. The rows of the A matrix correspond with the monomials $\theta_1, \theta_1^2, \theta_1\theta_2, \theta_1^3, \theta_1^2\theta_2, \theta_1\theta_2^2, \theta_1^4, \theta_1^3\theta_2, \theta_1^5$. Solving the generalized eigenvalue problem allows us then to reconstruct the kernel

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots \\ 0.52 & 3.12 & -5.00 & 10.72 & \dots \\ 0.22 & 3.12 & -15.01 & 71.51 & \dots \\ 0.27 & 9.76 & 25.02 & 115.03 & \dots \\ 0.11 & 9.76 & 75.08 & 766.98 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{matrix} 1 \\ \theta_1 \\ \theta_2 \\ \theta_1^2 \\ \theta_1\theta_2 \\ \vdots \end{matrix} .$$

which has 9 columns and 21 rows. Since our unknowns are probabilities their values need to be between 0 and 1. The only solution that satisfies this constraint is given by the first column of K and therefore $\hat{\theta} = (0.52, 0.22, 0.26)$. The mixing probability for the CG rich distribution is twice as big as the other mixing probabilities. One could therefore conclude that the sequence is more likely to come from a CpG island.

V. CONCLUSION

In this paper it was shown how for algebraic statistical models finding the maximum likelihood estimates is equivalent with finding the roots of a polynomial system. A new method was presented that finds all solutions to these problems by solving a generalized eigenvalue problem. Finding all solutions means that the global maximum is guaranteed to be found and local optima can be avoided. The algorithm was implemented in Matlab® but could be easily implemented using any numerical linear algebra platform (e.g. Lapack [17]).

This method is not limited in any way to algebraic statistical models. In fact, as soon as the maximum likelihood (or log-likelihood) estimation is equivalent to solving a polynomial system this method can be employed. Likewise, the assumption of having independent and identical distributed observations is not strictly necessary. It only allows to factorize the likelihood and hence reduces the complexity of writing down the polynomial system. It would be interesting to further investigate for which other discrete statistical models maximum likelihood estimation is equivalent with polynomial system solving.

The size of the M matrix grows polynomially ($O(d^n)$) as the degree increases. Computing the SVD is the most intensive computational operation in the algorithm and has a time complexity of $O(\min\{nk^2, n^2k\})$. This could potentially be problematic for the execution of the algorithm. Finding ways to reduce the size and storage requirements of M are therefore paramount. Fortunately, the M matrix is very sparse and structured (since the same coefficients get copied to entries more to the right). We believe these properties can be exploited to further optimize the algorithm.

Polynomial systems generally also have solutions at infinity. A simple example is, for example, the intersection of two parallel lines. These solutions have not been discussed in this paper but are in fact also included in the nullity of the M matrix. Special care is needed to deal with these spurious solutions and this will be the topic of future work.

ACKNOWLEDGMENTS

Kim Batselier is a research assistant at the Katholieke Universiteit Leuven, Belgium. Bart De Moor is a full professor at Katholieke Universiteit Leuven, Belgium. Research

supported by *Research Council KUL*: GOA/11/05 AM-BioRICS, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; *Flemish Government*: FWO: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC); IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; *Belgian Federal Science Policy Office*: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); *EU*: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); *Contract Research*: AMINAL; *Other*: Helmholtz: viCERP; ACCM; Bauknecht; Hoerbiger. I would also like to thank Prof. Johan Suykens for reading this paper and providing me with helpful comments.

REFERENCES

[1] J. Aldrich, "R. A. Fisher and the Making of Maximum Likelihood 1912-1922," *Statistical Science*, vol. 12, no. 3, pp. 162–176, Aug 1997.

[2] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, pp. 309–368, 1922.

[3] S. Levinson, L. Rabiner, and M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.

[4] M. Bishop and E. Thompson, "Maximum-Likelihood Alignment of DNA-Sequences," *Journal of Molecular Biology*, vol. 190, no. 2, pp. 159–165, Jul 20 1986.

[5] J. Henderson, S. Salzberg, and K. Fasman, "Finding Genes in DNA with a Hidden Markov Model," *Journal of Computational Biology*, vol. 4, no. 2, pp. 127–141, Sum 1997.

[6] J. Felsenstein and G. Churchill, "A Hidden Markov Model approach to Variation Among Sites in Rate of Evolution," *Mol Biol Evol*, vol. 13, no. 1, pp. 93–104, 1996.

[7] A. Siepel and D. Haussler, "Combining Phylogenetic and Hidden Markov Models in Biosequence Analysis," *Journal of Computational Biology*, vol. 11, no. 2-3, pp. 413–428, 2004, 7th Annual International Conference on Computational Biology, Berlin, Germany, Apr 10-13, 2003.

[8] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via Em Algorithm," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 39, no. 1, pp. 1–38, 1977.

[9] W. Hastings, "Monte-Carlo Sampling Methods using Markov Chains and their Applications," *Biometrika*, vol. 57, no. 1, pp. 97–&, 1970.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[11] C. Geyer, "Markov-Chain Monte-Carlo Maximum-Likelihood," in *Computing Science and Statistics*, Keramidas, Em, Ed., 1991, pp. 156–163, 23rd Symp on the Interface between Computing Science and Statistics - Critical Applications of Scientific Computing : Biology, Engineering, Medicine, Speech, Seattle, Wa, Apr 21-24, 1991.

[12] L. Pachter and B. Sturmfels, Eds., *Algebraic Statistics for Computational Biology*. Cambridge University Press, August 2005.

[13] D. A. Cox, J. B. Little, and D. O'Shea, *Ideals, Varieties and Algorithms*, 3rd ed. Springer-Verlag, 2007.

[14] M. Giusti, "Combinatorial Dimension Theory of Algebraic-Varieties," *Journal of Symbolic Computation*, vol. 6, no. 2-3, pp. 249–265, Oct-Dec 1988.

[15] A. Bird, "Cpg Islands as Gene Markers in the Vertebrate Nucleus," *Trends in Genetics*, vol. 3, no. 12, pp. 342–347, Dec 1987.

[16] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis*, eleventh ed., C. U. Press, Ed., 2006.

[17] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.