Maximum Likelihood Estimation of Ordered Multinomial Probabilities by Geometric Programming

Johan Lim, Xinlei Wang, and Wanseok Choi

December 2006; Revision October 2008

Abstract

We propose an efficient method to compute the maximum likelihood estimator of ordered multinomial probabilities. Using the monotonicity property of the likelihood function, we reformulate the estimation problem as a geometric program, a special type of mathematical optimization problem, which can be transformed into a convex optimization problem, and then solved globally and efficiently. We implement a numerical study to illustrate its computational merits in comparison to the m-PAV algorithm proposed by Jewell and Kalbfleisch (2004). We also apply our proposed method to the current status data in Jewell and Kalbfleisch (2004).

1 Introduction

This paper is motivated by the recent work of Jewell and Kalbfleisch (2004) on maximum likelihood estimation of ordered multinomial parameters. They studied a cross sectional dataset containing the age and menopausal status of 3581 female respondents between the year 1960 and 1962 from the Health Examination Survey conducted by the National Center for Health Statistics. The main problem, as in McMahon and Worcester (1966) and Krailo and

Pike (1983), is the estimation of survival probabilities in the presence of two competing risks, natural menopause versus operative menopause (i.e., menopause caused by surgery). By considering nonparametric estimation of the survival function related to each risk, they formulated the problem into obtaining the maximum likelihood estimator (MLE) of multinomial probabilities subject to constraints, where the cross-sectional data naturally induce order constraints on the probabilities of menopause caused by each risk among different age groups.

To obtain the MLE under order constraints, Jewell and Kalbfleisch (2004) proposed a modified pooled-adjacent violator (m-PAV) algorithm. The m-PAV algorithm iteratively estimates the age-specific probabilities of natural and operative menopause using a modified one dimensional PAV algorithm, which pools adjacent violators (of monotone constraints) by maximizing the local likelihood function of the violators. Thus, it needs to solve many local optimization problems, each of which is non-linear and not fast computationally. See Section 2 for further review on this.

In this paper, we suggest an alternative method to compute the MLE of ordered multinomial probabilities and compare it with the m-PAV algorithm. Our method is based on *geometric programming*, a special case of nonlinear programming. A geometric program (GP) can be transformed into a convex optimization problem, then solved globally and efficiently by modern interior-point methods. See Boyd et al. (2004) for details.

It should be mentioned that in the literature, geometric programming had been used to solve several interesting problems related to maximum likelihood estimation of multinomial probabilities. Alldredge and Armstrong

(1974) considered the problem of estimating overlap sizes created by interlocking sampling schemes. Mazumdar and Jefferson (1983) considered three examples: (1) estimation of gene frequencies of A, B, O from frequencies of the four blood types O, A, B and AB; (2) estimation of probabilities p and p' by maximizing a specific likelihood function $L(p, p') = p^a(1-p)^{A-a}p'^b(1-p')^{B-b}(pp')^c(1-p'p)^{C-c}$; (3) estimation of the success probabilities $p_1, \ldots, p_k$ when sums of k independent Bernoulli random variables are observed. Briker, Kortanek, and Xu (1997) considered estimation of cell probabilities in a two-dimension table under the constraints on local odd ratios. All these earlier applications of GP were restricted to specific and small-sized problems, perhaps due to the fact that solution methods for GP were very slow at the time. As will be seen in next section, the problem we solve is more general and harder in nature, with many existing applications from various fields. More attractively, recent technical breakthroughs in geometric programming allow us to handle large-sized data easily.

The paper is organized as follows. In Section 2, we formally state the problem of estimating ordered multinomial probabilities, and discuss the existing solution (i.e., the m-PAV algorithm). In Section 3, we briefly introduce geometric programming and then relax the problem under consideration to a GP. Section 4 shows two numerical examples, in which we compare the performance of the m-PAV algorithm and the proposed method. The first example is a re-analysis of the current status data in Jewell and Kalbfleisch (2004). The second studies the case with large number of groups. Section 5 concludes with a discussion.

2 The Problem and Existing Solution

Suppose that $X_1, \ldots, X_I$ are independent multinomial random vectors, each with J possible outcomes. That is, for $1 \leq i \leq I$, $X_i = (X_{i1}, \ldots, X_{iJ})$ has a multinomial distribution with known index n_i and probabilities $p_i = (p_{i1}, \ldots, p_{iJ})$ satisfying $\sum_{j=1}^{J} p_{ij} = 1$. Also, for some $j \in \{1, \ldots, J-1\}$, p_{ij}s are known to satisfy the order constraints

$$0 \leq p_{1j} \leq \cdots \leq p_{Ij} \leq 1. \tag{1}$$

Because the equality $\sum_{j=1}^{J} p_{ij} = 1$ holds for each i, there exists at least one j for which (1) does not hold. Hence in (1), without loss of generality, we assume $j \neq J$.

Given the observed cell counts x_{ij}s, the likelihood function is given by

$$\mathcal{L}(p_1, \ldots, p_I) = \prod_{i=1}^{I} \prod_{j=1}^{J} p_{ij}^{x_{ij}}. \tag{2}$$

The MLE of p_{ij}s is the solution to the optimization problem

$$\begin{array}{ll} \max & \mathcal{L}(p_1, \ldots, p_I) \\ \text{subject to} & 0 \leq p_{1j} \leq \cdots \leq p_{Ij} \leq 1, \quad \text{for some } j \neq J \\ & \sum_{j=1}^{J} p_{ij} = 1 \quad \text{for all } i \end{array}. \tag{3}$$

This problem is a signomial programming problem that is very difficult to solve in general (Boyd and Vandenberghe, 2003). However, it is of great practical importance since it is well related to current status data with competing risks, which have caught substantial attention recently (Jewell, Van Der Laan and Henneman, 2003).

Jewell and Kalbfleisch (2004) solved the problem iteratively in j through the m-PAV algorithm. For example, in the case of $J = 3$, let $X_i \sim$ multinomial $(n_i, p_{i1}, p_{i2}, p_{i3})$ for $1 \leq i \leq I$, and constraints of the form (1) hold for $j = 1, 2$. The m-PAV algorithm iterates between the following two steps until convergence:

(i) hold $(p_{i2})_{i=1}^{I}$ fixed at its estimate from the previous step (or its initial value if this is the first iteration), and maximize the likelihood function over $(p_{i1})_{i=1}^{I}$ under the order constraints on $(p_{i1})_{i=1}^{I}$;

(ii) hold $(p_{i1})_{i=1}^{I}$ fixed at its estimate from (i), and maximize the likelihood function over $(p_{i2})_{i=1}^{I}$ under the order constraints on $(p_{i2})_{i=1}^{I}$.

In step (i), the objective function of $\big(p_{11}, \ldots, p_{I1}\big)$ is given by

$$\begin{aligned} \phi\big(p_{11}, \ldots, p_{I1}\big) &= \sum_{i=1}^{I} \Big\{ x_{i1} \log p_{i1} + x_{i3} \log\big(1 - p_{i1} - p_{i2}\big) \Big\} \\ &= \sum_{i=1}^{I} \Big\{ x_{i1} \log p_{i1} + x_{i3} \log\big(c_i - p_{i1}\big) \Big\} \qquad (4) \end{aligned}$$

subject to the constraints $p_{11} \le \cdots \le p_{I1}$ and $0 \le p_{i1} \le c_i$, where $c_i = 1 - p_{i2}$ are constants satisfying $0 \le c_I \le \cdots \le c_1 \le 1$. To solve (4), the authors proposed a modification of the original PAV algorithm (Barlow et al. 1972): if the current estimates of $\{p_{i1}, i \in \alpha\}$ violate the isotonic constraint for the set α of consecutive integers from $\{1, \ldots I\}$, then pool the estimates of $\{p_{i1}, i \in \alpha\}$ by using the solution $\hat{q}$ to the following univariate maximization problem

$$\max_q \quad \textstyle\sum_{i \in \alpha} \{x_{i1} \log q + x_{i3} \log(c_i - q)\}. \qquad (5)$$

Here, unlike the classical isotonic regression, an explicit expression of the solution to (5) is not available because c_i varies over i. Thus, (5) can only be solved numerically. The details of step (ii) are the same as those of step (i).

It is easy to see that the m-PAV algorithm solves the overall multi-dimensional optimization problem through solving *many* one dimensional maximization problems in (5) iteratively. The number of the subproblems is proportional to the number of adjacent violators, and increases as the group number I increases. Though these subproblems are easier to solve than the overall problem (3), each of them is still nonlinear without an explicit solution, and so requires additional computation. Since we need to solve a large number of such univariate subproblems, the m-PAV algorithm might not be computationally efficient, especially when there exist many groups (i.e., large I).

The m-PAV algorithm also requires a special treatment of boundary probability estimation, especially in the case of zero observation at boundaries. Users need solve the optimization problem for boundary observations separately and then insert the estimates in the main m-PAV procedure.

In this paper, instead of solving (3) directly, we consider a relaxed problem,

$$\begin{array}{ll} \text{minimize} & \mathcal{L}^{-1}(p_1, \ldots, p_I) \\ \text{subject to} & 0 \leq p_{1j} \leq p_{2j} \leq \cdots \leq p_{Ij} \leq 1, \quad \text{for some } j \neq J \\ & \sum_{j=1}^{J} p_{ij} \leq 1 \quad \text{for all } i \end{array}, \tag{6}$$

which, we claim, is easy to solve via geometric programming, but gives the same solution as that to (3). Note that minimizing the inverse of the likelihood in (6) is equivalent to maximizing the likelihood in (3).

As will be shown later, our method is faster than the m-PAV algorithm and requires no additional effort in boundary probability estimation.

3 The GP-based Method

3.1 Geometric program

We first give a brief description of geometric programming.

We start with two classes of functions, *monomials* and *posynomials*, which define a geometric program. Let $x_1, \ldots, x_n$ denote n real positive variables, and $x = (x_1, \ldots, x_n)$ be a vector with components x_i. A *monomial* is a real valued function f of x with the form

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \tag{7}$$

where $c > 0$ and $a_i \in \mathbb{R}$. A *posynomial* is the sum of one or more monomials, namely

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}, \tag{8}$$

where each c_k is positive. Note that any monomial is a posynomial.

A *geometric program* is an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 1, \quad i = 1, \ldots, u, \\ & g_j(x) = 1, \quad j = 1, \ldots, v, \end{array}$$

where f_is are posynomials for $i = 0, \ldots, u$, all g_js are monomials, and x is the vector of positive optimization variables x_is.

A GP can be easily converted to *a nonlinear convex optimization problem*, i.e., a problem with a convex objective function, convex inequality constraints, and linear equality constraints. To do this, we use logarithmic transformation of the variables x_i, $y_i = \log x_i$ (so $x_i = e^{y_i}$). Instead of minimizing the objective f_0, we minimize its logarithm $\log f_0$. We replace the inequality constraints $f_i \leq 1$ with $\log f_i \leq 0$, and the equality constraints $g_i = 1$ with $\log g_i = 0$. The transformation results in the following problem

$$\begin{array}{ll} \text{minimize} & \log f_0(e^y) \\ \text{subject to} & \log f_i(e^y) \leq 0, \quad i = 1, \ldots, u, \\ & \log g_j(e^y) = 0, \quad j = 1, \ldots, v, \end{array} \tag{9}$$

with variables $y = (y_1, \ldots, y_n)$. Here, we use e^y to express componentwise exponentiation: $(e^y)_i = e^{y_i}$.

It can be verified that the logarithm of a posynomial of x is convex in $\log x$, and the logarithm of a monomial of x is linear in $\log x$. Thus, the above problem is a convex optimization problem for which efficient solution methods are well developed (Boyd and Vandenberghe, 2004; Nesterov and Nemirovsky, 1994; Norcedal and Wright, 1999). In particular, interior-point methods are very efficient and robust for convex problems. For example, standard interior-point algorithms can solve a GP with $1,000$ variables and $10,000$ constraints within minutes, on a small desktop computer (Boyd et al., 2004). It is also possible to optimize a GP solver for a particular application using its special structure to gain more efficiency. High-quality implementation of the primal-dual interior-point method for GPs is available from various existing solvers. Examples include CVX (Grant, Boyd and Ye, 2005), GGPLAB (Mutapcic, Koh, Kim and Boyd, 2006), and YALMIP (Löfberg, 2003), which all have simple interfaces that recognize and solve GPs.

3.2 GP relaxation of the main problem

For any GP, the objective and inequality constraint functions must be posynomials, and the equality constraint functions must be monomials. In our main problem (3), the objective function $\prod_{i=1}^{I}\prod_{j=1}^{J} p_{ij}^{x_{ij}}$ is a monomial of p_{ij}s, and the inequality constraints in (1) are monomials. Hence, they are all posynomials. However, (3) is not a GP, since its equality constraints, $\sum_{j=1}^{J} p_{ij} = 1$ for all i, are posynomials rather than monomials. Fortunately, the relaxed problem (6) is a GP. We now establish the equivalence between (3) and (6), by showing that (6) is optimized only when "=" is attained in the inequalities $\sum_{j=1}^{J} p_{ij} \leq 1$.

It can be seen that the posynomials $\sum_{j=1}^{J} p_{ij}$ are *monotone increasing* in p_{ij}s, the objective to be minimized in (6) is decreasing in p_{ij}s, and no p_{iJ} appears in the inequality constraints of the form (1). Thus, at the optimal point, each inequality constraint $\sum_{j=1}^{J} p_{ij} \leq 1$ must be tight. In other words, for any $\overline{p}_i = \big(\overline{p}_{i1}, \ldots, \overline{p}_{iJ}\big)$, if the inequality $\sum_{j=1}^{J} \overline{p}_{ij} < 1$ is strict, then $\overline{p}_i$ cannot be optimal since $\hat{p}_i$ with $\widehat{p}_{ij} = \overline{p}_{ij}$ for $j = 1, 2, \ldots, J-1$, and $\widehat{p}_{iJ} = \overline{p}_{iJ} + \big(1 - \sum_{j=1}^{J} p_{ij}\big)$ is feasible for (6) with $\sum_{j=1}^{J} \widehat{p}_{ij} = 1$ but

$$\mathcal{L}^{-1}\big(\overline{p}_1, \ldots, \overline{p}_i, \ldots, \overline{p}_I\big) > \mathcal{L}^{-1}\big(\overline{p}_1, \ldots, \widehat{p}_i, \ldots, \overline{p}_I\big).$$

Hence, problems (3) and (6) are equivalent, and so we can solve the original problem (3) through the tractable GP (6).

4 Examples

To compare our new method with the m-PAV algorithm, we conduct two numerical studies. The first is to re-analyze the current status data in Jewell and Kalbfleisch (2004), using the proposed GP-based method; in the second study, we focus on large scale problems and compare the computational efficiency of the methods. Note that (3) is a convex optimization problem, whose solution is unique. Thus, the estimates from the proposed method should be the same as those from m-PAV. We show below our method is much faster than m-PAV.

Our experiments were implemented in a 1.80GHz Pentium IV computer using the software MATLAB. The code is available at `http://eclass.yonsei.ac.kr/johanlim/multiMLE.html`. In MATLAB, a GP is solved through calling external functions ”gpsolve” or ”gpcvx”, of which inputs are the coefficients and exponents of posynomials (monomials) of the problem.

4.1 Current status data

The current status data we use contain 3581 female respondents of age between 26 and 59 in the Health Examination Survey, which can be found in Table 1 of Jewell and Kalbfleisch (2004). For each age group i, the data consist of four columns $(n_i, x_{i1}, x_{i2}, x_{i3})$, where n_i is the total number of women in the group, x_{i1}, x_{i2} and x_{i3} are the number of women with natural menopause, operative menopause and nonmenopause, respectively. Here $I = 26$ and $J = 3$ in the setting of Section 2.

Following Jewell and Kalbfleisch (2004), we assume that (x_{i1}, x_{i2}, x_{i3}) is from a multinomial distribution with $\big(n_i, p_{i1}, p_{i2}, p_{i3}\big)$. The probability p_{i1} (or p_{i2}) is the probability that each individual has natural (or operative) menopause before age i. Hence, both p_{i1} and p_{i2} are monotone increasing in i.

The maximum likelihood estimate is the solution to

$$\begin{array}{ll} \text{maximize} & \mathcal{L}\big(p_1, \ldots, p_I\big) = \prod_{i=1}^{I} \; p_{i1}^{x_{i1}} p_{i2}^{x_{i2}} p_{i3}^{x_{i3}} \\ \text{subject to} & p_{1j} \le p_{2j} \le \cdots \le p_{Ij} \quad \text{for } j = 1, 2. \\ & p_{i1} + p_{i2} + p_{i3} \le 1 \quad \text{for } i = 1, 2, \ldots, 26. \end{array} \tag{10}$$

We apply the GP-based method to solve the above optimization problem. The GP of this example has 78 variables and 76 constraints including 50 order constraints. The duality gap is set to 10^{-13}, which means that the maximum likelihood value we find is within the gap 10^{-13} from the true maximum likelihood value. The number of Newton steps required to solve the GP is 86 and the CPU time is about 1.35 seconds in total. In contrast, the m-PAV algorithm needs to solve about 106 univariate subproblems (5), each of which requires approximately 40 Newton steps; the CUP time is 52.80 seconds in total. In fact, the computing time of m-PAV relies on the algorithm chosen to solve (5). Since each of the subproblems is a GP too, we use geometric programming to solve it, which can be much faster than any other algorithm.

The estimates from our method, as expected, are the same as those reported in Jewell and Kalbfleisch (2004).

4.2 Large scale problems

To investigate the computational efficiency of the proposed approach in large scale problems, we generate 50 date sets of 120 time points with three competing events. So $I = 120$, $J = 3$, and each data set is a 120×3 matrix. The i-th row vector of each data set is independently generated from the multinomial distribution with $n_i = 200$ and $p_i = \big(0.004i, 0.003i, 1 - 0.007i\big)$.

We apply the GP-based method and the m-PAV algorithm to find the ordered multinomial probabilities for each data set. Here, when using the m-PAV algorithm, we do not separately solve the boundary problem if zero boundary observation occurs, since doing so requires extra programming effort. As a result, the reported CPU time for the m-PAV algorithm is shorter than its true CPU time. Table 2 reports the average CPU time and standard deviation in computing the MLEs of the 50 data sets. It shows that the GP-based method is much faster than the m-PAV algorithm.

	m-PAV	GP-method
mean (std)	743.9960 (111.7539) sec.	3.4095 (0.2614) sec.

5 Discussion

In this paper, we have described a non-iterative method for estimating ordered multinomial probabilities, based on geometric programming. As opposed to the m-PAV approach, we have demonstrated through two numerical studies that the GP-based method is computationally fast and easy to use. It relies on interior-point methods for GPs, which are very robust, in addition to great efficiency. It does not require any care of the boundary problem. With a GP solver and parser such as GGPLAB (available freely online), it is straightforward to translate a problem of estimating ordered multinomial probabilities into a standard GP format, and so coding a GP is easy for users.

We should emphasize that in our numerical studies, we speed up the m-PAV algorithm by using GP to solve each subproblem involved. Doing so directly addresses a potential concern that the CPU comparison could be misleading if the GP code is more highly optimized than the m-PAV code. It also demonstrates that the estimation problem we consider can be solved more efficiently by simultaneously optimizing all of the decision variables rather than by cyclically optimizing subsets of the decision variables, as kindly pointed out by the Associate Editor. This might be a useful observation, as the latter approach (sometimes called Dykstra's cyclical projections algorithm) is widely used in Statistics.

Finally, we note that the proposed method can be found useful in most situations where the m-PAV algorithm can be applied, such as applications in carcinogenicity testing, demography, economics, and epidemiology (Jewell and van der Laan, 2003).

References

Alldredge, J. and Armstrong, D. (1974), Maximum likelihood estimation for multinomial distribution using geometric programming. *Technometrics*, **16**, 585-587.

Barlow, R. E. and Bartholomew, D. J. and Bremner, J. M. and Brunk, H. D. (1972), *Statistical Inference under Order Restrictions*, New York: Wiley.

Boyd, S., Kim, S., Vandenberghe, L., and Hassibi, A. (2004), A tutorial on geometric programming. To appear in *Optimization and Engineering*, 2006. Available from `http://www.stanford.edu/~boyd/gp_tutorial.html`.

Boyd, S. and Vandenberghe, L. (2003), *Convex Optimization.* Cambridge University Press.

Briker, D., Kortanek, K., and Xu, L. (1997), Maximum likelihood estimates with order restrictions on probabilities and odds ratios: a geometric programming approach. *Journal of Applied Mathematics and Decision Sciences*, **1**, 53-65.

Grant, M., Boyd S., and Ye, Y. (2005), CVX: Matlab software for disciplined convex programming. Available from `www.stanford.edu/~boyd/cvx/`.

Jewell, N.P., Van Der Laan, M., and Henneman, T. (2003), Nonparametric estimation from current status data with competing risks. *Biometrika*, **90**, 183-197.

Jewell, N.P. and Kalbfleisch, J. (2004). Maximum likelihood estimation of ordered multinomial parameters. *Biostatistics*, **5**, 291-306.

Krailo, M.D. and Pike, M.C. (1983). Estimation of the distribution of age at natural menopause from prevalence data. *American Journal of Epidemiology*, **117**, 356-361.

Löfberg, J. (2003), YALMIP. Yet Another LMI Parser. Version 2.4. Available from `http://control.ee.thz.ch/~joloef/yalmip.php`.

Macmahon, B. and Worcester, J. (1966). Age at menopause, United States 1960-1962. *National Center for Health Statistics, Series 11: Data from the National Health Survey, no 19* Washington, DC: DHEW Publications no (HSM) 66-1000.

Mazumdar, M. and Jefferson, T. (1983). Maximum likelihood estimates for multinomial probabilities via geometric programming. *Biometrika*, **70**, 257-261.

Mutapcic A., Koh K., Kim S., and S. Boyd (2006), GGPLAB: A Matlab toolbox for geometric programming. Available from `www.stanford.edu/~boyd/ggplab/`.

Nesterov, Y. and Nemirovsky, A. (1994), *Interior-Point Polynomial Methods in Convex Programming*, volume 13 of *Studies in Applied Mathematics.* SIAM Philadelphia, PA.

Norcedal, J. and Wright, S. (1999), *Numerical Optimization*, Springer Series in Operations Research. Springer.