On the Estimation of Markov Random Field Parameters

Borges, C.F.

C.F. Borges, On the Estimation of Markov Random Field Parameters, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 3, March 1999, pp. 216-224.

http://hdl.handle.net/10945/38064

Downloaded from NPS Archive: Calhoun
On the Estimation of Markov Random Field Parameters

Carlos F. Borges, Member, IEEE

Abstract—We examine the histogram method proposed in [1] for estimating the parameters associated with a Markov random field. This method relies on the estimation of the local interaction sums from histogram data. We derive an estimator for these quantities that is optimal in a well-defined sense. Furthermore, we show that the final step of the histogram method, the solution of a least-squares problem, can be done substantially faster than one might expect if no equation culling is used. We also examine the use of weighted least-squares and see that this seems to lead to better estimates even with small amounts of data.

Index Terms—Markov random fields, parameter estimation.

1 INTRODUCTION

We will consider 2D random fields defined over a finite rectangular $N \times M$ lattice of pixels arranged in a toroidal topology. Given a random field $Y$, we denote a given realization by $y$ and the numerical value of the random variable $Y_{ij}$ associated with the $i, j$th pixel in this realization by $y_{ij}$. Each pixel in the image has an associated set of neighbors denoted by $\eta$, which is a subset of other pixels from the image (usually pixels that are nearby the $i, j$th pixel in some sense). The random field $Y$ is a Markov random field (MRF) if $P(Y = y) > 0$ for all $y$ and further

$$P(Y_{ij} = y \mid Y_{\eta(j)} = y_{\eta}) \text{ for all } (i, j) \in \eta,$$

$$P(Y_{ij} = y \mid Y_{\eta(i)} = y_{\eta}) \text{ for all } (s, t) \in \eta.$$

In other words, all realizations of the random field are possible and each pixel in the image is conditionally independent of all other pixels in the image given the values of its neighbors. Throughout this paper, we will focus on the binary autobinomial MRF with a uniform second-order neighborhood structure (see [2], [3] for details). For this model, a pixel may take on values of either zero or one with the conditional probability density that $Y_{ij} = 1$ given the values of its neighbors being given by

$$P = \frac{e^v}{1 + e^v},$$

where

$$v = \beta_0 + \beta_1(y_{i-1,j} + y_{i,j+1}) + \beta_2(y_{i-1,j} + y_{i,j+1}) + \beta_3(y_{i-1,j} + y_{i,j+1}) + \beta_4(y_{i-1,j} + y_{i,j+1})$$

(1)

Derin and Elliot [1] make the important observation that with the following definitions,

$$v = \begin{bmatrix} y_{i-1,j} + y_{i,j+1} \\ y_{i-1,j} + y_{i+1,j} \\ y_{i-1,j} + y_{i+1,j} \\ y_{i-1,j} + y_{i,j+1} \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix},$$

then (1) can be rewritten in the following form:

$$v = \mathbf{v}^T \hat{\beta}.$$

The vector $v$ is sometimes called the local interaction vector, $\beta$ is the vector of Markov random field parameters, and the value $v$ is often called the local interaction sum.

2 ESTIMATING THE MARKOV RANDOM FIELD PARAMETERS

An important problem is to estimate the MRF parameter vector $\beta$ from one or more realizations of the MRF. There are two common methods for accomplishing this task. The first, known as the coding method (see Besag [2]), is a maximum-likelihood approach that requires the solution of a large system of nonlinear equations. The second, which we shall consider here, was first proposed by Derin and Elliot [1] and is known as the histogram method. This approach relies on a couple of clever observations. First, that there are only a small number of possible neighborhood configurations and an even more limited number of possible local interaction vectors. In particular, for the case under consideration, there are only $2^4 = 256$ possible neighborhood configurations. Moreover, since each of the last four elements in the local interaction vector can take on only the values zero, one, or two, there are only $3^4 = 81$ possible local interaction vectors. If we index these then, independent of particular pixel locations, we have 81 equations of the form $v_i^T \hat{\beta} = v_i$. Henceforth, we will say that a pixel has index $i$ if its local interaction vector is equal to $v_i$.

The second observation is that if we have a sufficiently large image, then we can use a histogram technique to estimate the probability

\[0162-8828/99/U.S. government work not protected by U.S. copyright\]
that a pixel whose index is \( i \) will be a one. Once we have estimated that value, then we can recover the associated local interaction sum associated with that index with

\[
\nu_i = \ln \frac{p_i}{1-p_i}.
\]

Finally, we should be able to find the MRF parameters by solving the system

\[
V_{\beta} = \nu,
\]

where

\[
V = \begin{bmatrix}
V_1^T & \cdots & V_{81}^T
\end{bmatrix}
\quad \text{and} \quad
\nu = \begin{bmatrix}
\nu_1 \\
\nu_2 \\
\vdots \\
\nu_{81}
\end{bmatrix}.
\]

In practice, one usually looks for a solution in the least-squares sense, since there will be errors in estimating the local interaction sums. Moreover, it may not be possible to estimate all of the \( \nu_j \) in which case some rows of the equation may be dropped. It is clear that there are two steps to this algorithm:

1) estimating the local interaction sums and
2) solving the least-squares problem.

We will discuss each of these in more detail in the rest of this paper.

### 3 Estimating the Local Interaction Sums

The histogram algorithm proceeds by looking at the entire image (or some subset of its pixels) and building an indexed histogram. In particular, for each pixel in the sample set, determine the pixels index and then keep running tallies:

1) \( s \) — the number of times we have observed a pixel of index \( i \) whose value is one and
2) \( t \) — the number of times we have observed a pixel of index \( i \) whose value is zero.

Once we have built the histogram, we must estimate the corresponding local interaction sums, \( \nu \). We will now examine three possible approaches to this problem. Note that to simplify notation, in the following estimate derivations we will assume we are working with a fixed index \( i \) and will drop the subscript.

#### 3.1 The Classical Estimate

The estimator proposed by Derin and Elliot [1] is based on first estimating the mean parameter, \( p \), of the Bernoulli random variable. Assuming that we have \( s + t \) observations of the random variable and that \( s \) of those observations are ones and \( t \) are zeros, then the standard estimate of the mean parameter is

\[
p = \frac{s}{s + t}.
\]

And hence a logical estimate of \( \nu \) is

\[
\mathcal{A}_{s,t} = \ln \frac{s}{s + t} = \ln \frac{s}{t}.
\]

This estimator is well-defined provided \( s, t > 0 \). Furthermore, if we consider the estimate \( \mathcal{A}_s \), as a function of its subscripts, then it is clear that this function is real-valued and skew-symmetric, which is consistent with the nature of the quantity being estimated. More importantly, since the sample mean in (3) is known to be a consistent estimate of \( p \), it can be shown that as \( s + t \to \infty \), the quantity \( \mathcal{A}_s \) itself is a consistent estimate of \( \nu \) provided it is defined and \( p \in (0, 1) \).

Unfortunately, there are problems with this approach. Most troublesome is the fact that the estimate is not defined if either \( s = 0 \) or \( t = 0 \), which is quite likely if we are given a small number of total observations \( s + t \) and is always true if we have fewer than two observations. Since we lose a row from (2) for each of these estimates that is not defined, this can lead to rank-deficiency and preclude solution of the least-squares problem. The beauty of this estimate is that it is given in closed form and is very easy to compute.

#### 3.2 The Minimum Mean Square Bias Estimate

In [4], Gurelli and Onural propose a clever alternative to alleviate some of the problems associated with the standard estimate. Their estimate is chosen so that it minimizes the mean square bias of the estimate over a range of possible values of the mean parameter \( p \). In particular, given \( n \) observations of the random variable, they define their estimate \( B_{n-\text{est}} \) so that it minimizes

\[
\sum_{j=1}^{N} \left[ \sum_{k=0}^{n} \frac{n}{k} \left( p^k (1-p)^{n-k} \right) - \ln \frac{p}{1-p} \right]^2.
\]

This approach eliminates some of the problems associated with the \( \mathcal{A}_s \) estimate, since \( B_{n-\text{est}} \) can be defined when \( s = 0, t = 0 \), and when \( s + t \leq 1 \). Unfortunately, these estimates are not available in closed form, although they need only be computed once, after which they can be retrieved from a table. It is proposed in [4] that the \( B_{n-\text{est}} \) estimates be used when \( 5 \leq s + t \leq 11 \) and that the estimates \( \mathcal{A}_s \) be used when \( s + t > 11 \). When \( s + t < 5 \), they suggest that the corresponding row be dropped from (2), although this is not necessary. Furthermore, it is suggested that one use

\[
p_j \in \{ p \mid p = 0.5 \pm (0.05 + 0.01k), k = 0, 1, 2, ..., 15 \}
\]

for \( N = 5, 6, ..., 10 \) and

\[
p_j \in \{ p \mid p = 0.5 \pm (0.05 + 0.01k), k = 0, 1, 2, ..., 25 \}
\]

for \( N = 11 \).

This author’s experiments indicate that this approach works fairly well. However, there are still difficulties because the \( \mathcal{A}_s \) estimator does not provide estimates when \( s = 0 \) or \( t = 0 \) in cases where \( s + t > 11 \), which does occur reasonably often. It is possible to simply drop such rows from (2) but one might prefer to extend the \( B_{n-\text{est}} \) estimate to cover these cases. Unfortunately, this does not work well because for large \( N \), the least squares problem in (4) becomes so ill-
conditioned that it is not possible to solve it reliably, even if one uses the most stable known singular value decomposition approach. Moreover, before the ill-conditioning swamps the problem, the computed $B_j$ estimates can become inconsistent (e.g., one sometimes encounters negative estimates of $B_j$ when $s > t$).

### 3.3 A New Estimate

Recall that we are interested in developing an estimate $\hat{v}$ for the true value of

$$v = \ln \frac{p}{1-p}$$

based on observations of the Bernoulli random variable. One common and useful criterion for developing such estimates is to consider the mean square error, defined to be

$$E[(\hat{v} - v)^2]$$

and choose the estimate that minimizes this quantity. If $\hat{v}$ satisfies this criterion, then it is called a minimum mean squared error (MMSE) estimator.

For the problem at hand, we are given $n$ independent observations of a Bernoulli random variable with parameter $p$. We will let

$$C_{k,n-k}$$

denote our estimate of $v$ given that $k$ of the $n$ observations were ones.

Since we have $n$ observations, the mean square error of the estimate is given by

$$E[(\hat{v} - v)^2] = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2.$$  \hfill (5)

Clearly, if we are given $p$, then the problem is easily solved. Since $p$ is not known, we will use a Bayesian approach to eliminate it from the problem. In particular, assume that $p$ is a random variable with known distribution and minimize the expected value of the quantity in (5) over all possible values of the parameter $p$. In particular, find values of $C_{k,n-k}$ for $k = 0, 1, 2, ..., n$ that minimize

$$E\left[\sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2\right],$$  \hfill (6)

where the expectation operation is with respect to $p$.

Because $p$ is the parameter for a Bernoulli random variable, it is convenient for our Bayesian approach to assume that $p$ is a uniform random variable on $[0, 1]$, that is $p \sim U(0, 1)$.

With these assumptions, (6) becomes

$$\int_{0}^{1} \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2 dp,$$

which can be written as

$$\sum_{k=0}^{n} \binom{n}{k} \int_{0}^{1} p^k (1-p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2 dp.$$  \hfill (8)

Clearly, this is a positively weighted sum of positive quantities, since each of the definite integrals in (8) is positive. Hence, it follows that choosing $C_{k,n-k}$ to minimize

$$\int_{0}^{1} p^k (1-p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2 dp$$

for each $k = 0, 1, 2, ..., n$ will give the desired solution. Now, expanding (9) gives

$$C_{k,n-k}^2 \int_{0}^{1} p^k (1-p)^{n-k} dp - 2C_{k,n-k} \int_{0}^{1} p^k (1-p)^{n-k} \ln \frac{p}{1-p} dp + \int_{0}^{1} p^k (1-p)^{n-k} \ln^2 \frac{p}{1-p} dp.$$  \hfill (10)

Differentiating (10) with respect to $C_{k,n-k}$ setting equal to zero gives the following equation for the critical point:

$$C_{k,n-k} = \frac{\int_{0}^{1} p^k (1-p)^{n-k} dp}{\int_{0}^{1} p^k (1-p)^{n-k} dp}.$$  \hfill (11)

This critical point represents a minimum, since differentiating (10) twice yields

$$2\int_{0}^{1} p^k (1-p)^{n-k} dp,$$

which is clearly positive.

To complete our derivation of the estimator, we need to evaluate the definite integrals in (11). We will consider the denominator first. Begin by recalling that the beta function (Euler’s integral of the first kind) is defined by

$$B(z, w) = \int_{0}^{1} t^{z-1}(1-t)^{w-1} dt.$$  

This function is symmetric, $B(z, w) = B(w, z)$, and clearly

$$\int_{0}^{1} p^k (1-p)^{n-k} dp = B(k+1, n-k+1).$$

Moreover, it is known that the beta function is related to the gamma function by

$$B(z, w) = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)},$$

which, using the fact that $\Gamma(n+1) = n!$ at integer values, implies that

$$\int_{0}^{1} p^k (1-p)^{n-k} dp = \frac{k!(n-k)!}{(n+1)!}.$$  

Now let us consider the definite integral in the numerator. We begin by noting that

2. Although certainly not necessary, as we can choose any distribution we like over the admissible interval $[0, 1]$ or any appropriately measurable subset.
\[
\int_0^1 p^k (1 - p)^{n-k} \ln \frac{p}{1-p} dp = \\
\int_0^1 p^k (1 - p)^{n-k} \ln p dp - \int_0^1 p^k (1 - p)^{n-k} \ln(1 - p) dp = \\
\int_0^1 p^k (1 - p)^{n-k} \ln p dp - \int_0^1 p^{n-k} (1 - p)^k \ln p dp
\]

Next, it can be shown that
\[
\int_0^1 p^k (1 - p)^{n-k} \ln p dp = \\
B(k + 1, n - k + 1)\left\{\Psi(k + 1) - \Psi(n + 2)\right\}
\]
where the psi function (also called the digamma function) is defined to be
\[
\Psi(z) = \frac{d}{dz} \ln \Gamma(z).
\]

It is known that at integer values of the argument, we have
\[
\Psi(n) = -\gamma + \sum_{k=1}^{n-1} \frac{1}{k},
\]
where \(\gamma\) is Euler’s constant.

We are led to the following formula for the definite integral in the numerator:
\[
\int_0^1 p^k (1 - p)^{n-k} \ln \frac{p}{1-p} dp = \\
B(k + 1, n - k + 1)\left\{\Psi(k + 1) - \Psi(n - k + 1)\right\} = \\
B(k + 1, n - k + 1)\left[\sum_{m=1}^{k} \frac{1}{m} - \sum_{m=1}^{n-k+1} \frac{1}{m}\right]
\]

And we have the following closed form for the MMSE estimate:
\[
C_{k,n-k} = \sum_{m=1}^{k} \frac{1}{m} - \sum_{m=1}^{n-k+1} \frac{1}{m}.
\]

This estimate \(C_{s,t}\) is a real-valued skew-symmetric function of its subscripts just like the classical estimate \(A_{s,t}\). Note that this formula is not suggested for actual computation. For practical purposes, the estimate \(C_{s,t}\) where \(s > t\) is found by evaluating the sum
\[
C_{s,t} = \sum_{m=1}^{s} \frac{1}{m}.
\]

In floating-point computations, this sum should be accumulated in reverse order to minimize the rounding error, that is,
\[
C_{s,t} = \frac{1}{s} + \frac{1}{s-1} + \ldots + \frac{1}{t+1}.
\]
If \(s = t\), then the estimate is always zero, and if \(s < t\), then we compute \(C_{s,t} = -C_{t,s}\).

It is of passing interest to note that if we form the estimates into a matrix \(C_{i,j} = C_{i-1,j-1}\), then
\[
C = s\text{e}^s - e^{s^T},
\]
where \(e\) is a vector of ones, and \(s\) is a vector of partial sums of the harmonic series, that is,
\[
s = [0 \ 1/3 \ 1/2 \ 11/6 \ 25/12 \ldots].
\]

One can also write \(C\) in terms of a Lyapunov transformation of the Hilbert matrix,
\[
C = LH - HL^T,
\]
where \(L\) is a matrix with ones below the main diagonal and zeros elsewhere and \(H\) is the Hilbert matrix (i.e. \(H_{ij} = 1 / (i + j - 1)\)).

### 3.3.1 The Mean Square Error of the \(C_{s,t}\) Estimate

Now that we have derived a simple formula for the estimate, we would like to compute the mean square error of the estimate given that we have \(n\) observations. Recall that the total mean square error from (7) is
\[
\int_0^1 \sum_{k=0}^n \left(\frac{n}{k}\right) p^k (1 - p)^{n-k} \left(C_{k,n-k} - \ln \frac{p}{1-p}\right)^2 dp.
\]

Expanding the square and rearranging yields
\[
\sum_{k=0}^n \left(\frac{n}{k}\right) p^k (1 - p)^{n-k} dp - 2C_{k,n-k} \int_0^1 p^k (1 - p)^{n-k} \ln \frac{p}{1-p} dp = \\
+ \int_0^1 \left[\sum_{k=0}^n \left(\frac{n}{k}\right) p^k (1 - p)^{n-k}\right] \ln^2 \frac{p}{1-p} dp
\]

Note that by virtue of the binomial theorem, the last integral simplifies to
\[
\int_0^1 \ln^2 \frac{p}{1-p} dp = \frac{\pi^2}{3}.
\]

Moreover, recall from (11) that
\[
\int_0^1 p^k (1 - p)^{n-k} \ln \frac{p}{1-p} dp = C_{k,n-k} \int_0^1 p^k (1 - p)^{n-k} dp.
\]

And so the total mean square error is given by
\[
\frac{\pi^2}{3} - \frac{1}{n+1} \sum_{k=0}^n C_{k,n-k}^2.
\]

Note that for values of \(n > 400\), the following is a good rational approximation to the total mean square error:
\[
\frac{1}{3 + 0.5n}.
\]

### 3.3.2 Asymptotic Behavior of the \(C_{s,t}\) Estimate

We now show that the \(C_{s,t}\) estimate exhibits the appropriate asymptotic behavior as the number of observations approaches infinity. In particular, we will show that, in the limit, it behaves like the \(A_{s,t}\) estimate as the number of observations gets large. As a starting point, recall that the limit
\[
\lim_{k \to \infty} \frac{1}{k} \sum_{j=1}^{k} \frac{1}{j} = -\ln k = \gamma
\]
exists and that its limiting value, \(\gamma\), is known as Euler’s constant.

Now consider the following difference:
linear equations for the estimates: setting the gradient to zero yields the following system of entries

\[ 0 = \lim_{k \to +\infty} \left( \sum_{j=1}^{s} \frac{1}{j} - \ln s k \right) - \lim_{k \to +\infty} \left( \sum_{j=1}^{t} \frac{1}{j} - \ln t k \right) \]

\[ = \lim_{k \to +\infty} \left( \sum_{j=1}^{s} \frac{1}{j} - \ln s \right) - \left( \frac{1}{j} - \ln t k \right) \]

\[ = \lim_{k \to +\infty} \left( \sum_{j=1}^{s} \frac{1}{j} - \ln s \right) - \ln \frac{1}{T} \]

which implies that

\[ \lim C_{s,k,t} = \ln \frac{1}{T}. \]

And we see that asymptotically, \( C_{s} \to A_{s} \), as the number of observed ones \( (s) \) and the number of observed zeros \( (t) \) both approach infinity.

### 3.4 The Minimum Mean Square Bias Estimate Revisited

An obvious alternative method for deriving a minimum mean square bias estimate would be to assume that \( p \sim U(0, 1) \), as was done for the MMSE estimate. In that case, we find that we want to choose the \( B_{n-k} \) to minimize

\[ \int_{0}^{1} \left[ \sum_{k=0}^{n} \left( \sum_{j=1}^{n} k \cdot p^{k} \right) \right] \cdot \left[ \ln \left( \frac{p}{1-p} \right) \right]_{k=0}^{n-k} \quad \text{(15)} \]

Expanding and computing the partial with respect to \( B_{n-k} \) gives the \( k \)th element of the gradient

\[ \frac{\partial}{\partial B_{n-k}} = \]

\[ 2 \left( \sum_{k=0}^{n} \left[ \sum_{j=0}^{n} B_{i,j} \cdot \left( \ln p \right) \right] \cdot \left( 1-p \right)^{n-k} \right) \cdot \left( \ln \left( \frac{p}{1-p} \right) \right) \cdot dp \]

Evaluating the integrals (as in the previous section) and setting the gradient to zero yields the following system of linear equations for the estimates:

\[ \sum_{k=0}^{n} \left( \sum_{j=0}^{n} \left( (k+j)!/(2n-k-j)! \right) \right) \cdot B_{i,j} = \]

\[ k!(n-k)! \cdot \left[ \ln p - \ln (1-p) \right] \]

for \( k = 0, 1, 2, ..., n \). Note that if we let \( A \) be a matrix with entries

\[ A_{i,j} = \frac{n!(n+1)!}{(2n+1)!} \cdot (i+j)!/(n-i-j)! \cdot (2n-i-j)!/(n+1)! \cdot \ln \left( \frac{p}{1-p} \right) \]

then we can write

\[ \begin{bmatrix} B_{0,0} \\ B_{0,1} \\ \vdots \\ B_{n,0} \end{bmatrix} = \begin{bmatrix} C_{0,0} \\ C_{1,0} \\ \vdots \\ C_{n,0} \end{bmatrix} \approx \begin{bmatrix} 20 & 10 & 4 & 1 \\ 10 & 12 & 9 & 4 \\ 4 & 12 & 10 & 3 \\ 1 & 4 & 10 & 20 \end{bmatrix} \]

Solving this system yields the following estimates:

\[ \begin{bmatrix} B_{0,3} \\ B_{1,2} \\ B_{2,1} \\ B_{3,0} \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 6 & 3 \\ 11 \end{bmatrix} \]

which are not what we expect, since we get some positive estimates where we expect negative ones, and vice-versa. This behavior happens with all estimates for three or more trials, and we see that this estimator is inconsistent.

### 4 Solving the Least Squares Problem

Once we have obtained estimates for the local interaction sums, then we can set up and solve (2) in order to obtain our final estimate of the MRF parameters. In this section, we will examine this process in more detail and show some different approaches. Note that although we describe these methods in terms of the binary autobinomial MRF with a second-order neighborhood structure, they can be applied much more generally.

We begin by recalling that the local interaction vector for the \( i, j \) pixel is

\[ v = \begin{bmatrix} y_{i,j-1} + y_{i,j+1} \\ y_{i-1,j} + y_{i+1,j} \\ y_{i-1,j-1} + y_{i+1,j+1} \\ y_{i+1,j-1} + y_{i-1,j+1} \end{bmatrix} \]

Note that the first element is always one, and each of the last four elements may be either zero, one, or two, that is, \( v = [1, v_{2}, v_{3}, v_{4}] \), where \( v_{i} \in \{0, 1, 2\} \) for \( i = 2, 3, 4, 5 \). Given any one of the 81 possible vectors, we can use the following formula to determine a unique associated index:

\[ i = 1 + v_{2}3^{2} + v_{3}3^{3} + v_{4}3^{4} + v_{5}3^{5} \]

Similarly, it is possible to go from any integer \( i = 1, 2, ..., 81 \) to its corresponding local interaction vector by applying Horner’s rule to extract the digits in the base three expansion of \( i \), which become the elements of \( v \). Of course, many other orderings are possible but we will find this one convenient.

#### 4.1 Rapid Solution of the Full System

One of the most serious deficiencies of the \( A \) and \( B \) estimates is that they are not always defined. As a result, there are usually some indices \( i \) for which \( v \) cannot be estimated and hence the corresponding row must be deleted from (2) before solving the least-squares problem; we shall call this
process equation culling. In contrast, the $C$ estimate always exists. This means that it is possible to reach the second stage and solve (2) without removing any equations.

If we can avoid culling equations, then, for a given model, only the right-hand side of the least-squares problem will change for each different realization of the random field. This implies that we can simplify the solution process, since we can precompute the pseudo-inverse, $(V^TV)^{-1}V^T$, of the full $V$ and then form our estimate with a simple matrix multiply. We note that, in general, it is numerically unwise to compute the pseudo-inverse or use it as a method of solving the least-squares problem. However, we give it analytically here and will show that it leads to a very elegant algorithm because of its special structure.

It is tedious, but not difficult to show that the pseudo-inverse of $V$ can be written explicitly in the following form:

$$(V^TV)^{-1}V^T = \begin{bmatrix}
1/81 & 1/54 & 1/54 & 1/54 & 1/54 \\
0 & -1/54 & 0 & 0 & 0 \\
0 & 0 & 0 & -1/54 & 0 \\
0 & 0 & 0 & 0 & -1/54 \\
S,
\end{bmatrix}$$

where $S \in \mathbb{R}^{500}$ is a matrix whose entries are described in the following way. The first $3^{\epsilon_i}$ entries in the $i$th row are ones, followed by $3^{\epsilon_i}$ zeros, followed by $3^{\epsilon_i}$ minus ones. This pattern repeats until the row is filled. In particular, the first row is all ones, the second is $27$ ones, followed by $27$ zeros, followed by $27$ minus ones, and so on.

We note that the easiest way to derive this fact analytically is to compute the QR-factorization of $V$ via the Gram-Schmidt process. Once we have $QR = V$, then the pseudo-inverse is given by $R^{-1}Q^T$. For the case under consideration, it is then easily observed that $Q^T$ is sparse and can be diagonally scaled to have entries that are either zero or ±1. Assimilation of this diagonal scaling into $R^{-1}$ unnormlizes $Q$ and leads to the matrix $S$ just described. Given this simple form, we see that the estimates for the Markov parameters are easily calculated with the following formulas:

$$
\beta_4 = -\frac{1}{54} \sum_{i=0}^{27} (v_{3i+1} - v_{3i+3}) \\
\beta_5 = -\frac{1}{54} \sum_{i=0}^{27} \sum_{j=9i+1}^{9i+9} (v_j - v_{j+6}) \\
\beta_2 = -\frac{1}{54} \sum_{i=0}^{27} \sum_{j=27i+1}^{27i+9} (v_j - v_{j+18}) \\
\beta_1 = -\frac{1}{54} \sum_{j=1}^{81} (v_j - v_{j+54}) \\
\beta_0 = \frac{1}{81} \sum_{i=1}^{81} v_i - \beta_1 - \beta_2 - \beta_3 - \beta_4
$$

It is very noteworthy that these formulas yield the final estimates for the Markov parameters with only $4 \times 53 + 80 = 292$ adds and five divides. If intermediate sums are stored, then one can compute the estimates with only 160 adds and five divides. This is a very substantial savings over the $5 \times 81 = 405$ multiplies and $5 \times 80 = 400$ adds that are required to multiply a general $5 \times 81$ matrix against a column vector.

Although it appears that this rapid algorithm for computing the least-squares solution is very specialized to the specific MRF under consideration, it is in fact possible to derive similar forms for other MRF models.

### 4.2 Using Generalized Least Squares

Because we are not given the true values of the local interaction sums, $v$, we must use estimates, $\hat{v}$. Assume that the true value and the estimate are related by $v = \hat{v} + n$, where $n$ is the error of the estimate. Then (2) can be rewritten as

$$V\hat{\beta} = \hat{v} + n,$$

where we assume that the vector $n = [n_1, n_2, \ldots, n_n]^T$ is a noise vector with zero mean and known covariance $W$. A common approach is to solve the generalized least-squares problem

$$\min\|B^{-1}(V\hat{\beta} - \hat{v})\|_2^2,$$

where $B$ is the Cholesky factor of $W$. For our purposes, we will assume that the $n$ are uncorrelated so that $W$ will be diagonal. This leads to a weighted least-squares problem [5] where one can vary the contribution that each equation makes toward the final solution. A similar approach was used very successfully in [6], but their algorithm relies on heuristic methods for determining the weights. In contrast, we will assume that the variance of $n$ is equal to the mean squared error of the estimate $\hat{v}$, which is given by (13). Specific algorithms for solving the generalized least-squares problem can be found in [7] and references therein.

### 5 Numerical Experiments

In this section, we present the results of some simple numerical experiments. We note that all experiments were carried out using synthetic MRF textures that were generated with a fast Gibbs sampler algorithm. For each MRF texture, its corresponding histogram is generated over a single independent coding of the lattice. In particular, all textures are $N \times N$ with $N$ an even integer, and the histogram is generated by looking only at pixels with odd indices. This is in keeping with the restrictions on conditional independence, although it should be noted that if one bases the histogram on the entire texture, then the observed outcomes of the various experiments are substantially the same.

Of course, since we are considering algorithms for the estimation of the texture parameters of an MRF, we will present statistics on these in the tables that summarize each experiment. However, it is important to note that these numbers can be somewhat misleading. What really drives the process of texture formation is the conditional probabilities, and these are related to the texture parameters by an invertible but complex nonlinear transformation. This makes it difficult to establish a meaningful metric in the
parameter space, since it is possible to have large variations in a texture parameter that lead to only small variations in the conditional probabilities. Therefore, it is also important to try to measure how well the texture parameters generated by the different algorithms reproduce the correct conditional probabilities.

Given a set of texture parameters $b$ and an estimated set $\hat{b}$, we will consider the following measure:

$$\zeta = \sum_{i=1}^{81} \frac{e^{V^T \hat{b}}}{1 + e^{V^T \hat{b}}} - \frac{e^{V^T b}}{1 + e^{V^T b}}^2,$$

which is the squared distance from the true conditional probabilities to those given by the estimated texture parameters.

In the first set of experiments, we compare three estimation procedures. We note that for all three, we discard any equations for which fewer than five pixels with the corresponding index were observed in the histogram. The three procedures are:

1) The algorithm proposed by Derin and Elliot [1]. This uses the $A$ estimates, and hence we must also discard all equations for which the estimated marginals are zero or one.

2) The algorithm proposed by Gurelli and Onural [4]. This uses the $B$ estimates for any index with fewer than 12 observations, and the $A$ estimates otherwise. All equations with 12 or more observations for which the estimated marginals are zero or one must be discarded.

3) In this algorithm, we use the $C$ estimates, so that no additional equations need to be discarded.

The results of this experiment appear in Table 1. Because the three methods discard different amounts of information, we also keep track of $N$, the number of admissible equations that appears in the least squares problem. It is interesting to see just how few equations are used, on average, to estimate the parameters. In fact, for the $30 \times 30$ regions, we find that Method I averages only eight equations and in fact fails roughly 3% of the time because of rank deficiency.

In the second experiment, we look at using the $C$ estimates and then solving the full unculled least squares problem using the fast algorithm. These results appear in Table 2. In this experiment, we note that although the texture parameter estimates are not as close as in the first experiment, they do exhibit smaller standard deviation.

Based on 1,000 trials for each region size. Note that Method I fails 33 times with the $30 \times 30$ regions, so that the corresponding statistics in that case are based on only 967 trials. For this experiment, the textures were generated using parameters $b_1 = 4$ and $b_2 = b_3 = b_4 = 1$.
Moreover, the $\zeta$ error measure appears to be uniformly better, which is our first indication that there may be useful information in the fact that certain indexes are not observed.

The third experiment again uses the $C$ estimates but then considers solution via weighted least squares. For practical purposes, the mean square error values for $n = 0, 1, 2, ..., 999$ were separately computed and stored in a look-up table, for $n > 999$, the rational approximation from (14) is used. We consider two variations of this approach: one using the full (or unculled) set of equations and the other using only those equations for which there are observations. These results appear in Table 3. We note that both approaches are comparable: the first having smaller $\zeta$ errors and the second giving slightly better average estimates of the texture parameters. Again it appears that equation culling leads to larger $\zeta$ errors but lower bias in the texture parameter estimates. An excellent analysis of the use of weighted least-squares methods for estimating texture parameters can be found in [6]. They also present results indicating that weighted least squares compares quite favorably with maximum-likelihood methods.

6 Conclusions

We have examined the histogram method for estimating the parameters of a Markov random field. It is seen that two of the major steps in implementing this algorithm are to estimate the local interaction sums and to solve a least squares problem. We derived a new estimator for the local interaction sums that was shown to be optimal in the minimum mean squared error sense. Moreover, we saw that this estimator has the proper asymptotic behavior as the number of observations gets large. Finally, unlike the other estimates in common use, this estimate is defined for all possible collections of observations.
Second, we showed that the least-squares problem can be solved extremely rapidly if no equation culling was required after estimation of the local interaction sums. We also showed how to apply generalized least squares methods to this problem. It was seen that the generalized least squares approach gave excellent estimates even if the size of the sample realization was small.

ACKNOWLEDGMENTS

The author wishes to thank Alessandro Beghi for his insight and numerous patient explanations. The author also gratefully acknowledges partial support from TENCAP Command, U.S. Navy.

REFERENCES

[1] H. Derin and H. Elliot, “Modelling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, 1987.

[2] J. Besag, “Spatial Interaction and the Statistical Analysis of Lattice Systems,” *J. Royal Statistical Soc., Series B*, vol. 2, 1974.

[3] G. Cross and A. Jain, “Markov Random Field Texture Models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, 1983.

[4] M. Gurelli and L. Onural, “On a Parameter Estimation Method for Gibbs-Markov Random Fields,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, Apr. 1994.

[5] C. Gauss, “Theoria Combinationis Observationum Erroribus Minimis Obnoxiae,” Classics in Applied Mathematics, SIAM, 1995.

[6] B. Calder, L. Linnett, S. Clarke, and D. Carmichael, “Improvements in Markov Random Field Parameter Estimation,” *IEEE E4 Colloquium on Multiresolution Image Processing*. London: IEE, Apr. 1995, no. 1995/077.

[7] G. Golub and C.V. Loan, *Matrix Computations*, third ed. Baltimore: Johns Hopkins Univ. Press, 1996.

Carlos F. Borges received the BS degree in computer science in 1984, the MS degree in applied mathematics in 1986, and the PhD degree in applied mathematics in 1990, all from the University of California, Davis. From 1986 to 1990, he was a member of the technical staff at GTE Government Systems, Western Division. In 1991, he joined the Department of Mathematics at the Naval Postgraduate School, where he is now an associate professor. His research interests include numerical computation, signal and image processing, and applied stochastics.