Journal of Statistical Software

October 2007, Volume 21, Issue 3. *http://www.jstatsoft.org/*

elrm: Software Implementing Exact-like Inference for Logistic Regression Models

David Zamar
Simon Fraser University

Brad McNeney
Simon Fraser University

Jinko Graham
Simon Fraser University

Abstract

Exact inference is based on the conditional distribution of the sufficient statistics for the parameters of interest given the observed values for the remaining sufficient statistics. Exact inference for logistic regression can be problematic when data sets are large and the support of the conditional distribution cannot be represented in memory. Additionally, these methods are not widely implemented except in commercial software packages such as **LogXact** and SAS. Therefore, we have developed **elrm**, software for R implementing (approximate) exact inference for binomial regression models from large data sets. We provide a description of the underlying statistical methods and illustrate the use of **elrm** with examples. We also evaluate **elrm** by comparing results with those obtained using other methods.

Keywords: conditional inference, exact test, logistic regression, Markov chain Monte Carlo, Metropolis-Hastings algorithm.

1. Introduction

Statistical inference for logistic regression models typically involves large sample approximations based on the unconditional likelihood. Unfortunately, these asymptotic approximations are unreliable when sample sizes are small or the data are sparse or skewed. In these situations, exact inference is reliable no matter how small or imbalanced the data set. Exact inference is based on the conditional distribution of the sufficient statistics for the parameters of interest given the observed values for the remaining sufficient statistics. As the sample size grows and the data become better balanced and less sparse, conventional large sample inference will coincide with exact inference. Exact logistic regression refers to exact conditional inference for binomial data modelled by a logistic regression. Current implementations of exact logistic regression have difficulty handling large data sets with conditional distributions

whose support is too large to be represented in memory. We extend an existing algorithm for (approximate) exact inference to accommodate large data sets and implement this extension in an R (R Development Core Team 2007) package called **elrm**. We begin this paper with a short review of exact logistic regression in Section 2. In Section 3, we discuss related work and our extension. Section 4 describes the inference provided by **elrm**, our implementation of this extension. In Section 5 we illustrate the usage of **elrm** and its features. In Section 7, we evaluate our package and present the results. Section 8 provides a summary of our work.

2. Exact logistic regression

Hirji (2006) provides a useful introduction to exact inference and to approximate exact inference. In this article, we focus on approximate exact inference for logistic regression models.

In logistic regression, the outcome of interest is modeled as a binomial random variable. Let Y_i be the i^{th} binomial response with m_i trials and success probability p_i. The logistic regression model is

$$logit\left(p_i\right) = \mathbf{w}_i^\top \beta + \mathbf{z}_i^\top \gamma, \quad i = 1, \ldots, n,$$

where β is a vector of nuisance parameters corresponding to the first p explanatory variables $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{ip})^\top$ for the i^{th} response, γ is a vector of parameters corresponding to the remaining q explanatory variables $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iq})^\top$ and n is the number of responses. We are not interested in making inferences about β; however, including the $\mathbf{w}_i$'s in the model reduces noise and provides better inference about the regression parameters, γ, of interest. Ultimately, we are interested in studying the relationship between p_i and $\mathbf{z}_i$.

Let $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$, $\mathbf{W}$ be an $n \times p$ matrix whose ith row is $\mathbf{w}_i^\top$ and $\mathbf{Z}$ be an $n \times q$ matrix whose ith row is $\mathbf{z}_i^\top$. Exact conditional inference is based on the distribution of the sufficient statistic $\mathbf{T} = \mathbf{Z}^\top \mathbf{Y}$ for the parameters of interest, γ, given the sufficient statistic $\mathbf{S} = \mathbf{W}^\top \mathbf{Y}$ for the nuisance parameters β. Equivalently, inference is based on the conditional distribution of $\mathbf{Y}$ given $\mathbf{S}$,

$$f\left(\mathbf{y}|\mathbf{S} = \mathbf{s}\right) \propto \left[\prod_{i=1}^{n} \binom{m_i}{y_i}\right] \exp\left\{\gamma^\top \mathbf{Z}^\top \mathbf{y}\right\}. \tag{1}$$

This distribution does not depend on β since we are conditioning on its sufficient statistic $\mathbf{S}$.

To make exact conditional inference about γ, we need to be able to evaluate the distribution $f\left(\mathbf{y}|\mathbf{S} = \mathbf{s}\right)$. Approximate exact inference is based on an estimate of $f\left(\mathbf{y}|\mathbf{S} = \mathbf{s}\right)$ that is obtained by sampling from the distribution. However, computation of the proportionality constant in equation (1) can be problematic for large data sets, because it requires enumeration of the potentially large support of $f\left(\mathbf{y}|\mathbf{S} = \mathbf{s}\right)$. Fortunately, Markov chain Monte Carlo (MCMC) approaches require knowledge of $f\left(\mathbf{y}|\mathbf{S} = \mathbf{s}\right)$ up to a proportionality constant only.

3. Related work and extensions

3.1. Currently available methods

Oster (2002) and Oster (2003) review and compare exact methods implemented in various software packages. For logistic regression, exact inference is based on the conditional distribution of the sufficient statistics for the regression parameters of interest given the sufficient

statistics for the remaining nuisance parameters. A recursive algorithm for generating the required conditional distribution is implemented in the commercial software package **LogXact** (Cytel Inc. 2006a). However, the algorithm can only handle problems with modest samples sizes and numbers of covariates (Corcoran *et al.* 2001). To increase the size of problem that can be analyzed, Mehta *et al.* (2000) developed a Monte Carlo method for (approximate) exact inference and implemented it in **LogXact**. Their method represents the support of the conditional distribution by a network of arcs and nodes. The limiting factor for their approach is the size of the network, which must be stored in memory. Forster *et al.* (1996) circumvented the need to represent the support by developing a Gibbs sampler to generate dependent Monte Carlo samples. One potential drawback of a Gibbs sampling approach is that it would sample from the conditional distribution of a particular sufficient statistic given the observed values of the sufficient statistics for the nuisance parameters *and* the current values of the sufficient statistics for the remaining parameters of interest. In exact conditional inference for logistic regression, conditioning on too many sufficient statistics can greatly restrict the set of support points for the conditional distribution, making the distribution highly discrete or even degenerate. This "overconditioning" problem is particularly acute when conditioning on sufficient statistics associated with continuous covariates in the logistic regression model. In the context of Gibb's sampling, such overconditioning can lead to poor mixing or even a degenerate conditional distribution for the complete vector of sufficient statistics of interest. For large problems, in which storage of the network is not possible and the Gibbs sampler proves unreliable, Forster *et al.* (2003) propose an alternative method that makes use of the Metropolis-Hastings algorithm.

3.2. The Forster et al. (2003) algorithm

The Metropolis-Hastings algorithm proposed by Forster *et al.* (2003) generates proposals for the binomial response vector that differ only in a few entries from the current state of the Markov chain, such that the values of the sufficient statistics for the nuisance parameters remain the same. Specifically, the algorithm involves generating proposals $\mathbf{y}^*$ of the form $\mathbf{y}^* = \mathbf{y} + d \cdot \mathbf{v}$, where, for a given integer r, the perturbation $\mathbf{v}$ is a vector from

$$\mathbf{V} = \left\{ \mathbf{v} : \sum_{i=1}^{n} |v_i| \leq r \text{ and } v_i \text{ coprime for } i = 1, ..., n \text{ and } \mathbf{W}^\top \mathbf{v} = \mathbf{0} \right\}$$

and d is an integer such that $0 \leq y_i + dv_i \leq m_i$ for $i = 1, ..., n$. Initially, the set

$$\mathbf{V}' = \left\{ \mathbf{v} : \sum_{i=1}^{n} |v_i| \leq r \text{ and } v_i \text{ coprime for } i = 1, \ldots, n \right\}$$

is enumerated for a given r chosen so that enumeration is feasible. Only those $\mathbf{v}$ for which $\mathbf{W}^\top \mathbf{v} = 0$ are kept. Usually, the vector of ones is in the column space of the design matrix $\mathbf{W}$ because a constant term is included in the linear model. Hence $\sum_{i=1}^{n} v_i = 0$ and so $\sum_{i=1}^{n} |v_i|$ and therefore r must be even. The Metropolis-Hastings algorithm involves first selecting one of the possible $\mathbf{v} \in \mathbf{V}$ with equal probability and then generating d using

$$q\left(d|\mathbf{v}, \mathbf{y}\right) \propto \exp\left\{\gamma^\top \mathbf{Z}^\top \left(\mathbf{y} + d\mathbf{v}\right)\right\} \prod_{i=1}^{n} \binom{m_i}{y_i + dv_i},$$

where the support of $q(d|\mathbf{v}, \mathbf{y})$ is given by

$$0 \leq y_i + dv_i \leq m_i \tag{2}$$

for all i. Since $\mathbf{y}^* = \mathbf{y} + d\mathbf{v}$, where $\mathbf{W}^\top \mathbf{v} = \mathbf{0}$, the sufficient statistics $\mathbf{W}^\top \mathbf{y}^*$ for the nuisance parameters are maintained. In order to obtain the required stationary distribution, the algorithm accepts the proposal $\mathbf{y}^*$ with probability 1. The selected value of r controls the mixing of the Markov chain. Large values allow for larger transitions in the chain and better mixing. On the other hand, since the size of the initial set $\mathbf{V}'$ increases with r, smaller values of r ensure its enumeration is feasible. Additionally, r affects the second-stage sampling of d conditional on the realized value of $\mathbf{v}$ and $\mathbf{y}$. In particular, large values of r will increase the chance that $d = 0$ is the only integer satisfying the constraints (2). If $d = 0$ with high probability the Markov chain will mix poorly as this represents a "transition" to the current state. Forster *et al.* (2003) suggest choosing r to be 4, 6, or 8. Small values of r correspond to more transitions to new states, but the Markov chain may remain trapped in a local neighborhood (Forster *et al.* 2003; Zamar 2006).

3.3. The elrm algorithm

The Forster *et al.* (2003) algorithm proposes uniform sampling of perturbation vectors from the set $\mathbf{V}$ after enumerating $\mathbf{V}$ and storing it in memory. However, the size of the initial set $\mathbf{V}'$ that is used to construct $\mathbf{V}$ grows rapidly with the length of the response vector. Thus, for large data sets, the required enumeration of $\mathbf{V}'$ can be impractical. Additionally, $\mathbf{V}$ may be too large to store in memory. To accommodate large data sets, we implement an extension of this algorithm with two important differences:

1. We sample from a subset $\mathbf{V_A}$ of $\mathbf{V}$ whose vectors satisfy the additional constraint that $|v_i| \leq m_i$ for all $1 \leq i \leq n$.

2. We sample vectors uniformly from $\mathbf{V_A}$ without enumerating a larger set $\mathbf{V}'$ or storing $\mathbf{V_A}$.

Sampling from $\mathbf{V_A}$ instead of $\mathbf{V}$ improves mixing because vectors for which some $|v_i| > m_i$ will only satisfy constraint (2) if $d = 0$, so that $\mathbf{y}^* = \mathbf{y}$ with probability one. For details on how uniform samples from $\mathbf{V_A}$ are obtained, readers are referred to Zamar (2006).

4. Inference provided by elrm

Let $S_1, ..., S_p$ denote the sufficient statistics for the nuisance parameters $\beta_1, ..., \beta_p$ in the logistic regression model. Likewise let $T_1, ..., T_q$ denote the sufficient statistics for $\gamma_1, ..., \gamma_q$, the parameters of interest. In this section, we describe the methods used by **elrm** to conduct hypothesis tests and produce point and interval estimates for the parameters of interest.

4.1. Hypothesis tests

To conduct joint inference on $\gamma_1, ..., \gamma_q$, **elrm** first produces a sample of dependent observations from the joint distribution

$$f_{T_1,...,T_q}(t_1, ..., t_q \mid S_1 = s_1, ..., S_p = s_p,\ \gamma_1 = \cdots = \gamma_q = 0)\,. \tag{3}$$

In order to test

$$H_0 : \gamma_1 = \cdots = \gamma_q = 0$$

against the two-sided alternative

$$H_1 : \exists\ \gamma_i \neq 0, \quad i = 1, ..., q$$

we compute an approximate two-sided p value for the conditional probabilities test (e.g. Mehta and Patel 1995). The two-sided p value for the conditional probabilities test is obtained by summing estimates of the probabilities (3) over the critical region

$$E_{cp} = \left\{ \mathbf{u} : \hat{f}_{\mathbf{T}} \left(\mathbf{u} \mid \mathbf{S} = \mathbf{s},\ \gamma = \mathbf{0} \right) \leq \hat{f}_{\mathbf{T}} \left(\mathbf{t} \mid \mathbf{S} = \mathbf{s},\ \gamma = \mathbf{0} \right) \right\},$$

where $\mathbf{t}$ is the observed value of the sufficient statistics for $\gamma_1, ..., \gamma_q$ and $\hat{f}_{\mathbf{T}}$ is an estimate of (3). The Monte Carlo standard error of the resulting p value estimate is computed by the batch-means method (e.g. Geyer 1992).

To conduct separate inference on each γ_i, we consider $\gamma_1, ..., \gamma_{i-1}, \gamma_{i+1}, ..., \gamma_q$ together with $\beta_1, ..., \beta_p$ as nuisance parameters. Generalizing the distribution (3), inference is based on a sample of dependent observation from

$$f_{T_i} \left(u \mid \mathbf{S} = \mathbf{s},\ \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = 0 \right), \tag{4}$$

where $\mathbf{T}_{-i}$ and $\mathbf{t}_{-i}$ are, respectively, the vector of sufficient statistics for the parameters of interest and its observed value for all but the i^{th} element. The required sample may be extracted from the original Markov chain generated for the joint hypothesis test. Consequently, this extracted sample may be much smaller than the length of the original chain, especially if the joint hypothesis test involves several parameters. If accurate inference for a particular γ_i is required, it may be best to re-run **elrm** with γ_i as the only parameter of interest. That said, we may still attempt to use the existing chain to test

$$H_0 : \gamma_i = 0$$

against the two-sided alternative

$$H_1 : \gamma_i \neq 0.$$

We compute an approximate two-sided p value by summing estimates of the probabilities (4) over the critical region

$$E_{cp} = \left\{ u : \hat{f}_{T_i} \left(u \mid \mathbf{S} = \mathbf{s},\ \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = 0 \right) \leq \hat{f}_{T_i} \left(t_i \mid \mathbf{S} = \mathbf{s},\ \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = 0 \right) \right\}.$$

The Monte Carlo standard error of each resulting p value is again computed by the batch-means method.

4.2. Point and interval estimates

The **elrm** package returns a point estimate and confidence interval for each γ_i of interest. Where possible, the estimate of each γ_i is obtained by maximizing the conditional likelihood function in (4) with respect to γ_i. Estimation of the conditional distribution of T_i under

different values $\acute{\gamma}_i$ of the parameter of interest is conveniently achieved by re-weighting the sample frequencies under $\gamma_i = 0$ as

$$\hat{f}\left(T_i = t \mid \mathbf{S} = \mathbf{s}, \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = \acute{\gamma}_i\right) = \frac{\hat{f}\left(T_i = t \mid \mathbf{S} = \mathbf{s}, \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = 0\right) \exp\left\{\acute{\gamma}_i\, t\right\}}{\sum_u \hat{f}\left(T_i = u \mid \mathbf{S} = \mathbf{s},\ \mathbf{T}_{-i} = \mathbf{t}_{-i},\ \gamma_i = 0\right) \exp\left\{\acute{\gamma}_i\, u\right\}}.$$

Sometimes maximization of the conditional likelihood is not possible, because the observed value, t_i, of the sufficient statistic for γ_i lies at one extreme of its range. In this case the median unbiased point estimate (MUE) is reported (Mehta and Patel 1995).

We obtain a level-α confidence interval, (γ_-, γ_+) for γ_i, by inverting two one-sided likelihood ratio tests for γ_i. More precisely, following Mehta and Patel (1995), we define

$$F_1\left(t_i|\gamma_i\right) = \sum_{v \geq t_i} f_{T_i}\left(v|\gamma_i\right)$$

and

$$F_2\left(t_i|\gamma_i\right) = \sum_{v \leq t_i} f_{T_i}\left(v|\gamma_i\right),$$

where $f_{T_i}\left(v|\gamma_i\right)$ is given by (4) with the conditioning arguments omitted for brevity in the current notation. Let t_{min} and t_{max} be the smallest and largest possible values of t_i in the distribution (4). The lower confidence bound γ_-, is such that

$$\begin{aligned} F_1\left(t_i|\gamma_-\right) &= \alpha/2 \quad \text{if} \quad t_{min} < t_i \leq t_{max} \\ \gamma_- &= -\infty \quad \text{if} \quad t_i = t_{min}. \end{aligned}$$

Similarly,

$$\begin{aligned} F_2\left(t_i|\gamma_+\right) &= \alpha/2 \quad \text{if} \quad t_{min} \leq t_i < t_{max} \\ \gamma_+ &= \infty \quad \text{if} \quad t_i = t_{max}. \end{aligned}$$

5. Using elrm and its features

Our contributed R package, **elrm**, is available for download from the Comprehensive R Archive Network (CRAN) website at `http://CRAN.R-project.org/`.

The main function of the **elrm** package is `elrm()`, which returns an object of class "`elrm`" for which `summary`, `plot` and `update` methods are available. A call to `elrm()` will both generate the Markov chain of sampled sufficient statistics for the parameters of interest in the logistic regression model (conditional on the observed values of the sufficient statistics for the nuisance parameters) and conduct inference. The generated chain, saved as an "`mcmc`" object from the **coda** package (Plummer *et al.* 2006), is stored along with inference results in the "`elrm`" object that is returned. The user specifies the logistic regression model and regression parameters of interest by passing:

1. `formula`: a symbolic description in R formula format of the logistic regression model (including nuisance parameters and parameters of interest). One exception is that the binomial response should be specified as success/trials, where success gives the number of successes and trials gives the number of binomial trials for each row of the dataset.

2. `interest`: a symbolic description in R formula format of the model terms for which exact conditional inference is of interest.

3. `dataset`: a "`data.frame`" object where the data are stored. The "`data.frame`" object must include a column specifying the number of successes for each row and a column specifying the number of binomial trials for each row.

For a list of the four other arguments to `elrm()` and their default values, see the help file.

The `summary()` method for **elrm** formats and prints the "`elrm`" object. The summary includes the matched call, coefficient estimates and confidence intervals for each model term of interest, estimated p value for jointly testing that the parameters of interest are equal to zero, full conditional p values from separately testing each parameter equal to zero, length of the Markov chain upon which inference was based, and the Monte Carlo standard error of each reported p value.

The `plot` method can be used as a diagnostic tool to check whether the Markov chain has converged; it produces both a trace plot and histogram of the sampled values of the sufficient statistic for each parameter of interest. Sampled values within the burn-in period are included in the plot. A separate graphics page is used to display the plots corresponding to each parameter of interest. A trace plot displays the sampled value at iteration t against the iteration number t. If the Markov chain has converged, the trace will vary around the mode of the distribution. A clear sign of non-convergence is when a trend is observed in the trace plot. The histogram provides a quick summary of the range and frequency of the sampled values. Sometimes, non-convergence may be reflected by severe multimodality (Gilks *et al.* 1996). In this case, it is important to let the algorithm run longer. The trace plot and histogram are based on a random sample consisting of $\mathtt{p}\times 100\%$ of all the observations in the Markov chain, where the sampling fraction $0 < \mathtt{p} \leq 1$ is specified by the user. The default value of `p` is 1. The observations in the random sample remain in the order in which they were generated by the Markov chain.

The `update()` method is used to extend the Markov chain of an "`elrm`" object by a specified number of iterations. This is done by creating a new Markov chain of the specified length using the last sampled value as the starting point. The newly created chain is then appended to the original and inference is based on the extended Markov chain.

6. Examples

This section illustrates the use of **elrm** with examples.

6.1. Simulated diabetes example

The simulated dataset, `diabDat`, from the **elrm** package will be used for this example and can be loaded into R with the command:

```
R> data("diabDat")
```

These simulated data mimic data from 669 cases in an existing case-control study of type 1 diabetes (Graham *et al.* 1999). In the current investigation, age-specific, gender-adjusted

associations between concentration levels (low and high) of the islet antigen 2 antibody (`IA2A`) and HLA-DQ haplotypes 2, 8 and 6.2 were of interest. Covariates included in the analysis are therefore `age` (rounded to the nearest year), `gender` (coded 0 for females and 1 for males), and the number of copies (0,1 or 2) of the HLA-DQ2, HLA-DQ8 and HLA-DQ6.2 haplotypes (`nDQ2`, `nDQ8` and `nDQ6.2` respectively). The response vector for the simulated data was generated under the following logistic regression model, which includes all main effects and two way interactions involving genetic effects and age:

```
IA2A ~ age + gender + nDQ2 + nDQ8 + nDQ6.2 + age:nDQ2 + age:nDQ8 + age:nDQ6.2
```

The coefficients for `nDQ6.2` and `age:nDQ6.2` were set to zero and the coefficients for the remaining regression parameters were assigned their estimated values based on the original data. The first five rows of the simulated data are shown below.

```
R> head(diabDat)
```

```
  n IA2A gender age nDQ2 nDQ8 nDQ6.2
1 9    4      1  14    1    1      0
2 7    3      0   6    1    1      0
3 1    0      0  20    1    0      0
4 6    2      1  34    0    1      0
5 3    0      0  12    1    0      0
6 3    1      0  34    0    1      0
```

Since the HLA-DQ6.2 haplotype is negatively associated with type 1 diabetes (e.g. Graham *et al.* 1999), few patients have a copy of this haplotype. Large-sample inference may thus be unreliable. In these simulated data, none of the 7 patients who carried the DQ6.2 haplotype were antibody positive.

Exact inference for the joint effect of `nDQ6.2` and `age:nDQ6.2` could not be obtained by available versions of the **LogXact** program. The approximate exact 'Monte Carlo' method in **LogXact** ran out of memory during the network construction phase. The Gibb's sampler 'MCMC' method in **LogXact** produced a degenerate chain. In contrast, **elrm** was able to provide results. The estimated exact p value and its Monte Carlo standard error are based on a Markov chain of length 99,500 (after a burn-in of 500 iterations). Inference was obtained with the following call:

```
R> simDiab.elrm <- elrm(IA2A/n ~ gender + age + nDQ2 + nDQ8 + nDQ6.2 +
age:nDQ2 + age:nDQ8 + age:nDQ6.2, interest = ~age:nDQ6.2 + nDQ6.2,
iter = 100000, burnIn = 500, dataset = diabDat)
```

```
Generating the Markov chain ...
Progress: 100%
Generation of the Markov Chain required 1.0744 hours
Conducting inference ...
Warning messages:
1: 'nDQ6.2' conditional distribution of the sufficient statistic was found to
be degenerate
```

```
2: 'age:nDQ6.2' conditional distribution of the sufficient statistic was found
to be degenerate
Inference required 7 secs
```

Once finished, the `elrm()` method displays any warning messages that may have arisen, and reports the time needed to generate the chain and conduct inference. The warnings above indicate that the estimated full conditional distributions of the sufficient statistics for `nDQ6.2` and `age:nDQ6.2` were degenerate. These two variables are highly correlated and so conditioning on the sufficient statistic for one greatly restricts the possible values of the sufficient statistic for the other. Such degeneracy arises from over-conditioning. Applying the `summary()` method gives the following results:

```
R> summary(simDiab.elrm)

Call:
[[1]]
elrm(formula = IA2A/n ~ gender + age + nDQ2 + nDQ8 + nDQ6.2 +
    age:nDQ2 + age:nDQ8 + age:nDQ6.2, interest = ~age:nDQ6.2 +
    nDQ6.2, iter = 1e+05, dataset = diabDat, burnIn = 500)

Results:
           estimate p-value p-value_se mc_size
joint            NA 0.76555    0.01838   99500
nDQ6.2           NA      NA         NA   10142
age:nDQ6.2       NA      NA         NA   10142

95% Confidence Intervals for Parameters

           lower upper
nDQ6.2        NA    NA
age:nDQ6.2    NA    NA
```

The resulting p value of 0.76555 for the joint effects of `nDQ6.2` and `age:nDQ6.2` is consistent with the model used to generate these simulated data. The Markov chains produced for separately testing `nDQ6.2` and `age:nDQ6.2` are smaller than that produced for the joint test because they are extracted from the chain for the joint test. No confidence intervals are reported for `nDQ6.2` and `age:nDQ6.2` because the estimated full conditional distribution of the sufficient statistic for each parameter is degenerate.

6.2. Urinary tract infection example

The `utiDat` dataset from the **elrm** package can be loaded into R with the command:

```
R> data("utiDat")
```

The data arise from a study of how first-time urinary tract infection (UTI) is related to contraceptive use and were gathered by the Department of Epidemiology at the University of

Michigan (Cytel Inc. 2006b). The contraceptive use of 447 sexually active college women was surveyed. The binary covariates included in the analysis were `age` (coded as 0 for women less than 24 years old and 1 otherwise), `current` (1 = no regular current sex partner), `dia` (1 = diaphragm use), `oc` (1 = oral contraceptive), `pastyr` (1 = no regular partner with relationship < 1yr), `vi` (1 = vaginal intercourse), `vic` (1 = vaginal intercourse with condom), `vicl` (1 = vaginal intercourse with lubricated condom), `vis` (1 = vaginal intercourse with spermicide). The first five rows of the dataset are shown below.

```
R> utiDat[1:5,]
```

```
  uti  n age current dia oc pastyr vi vic vicl vis
1   1 10   0       0   0  0      0  0   0    0   0
2   0  1   0       0   0  0      0  0   0    0   1
3   0  4   0       0   0  0      0  1   0    0   0
4   4  4   0       0   0  0      0  1   1    0   0
5   1  2   0       0   0  0      0  1   1    0   1
```

The investigators were interested in whether diaphragm use increases UTI risk once the other confounding variables are taken into account. Diaphragm use (`dia`) appears to be important because all 7 diaphragm users developed UTI. To obtain exact inference for the effect of diaphragm use, we make the following call:

```
R> uti.elrm <- elrm(formula = uti/n ~ age + current + dia + oc +
pastyr + vi + vic + vicl + vis, interest = ~dia, iter = 50000,
burnIn = 1000, dataset = utiDat)
```

```
Generating the Markov chain ...
Progress: 100%
Generation of the Markov Chain required 4.55 mins
Conducting inference ...
Inference required 3 secs
```

Applying the `summary()` method gives the following results:

```
R> summary(uti.elrm)
```

```
Call:
[[1]]
elrm(formula = uti/n ~ age + current + dia + oc + pastyr + vi + vic + vicl
     + vis, interest = ~dia, iter = 50000, dataset = utiDat, burnIn = 1000)

Results:
    estimate p-value p-value_se mc_size
dia  1.96395 0.03365    0.00571   49000

95% Confidence Intervals for Parameters

          lower upper
dia -0.07632582   Inf
```

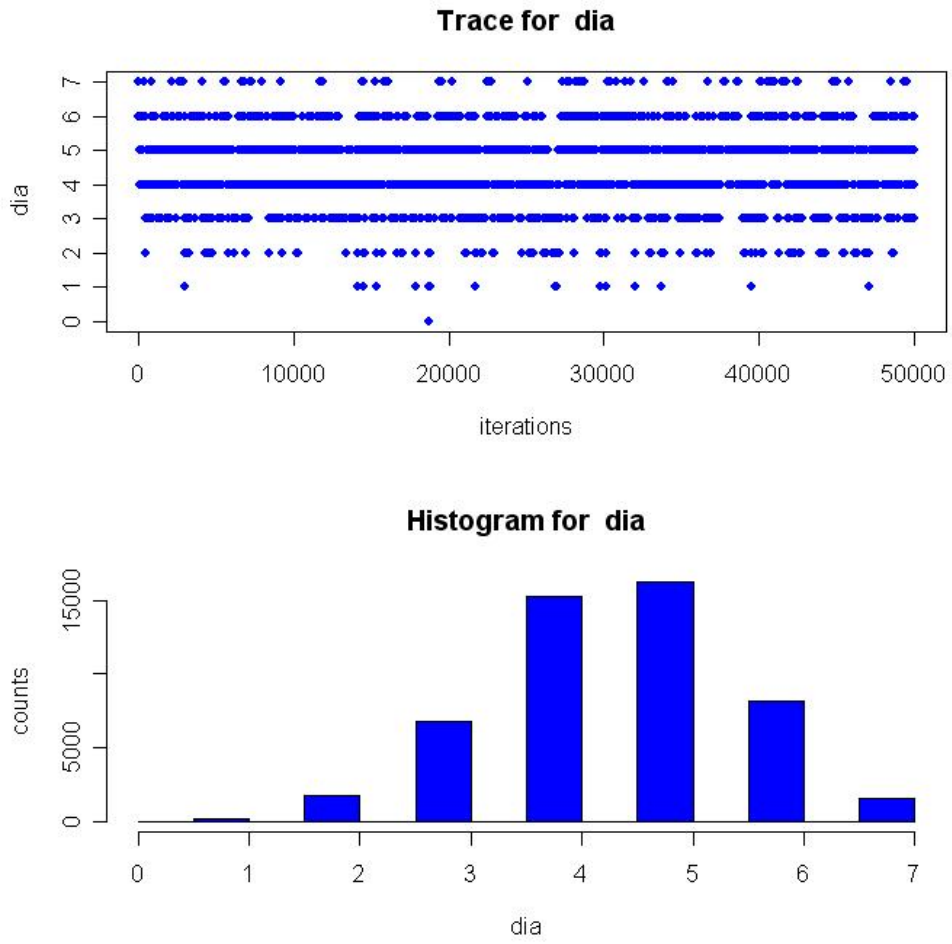

Figure 1: Plot of the Markov chain produced for the `dia` parameter in the UTI example.

The estimated exact p value for the effect of `dia` and its Monte Carlo standard error are based on a Markov chain of length 49,000 (after a burn-in of 1000). Notice that the estimated exact p value is less than 0.05, but the 95% confidence interval for `dia` contains 0. The apparent disagreement arises because the reported p value is based on the conditional probabilities test while the confidence interval is based on the conditional likelihood ratio test. A finite upper bound for the confidence interval could not be obtained because the observed value of the sufficient statistic is the maximum possible value. A trace plot and histogram of values of the sufficient statistic for `dia` sampled by the Markov chain are shown in Figure 6.2. The command used to produce the figure is

```
R> plot(uti.elrm)
```

The estimated conditional distribution of the sufficient statistic for `dia` shown in the histogram is stored in the "`elrm`" object and may be displayed by typing

```
R> uti.elrm$dis
```

```
$dia
     dia         freq
[1,]   0 0.0001428571
[2,]   1 0.0032040816
[3,]   7 0.0303061224
[4,]   2 0.0360000000
[5,]   3 0.1350816327
[6,]   6 0.1638367347
[7,]   4 0.3051836735
[8,]   5 0.3262448980
```

6.3. Hypothetical drug experiment example

The drugDat dataset from the **elrm**, shown below, can be loaded into R with the command:

```
R> data("drugDat")
```

These simulated data are for a hypothetical drug experiment comparing control versus treatment. The response variable, `recovered`, indicates whether or not the patient recovered from a given condition. The covariates of interest are `sex` (1=male, 0=female) and `treatment` (1=treatment, 0=control).

```
R> drugDat
```

```
  sex treatment recovered  n
1   1         1        16 27
2   0         1        10 19
3   1         0        13 32
4   0         0         7 21
```

For a rough assessment, based on only 2000 Markov chain iterations, of whether the effects of `sex` and `treatment` are jointly significant, we could call the `elrm()` method as follows.

```
R> drug.elrm <- elrm(formula = recovered/n ~ sex + treatment,
interest = ~sex + treatment, iter = 2000, dataset = drugDat)
```

```
Generating the Markov chain ...
Progress: 100%
Generation of the Markov Chain required 1 secs
Conducting inference ...
Warning messages:
1: 'sex' extracted sample is too small for inference (less than 1000)
2: 'treatment' extracted sample is too small for inference (less than 1000)
Inference required 0 secs
```

The warnings indicate that full conditional inference for `sex` and `treatment` will be unreliable because the extracted Markov chains are too small. Whenever full conditional inference for a

parameter is based on an extracted Markov chain of length less than 1000, **elrm** will print a warning message and will not return the associated results. Applying the `summary()` method, we obtain:

```
R> summary(drug.elrm)

Call:
[[1]]
elrm(formula = recovered/n ~ sex + treatment,
     interest = ~sex + treatment, iter = 2000,
     dataset = drugDat)

Results:
          estimate p-value p-value_se mc_size
joint           NA   0.097     0.0141    2000
sex             NA      NA         NA      69
treatment       NA      NA         NA     240

95% Confidence Intervals for Parameters

          lower upper
sex          NA    NA
treatment    NA    NA
```

To obtain results for full conditional inference on the separate effects of `sex` and `treatment`, we may try augmenting the Markov chain with a call to `update()`. For example, we could increase the length of the chain by 50,000 iterations (from 2000 to 52,000) and use a burn-in period of 5000:

```
R> drug.elrm <- update(drug.elrm, iter = 50000, burnIn = 5000)

Generating the Markov chain ...
Progress: 100%
Generation of the Markov Chain required 24 secs
Conducting inference ...
Inference required 6 secs
```

Once the `update()` is complete, applying the `summary()` method gives the following results:

```
R> summary(drug.elrm)

Call: [[1]] elrm(formula = recovered/n ~ sex + treatment,
                 interest = ~sex + treatment, iter = 2000,
                 dataset = drugDat)

[[2]] update.elrm(object = drug.elrm, iter = 50000, burnIn = 5000)
```

```
Results:
          estimate p-value p-value_se mc_size
joint           NA 0.15319    0.00290   47000
sex        0.30934 0.54259    0.01499    1397
treatment  0.75603 0.07359    0.00481    6305

95% Confidence Intervals for Parameters

               lower    upper
sex       -0.6048941 1.292658
treatment -0.1285475 1.866684
```

The estimated exact p value for the joint effect of `sex` and `treatment` and its Monte Carlo standard error are based on a Markov chain of length 47,000 (after a burn-in of 5000). Full conditional inferences for `sex` and `treatment` are based on the shorter extracted Markov chains of length 1397 and 6305, respectively.

7. Evaluation

In this section we compare the results obtained by **elrm** and **LogXact** for the urinary tract infection data and the hypothetical drug experiment data.

7.1. Urinary tract infection example

Exact inference for the `dia` parameter could not be obtained by **LogXact** 7 due to memory constraints, while the Gibb's sampler 'MCMC' option produced a degenerate chain. However, the 'Monte Carlo' approximate exact method in **LogXact** 7 was able to conduct the inference. The **LogXact** 7 results were obtained using the default setting (10,000 independent observations) for the Monte Carlo method, which took 10 minutes to complete and required a cumbersome 1042 MB of memory. In contrast, **elrm** took 4.6 minutes to produce a chain of 50,000 dependent observations and required only 75 MB of memory.

Inferences for the `dia` regression parameter obtained by **LogXact** 7 and **elrm** are shown in Table 1, and are similar. However, as shown in Table 2, some differences may be observed in the corresponding conditional distributions estimated by each method. A noticeable difference is that **LogXact** 7 does not sample the value zero, suggesting that the **elrm** Markov chain mixed well.

7.2. Hypothetical drug experiment example

The results obtained by **elrm** for the `drugDat` dataset are summarized in Table 3. Also

	estimate	95% CI	p value	SE of p value
`dia` (**LogXact** 7)	2.0500 MUE	(-0.0726, +INF)	0.0298	0.0033
`dia` (**elrm**)	1.9640 MUE	(-0.0763, +INF)	0.0337	0.0057

Table 1: Inference for the `dia` parameter in the UTI example.

dia	elrm	LogXact 7
0	0.0001	–
1	0.0032	0.0013
2	0.0360	0.0115
3	0.1351	0.0693
4	0.3052	0.2224
5	0.3262	0.4566
6	0.1638	0.2219
7	0.0303	0.0170

Table 2: Empirical conditional distribution of the sufficient statistic for the `dia` parameter in the UTI example.

included in Table 3 are the exact results obtained by **LogXact** 7 and the absolute relative error between the **elrm** and **LogXact** 7 results. The **elrm** results are in close agreement with those produced by **LogXact** 7's exact method.

The percentage errors, obtained by multiplying the relative errors by 100%, are all less than 10 percent, which is quite good given that the Markov chain was moderately small with a length of 52,000 and that full conditional inference for `sex` and `treatment` was based on relatively short Markov chains of length 1397 and 6305, respectively.

	elrm		**LogXact** 7		relative error	
	estimate	p value	estimate	p value	estimate	p value
joint	–	0.1532	–	0.1409	–	0.0872
`sex`	0.3093	0.5426	0.2862	0.5371	0.0809	0.0102
`treatment`	0.7560	0.0736	0.7559	0.0720	0.0002	0.0221

Table 3: Comparison of **elrm** and **LogXact** 7 results for the hypothetical drug experiment data set.

8. Summary

Exact conditional inference is based on the distribution of the sufficient statistics for the parameters of interest given the observed values of the sufficient statistics for the remaining nuisance parameters. When data are sparse and asymptotic approximations based on the unconditional likelihood are unreliable, exact inference can still be made. We consider exact conditional inference for logistic regression models. Commercial software packages such as **LogXact** (Cytel Inc. 2006a) and SAS (SAS Institute Inc. 2003) require large amounts of computer memory to make such inference from large data sets. As pointed out by a reviewer, during the review of this manuscript, the commercial software package Stata 10 (StataCorp. 2007) was released with a new command `exlogistic` that performs exact inference for logistic regression models faster than **LogXact**. However, `exlogistic` was unable to make inference for the larger urinary tract infection (UTI) and diabetes data sets used in our examples. (For the smaller data set from the hypothetical drug experiment, however, `exlogistic` gave sim-

ilar results to the corresponding procedure in **LogXact**.) To allow exact-like inference from larger data sets, such as the UTI and diabetes data sets, we have developed **elrm**, an R package for conducting approximate exact inference in logistic regression models. The Markov chain Monte Carlo algorithm implemented in **elrm** extends the algorithm proposed by Forster *et al.* (2003) to enable its application to large data sets. The extensions we make relax the potential enumeration and memory constraints of their algorithm and should enhance mixing of the chain.

Users of R should find **elrm** easy to work with. The logistic model and parameters of interest are specified using R formula notation similar to that of `glm`. Three input arguments upon which the **elrm** algorithm depends are the number of iterations of the Markov chain (default `iter = 1000`), the burn-in period (default `burnIn = 0`) and the value of the Markov chain mixing parameter (default `r = 4`). Large values of the mixing parameter `r` correspond to larger, less frequent transitions in the Markov chain, while smaller values of `r` correspond to smaller, more frequent transitions in the chain. Typical values of `r` recommended by Forster *et al.* (2003) are 4, 6 or 8. Inference provided by **elrm** includes an approximate exact p value for jointly testing that the parameters of interest are equal to zero, an approximate exact p value for separately testing each parameter of interest is equal to zero, the Monte Carlo standard error of each reported p value, and point and interval estimates of the coefficients of interest in the logistic regression model.

Acknowledgments

We would like to thank Åke Lernmark, the Swedish Childhood Diabetes Study Group and the Diabetes Incidence in Sweden Study Group for providing access to the diabetes data that motivated this work. Thanks also to a referee for helpful comments. This research was supported by Natural Sciences and Engineering Research Council of Canada grants 227972-00 and 213124-03, by Canadian Institutes of Health Research grants NPG-64869 and ATF-66667, and in part by the Mathematics of Information Technology and Complex Systems, Canadian National Centres of Excellence. JG is a Scholar of the Michael Smith Foundation for Health Research.

References

Corcoran C, Metha CR, Patel NR, Senchaudhuri P (2001). "Computational Tools for Exact Conditional Logistic Regression." *Statistics in Medicine*, **20**, 2723–2739.

Cytel Inc (2006a). **LogXact** *8: Discrete Regression Software Featuring Exact Methods*. Cytel Software Corporation, Cambridge, MA. URL http://www.cytel.com/.

Cytel Inc (2006b). "**LogXact** 8 Examples." URL http://www.cytel.com/Products/Logxact/Examples.asp.

Forster JJ, McDonald JW, Smith PWF (1996). "Monte Carlo Exact Conditional Tests for Log-linear and Logistic Models." *Journal of the Royal Statistical Society B*, **58**, 445–453.

Forster JJ, McDonald JW, Smith PWF (2003). "Markov Chain Monte Carlo Exact Inference for Binomial and Multinomial Logistic Regression Models." *Statistics and Computing*, **13**, 169–177.

Geyer CJ (1992). "Practical Markov Chain Monte Carlo." *Statistical Science*, **7**, 473–511.

Gilks WR, Richardson S, Spiegelhalter DJ (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, Boca Raton, Florida.

Graham J, Kockum I, Sanjeevi CB, Landin-Olsson M, Nystrom L, Sundkvist G, Arnqvist H, Blohme G, Lithner F, Littorin B, Schersten B, Wibell L, Ostman J, Lernmark A, Breslow N, Dahlquist G for the Swedish Childhood Diabetes Study Group (1999). "Negative Association Between Type 1 Diabetes and HLA DQB1*0602-DQA1*0102 is Attenuated with Age at Onset." *European Journal of Immunogenetics*, **26**, 117–127.

Hirji KF (2006). *Exact Analysis of Discrete Data*. Chapman and Hall/CRC, Boca Raton, Florida.

Mehta CR, Patel NR (1995). "Exact Logistic Regression: Theory and Examples." *Statistics in Medicine*, **14**, 2143–2160.

Mehta CR, Patel NR, Senchaudhuri P (2000). "Efficient Monte Carlo Methods for Conditional Logistic Regression." *Journal of The American Statistical Association*, **95**(449), 99–108.

Oster R (2002). "An Examination of Statistical Software Packages for Categorical Data Analysis Using Exact Methods." *The American Statistician*, **56**(3), 235–246.

Oster R (2003). "An Examination of Statistical Software Packages for Categorical Data Analysis Using Exact Methods – Part II." *The American Statistician*, **57**(3), 201–213.

Plummer M, Best N, Cowles K, Vines K (2006). "**coda**: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11. URL http://CRAN.R-project.org/doc/Rnews/.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

SAS Institute Inc (2003). *SAS/STAT Software, Version 9.1*. Cary, NC. URL http://www.sas.com/.

StataCorp (2007). *Stata Statistical Software: Release 10*. StataCorp LP, College Station, TX. URL http://www.stata.com/.

Zamar D (2006). *Markov Chain Monte Carlo Exact Inference For Logistic Regression Models*. Master's thesis, Statistics and Actuarial Science: Simon Fraser University.

Affiliation:

David Zamar, Brad McNeney, Jinko Graham
Department of Statistics and Actuarial Science
Simon Fraser University
Burnaby BC V5A 1S6, Canada
E-mail: dzamar@sfu.ca, mcneney@stat.sfu.ca, jgraham@stat.sfu.ca
URL: http://www.sfu.ca/~dzamar, http://www.stat.sfu.ca/~mcneney, http://www.stat.sfu.ca/~jgraham

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 21, Issue 3 *Submitted:* 2007-01-15
October 2007 *Accepted:* 2007-09-03