

Stat 8931 (Exponential Families) Lecture Notes
Maximum Likelihood Estimation for Potts Models
Charles J. Geyer
March 1, 2024

1 Licence

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License <http://creativecommons.org/licenses/by-sa/4.0/>.

2 R

- The version of `R` used to make this document is 4.3.2.
- The version of R package `knitr` used to make this document is 1.45.
- The version of R package `potts` used to make this document is 0.5.11.

Load libraries.

```
library("potts")
```

Also set the random number generator seed for reproducibility.

```
set.seed(42)
```

3 Introduction

We use the method of (Geyer, 1990, Chapter 6).

4 Data

First we generate some data.

```

ncolor <- as.integer(2)
beta <- log(1 + sqrt(ncolor))
theta <- c(rep(0, ncolor), beta)

nrow <- 1000
ncol <- 1000
x <- matrix(1, nrow = nrow, ncol = ncol)
foo <- packPotts(x, ncolor)

out <- potts(foo, theta, nbatch = 1, nspac = 1000)
cstat <- as.vector(out$batch)
cstat

## [1] 747318 252682 1704974

```

According to the documentation (`?potts::potts`), the components of the canonical statistic vector (`cstat`) are the number of pixels of color one, the number of pixels of color two, and the number of concordant (same color) neighbor pairs of pixels. Each of the first two has a maximum value (with toroidal boundary conditions, which are being used here) of $nrow * ncol = 10^6$. The last has a maximum value (with toroidal boundary conditions) of $2 * nrow * ncol = 2 \times 10^6$.

5 Sampling the Distribution

5.1 Markov Chain Monte Carlo Sample

Now we want a sample from the distribution of the canonical statistic vector. This has to be a Markov chain Monte Carlo (MCMC) sample, since there is no way to do ordinary Monte Carlo (OMC) that we have implemented in R ([Propp and Wilson, 1996](#), show how to do OMC with the assistance of Markov chains).

```

out <- potts(out, nbatch = 1e4, nspac = 1)

```

5.2 Diagnostic Plots

We do some MCMC diagnostic plots on our sample, particularly looking at the third canonical statistic (number of concordant neighbor pairs).

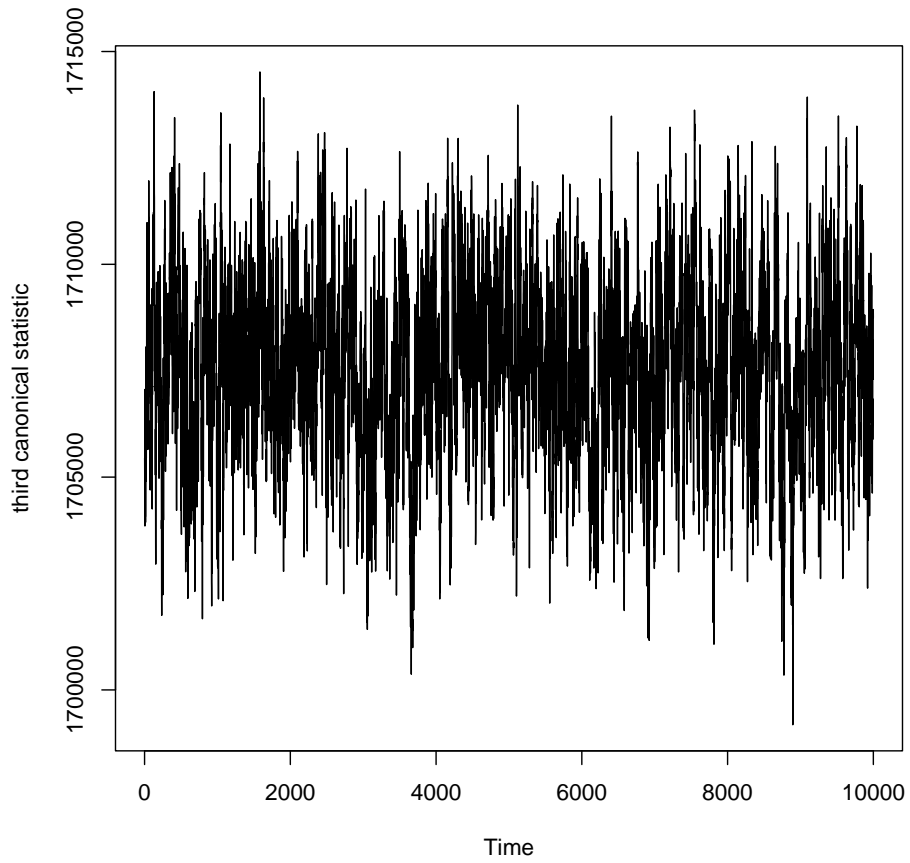


Figure 1: Time series plot of third canonical statistic (number of concordant neighbor pairs) for Ising model with zero external field and at critical temperature.

Figure 1 (page 3) shows a time series plot for the third canonical statistic. It shows no trend and good mixing.

Figure 2 (page 5) shows statistically significant autocorrelation perhaps to lag 100 or perhaps to lag 150 (the negative sample autocorrelations are probably just noise).

5.3 Distribution of the Canonical Statistic Vector

5.3.1 At Critical Temperature

Figure 3 (page 6) shows the distribution of the canonical statistic vectors as a scatterplot.

5.3.2 Above and Below Critical Temperature

We redo Figure 3 changing β to be above critical temperature (lower β , lower dependence, magnetization disappears).

```
frac.above <- 0.997
theta <- c(rep(0, ncolor), frac.above * beta)
out.above <- potts(out, theta, nbatch = 1e4, nspac = 1)
```

```
frac.below <- 1.003
theta <- c(rep(0, ncolor), frac.below * beta)
out.below <- potts(out, theta, nbatch = 1e4, nspac = 1)
```

Figure 4 (page 7) shows this distribution.

6 One Dimension

6.1 Monte Carlo Likelihood Approximation

Here we assume we know the first two canonical parameters (external field) are zero, so we have only the third canonical parameter to estimate.

We use the details of being careful with inexact computer arithmetic discussed in Geyer (2020, Section 9).

The approximate log likelihood is given by

$$l(\beta) = x_{\text{obs}}\beta - \log \left(\sum_{i=1}^n e^{x_i(\beta - \beta_{\text{sim}}) - c} \right)$$

where

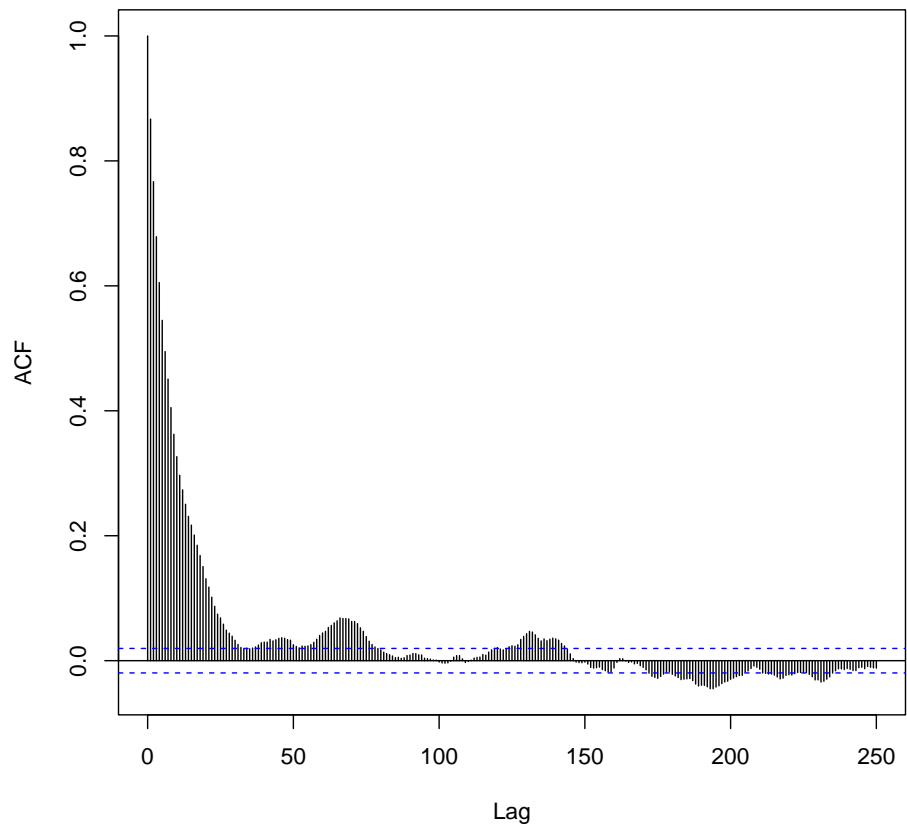


Figure 2: Autocorrelation plot of third canonical statistic (number of concordant neighbor pairs) for Ising model with zero external field and at critical temperature.

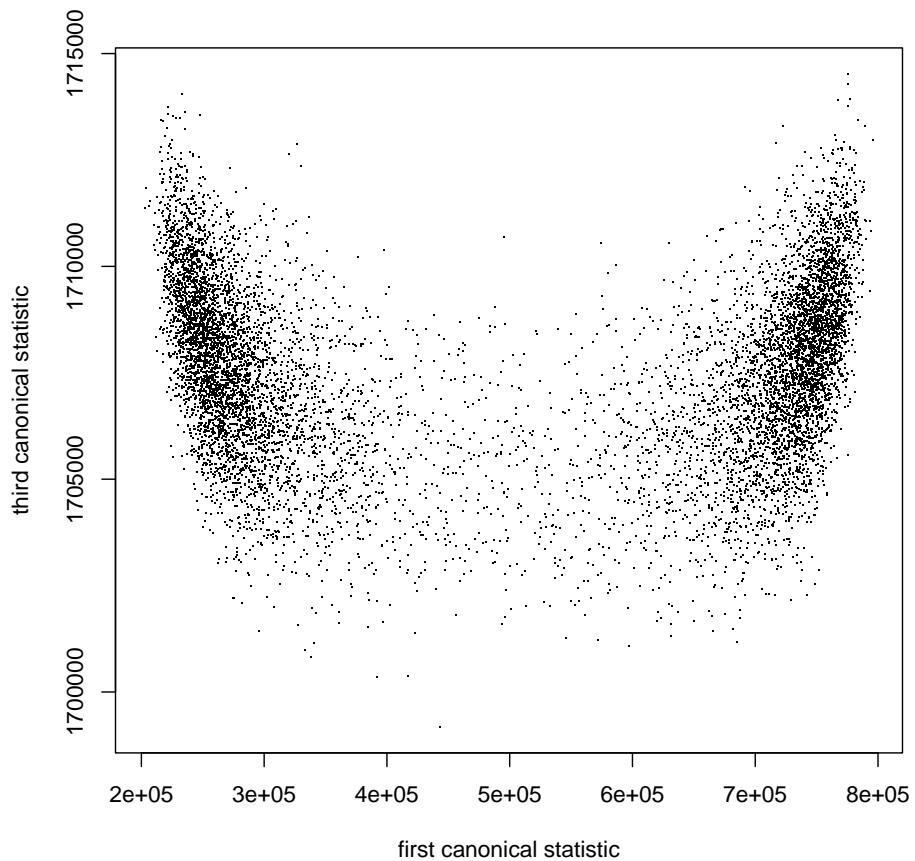


Figure 3: Scatterplot of canonical statistics for Ising model at critical temperature ($\beta = 0.88137$) and zero external field. First canonical statistic is number of pixels of first color. Third canonical statistic is number of concordant neighbor pairs of pixels. (Second canonical statistic is redundant, lattice size minus first canonical statistic.)

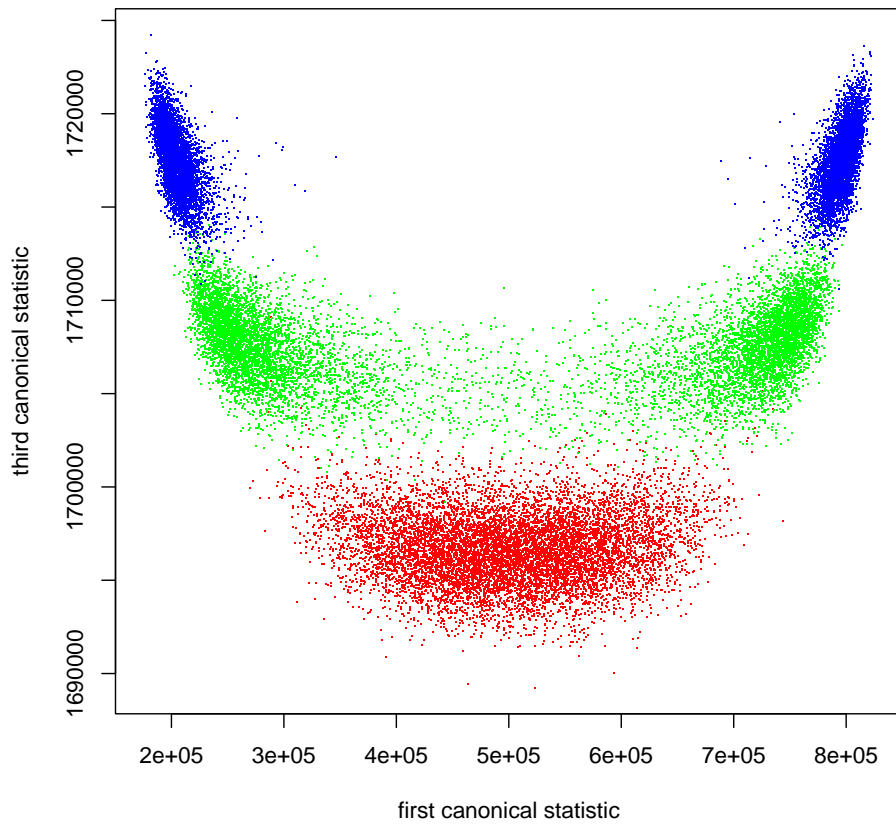


Figure 4: Scatterplot of canonical statistic vector for Ising models. All models have external field zero. Red dots sample from 0.997 of critical canonical parameter value. Green dots sample from critical canonical parameter value. Blue dots sample from 1.003 of critical canonical parameter value.

- x_{obs} is the observed value of the (third) canonical statistic.
- β is the corresponding canonical parameter, a variable.
- β_{sim} is the another value of this canonical parameter, a known constant.
- x is a vector having components x_i which is a MCMC sample from the distribution having parameter β_{sim} .
- c is an arbitrary constant, which we choose to be

$$c = \max_{1 \leq i \leq n} x_i (\beta - \beta_{\text{sim}})$$

This choice of c avoids overflow and minimizes underflow and catastrophic cancellation in the log.

We use a function factory to make the log likelihood.

```
logl.factory <- function(xobs, x, beta.sim) {
  stopifnot(is.numeric(xobs))
  stopifnot(is.finite(xobs))
  stopifnot(length(xobs) == 1)
  stopifnot(is.numeric(x))
  stopifnot(is.finite(x))
  stopifnot(is.numeric(beta.sim))
  stopifnot(is.finite(beta.sim))
  stopifnot(length(beta.sim) == 1)
  function(beta) {
    stopifnot(is.numeric(beta))
    stopifnot(is.finite(beta))
    stopifnot(length(beta) == 1)
    foo <- x * (beta - beta.sim)
    foomax <- max(foo)
    i <- which(foo == foomax)
    i <- i[1] # just in case there was more than one largest term
    foo <- foo[- i]
    bar <- foomax + log1p(sum(exp(foo - foomax)))
    xobs * beta - bar
  }
}

logl <- logl.factory(cstat[3], out$batch[ , 3], beta)
```


6.2 Monte Carlo Maximum Likelihood Estimate

Now find the (Monte Carlo approximation to the) maximum likelihood estimate (MCMLE) of this canonical parameter.

```
out <- optimize(logl, lower = beta / 2, upper = beta * 2, maximum = TRUE)
beta.hat <- out$maximum
beta.hat

## [1] 0.8808082
```

6.3 Parametric Bootstrap of MCMLE

In a real application we would not know the true unknown parameter value. So we would simulate from the distribution given by the MCMLE $\hat{\beta}$.

We get a smaller sample for the bootstrap than we used to approximate the log likelihood. This is important. In order for the approximate log likelihood to be good over the whole range of estimates, we need a very good likelihood approximation. Hence a very large sample for that.

```
theta.hat <- c(rep(0, ncolor), beta.hat)
out.boot <- potts(out, theta.hat, nbatch = 199, nspac = 50)
```

Now we estimate for each bootstrap data point.

```
xboot <- out.boot$batch[ , 3]
nboot <- length(xboot)
beta.star <- double(nboot)
for (iboot in 1:nboot) {
  logl <- logl.factory(xboot[iboot], out$batch[ , 3], beta)
  out <- optimize(logl, lower = beta / 2, upper = beta * 2, maximum = TRUE)
  beta.star[iboot] <- out$maximum
}
```

Now we look at the bootstrap distribution (Figure 5, page 10). That is remarkably ugly. How many bootstrap estimates fail?

```
sum(beta.star < 0.8)

## [1] 2
```

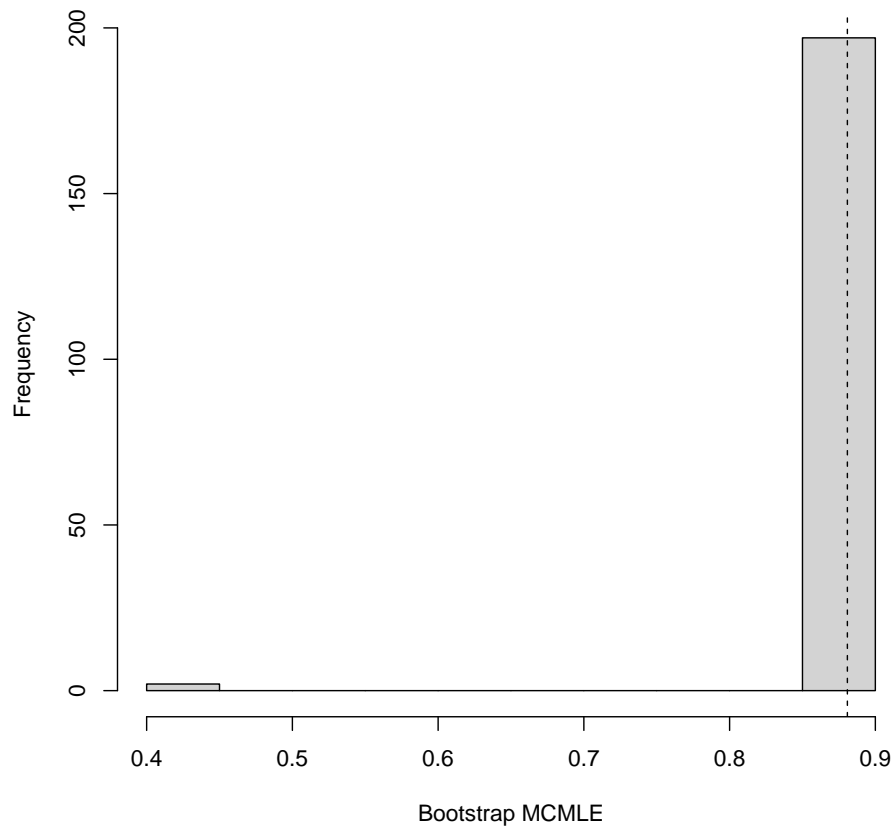


Figure 5: Histogram of bootstrap distribution of MCMLE. Vertical dashed line is MCMLE.

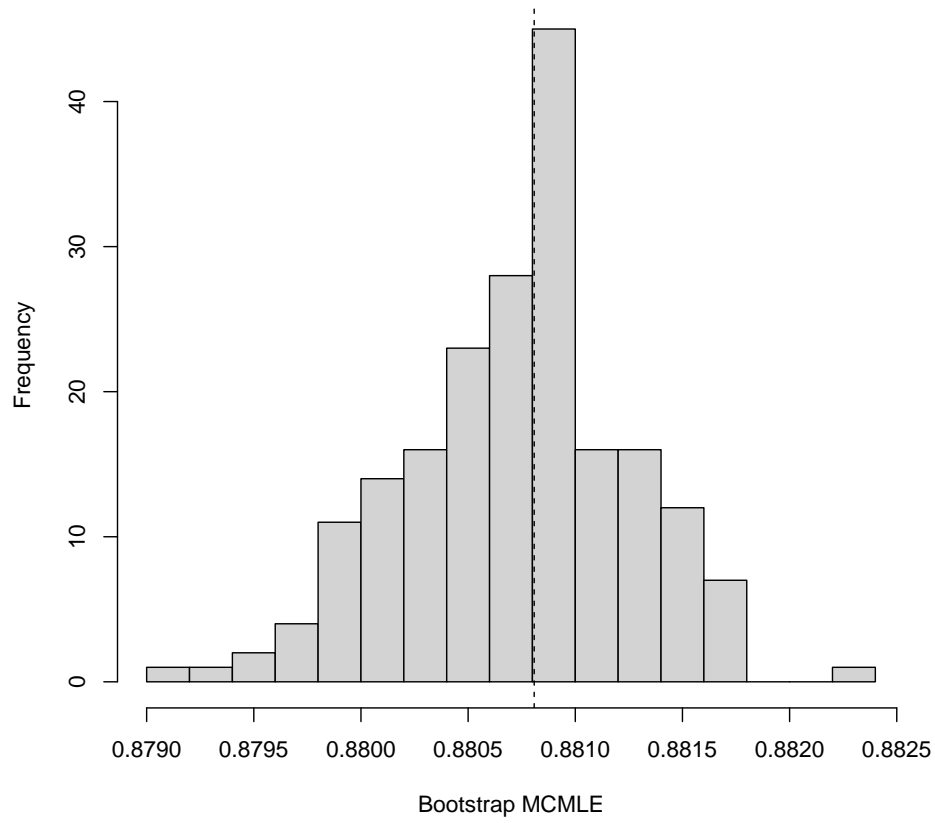


Figure 6: Histogram of bootstrap distribution of MCMLE. Vertical dashed line is MCMLE. Two bootstrap estimates were off the figure to the left.

Just 2 out of 199. So that is not a problem.

Try again (Figure 6, page 11).

Since this distribution seems to be fairly close to symmetric about the MLE (which plays the role of the true unknown parameter in the bootstrap world) perhaps a bootstrap percentile interval is not too bad.

We chose bootstrap sample size 199 so that order statistics would be natural estimators of quantiles having denominators 200. So for 95% confidence two-sided we choose quantiles

```
conf.level <- 0.95
idx <- (length(beta.star) + 1) * c(1 - conf.level, 1 + conf.level) / 2
idx

## [1] 5 195

sort(beta.star)[idx]

## [1] 0.8794244 0.8816619

beta

## [1] 0.8813736
```

We see that our confidence interval happens to work. Although, of course, it needn't have. It is supposed to miss 5% of the time.

References

- Geyer, C. J. (1990). Likelihood and Exponential Families. PhD thesis, University of Washington. <https://hdl.handle.net/11299/56330>.
- Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society, Series B*, **56**, 261–274. doi:[10.1111/j.2517-6161.1994.tb01976.x](https://doi.org/10.1111/j.2517-6161.1994.tb01976.x).
- Geyer, C. J. (2013). Asymptotics of maximum likelihood without the LLN or CLT or sample size going to infinity. In *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, G. L. Jones and X. Shen eds. *IMS Collections*, **10**, 1–24. Institute of Mathematical Statistics: Hayward, CA. doi:[10.1214/12-IMSCOLL1001](https://doi.org/10.1214/12-IMSCOLL1001).

- Geyer, C. J. (2020). Statistics 3701 (Geyer, Fall 2020) Lecture Notes: Computer Arithmetic. <https://www.stat.umn.edu/geyer/3701/notes/arithmetric.html>.
- Okabayashi, S., Johnson, L. and Geyer, C. J. (2011). Extending pseudolikelihood for Potts models. *Statistica Sinica*, **21**, 331–347. <https://www.jstor.org/stable/24309274>.
- Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, **9**, 223–252. doi:10.1002/(SICI)1098-2418(199608/09)9:1/2%3C223::AID-RSA14%3E3.0.CO;2-O.
- Sung, Y. J. and Geyer, C. J. (2007). Monte Carlo likelihood inference for missing data models. *Annals of Statistics*, **35**, 990–1011. doi:10.1214/009053606000001389. <https://github.com/cjgeyer/bernor>.