ISC 5228
Markov Chain Monte Carlo

Missing Data and Gibbs Sampling

1 Introduction

Suppose you were recently hired as the manager of a high-performance computing center. In the past year ($N = 365$ days), someone logged the number of compute nodes that failed each day. At the end of the year, a summary table was prepared, which is now available to you.[1]

j = failures/day	n_j times
0	142
1	129
2	56
3	25
4+	13

1.1 Goal

You want to find the mean rate of failure λ, assuming a Poisson process for failure.

$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, ..., \infty \tag{1}$$

1.2 Complications

Unfortunately, the original raw data was lost, and the only the summary table has been salvaged. Note the catch-all category "4+". Ideally, we would have liked a better breakdown (4, 5, 6, ..., instead of just "4+"). In the ideal case for instance, one could try to fit the histogram with the Poisson distribution (eqn 1) to determine λ.

This is a garden-variety *missing data problem*.

We have good data ($\mathbf{x}$) on 365 - 13 = 352 days; the remaining 13 days, we have missing data (latent variables $\mathbf{z}$). However, we know **something** about the latent variables; the failure/day on those days was 4 or more.

2 Setup

Let $\mathbf{x} = (x_1, x_2, ..., x_{352})$ represent the observed data. x_i is the number of failures on a particular day, and takes a value between 0 and 3. Let $\mathbf{z} = (z_1, z_2, ..., z_{13})$ represent the missing data. Each z_i takes a value between 4 and infinity.

How can we combine our knowledge of observed and missing data, and the assumption of an underlying Poisson process, to come up with an estimate (PDF) of λ?

[1] This example adapted from Casella class notes.

One potential answer is suggested by Bayesian analysis. We want to find $p(\lambda|\mathbf{x})$. Let's try to develop an expression for this.

$$p(\lambda|\mathbf{x}) = \int p(\lambda|\mathbf{x}, \mathbf{z})\, d\mathbf{z}, \qquad \text{marginalization} \tag{2}$$

$$= \int p(\mathbf{x}, \mathbf{z}|\lambda)\, p(\lambda)\, d\mathbf{z} \qquad \text{Bayes formula} \tag{3}$$

$$= \int p(\mathbf{z}|\mathbf{x}, \lambda)\, p(\mathbf{x}|\lambda)\, p(\lambda)\, d\mathbf{z} \tag{4}$$

The last step uses the relationship between joint, marginal and conditional distributions, namely $p(A, B) = p(A|B)\ p(B)$.

2.1 Model

2.1.1 Prior

Let us set the prior as $p(\lambda) \sim 1/\lambda$ in equation 4.

2.1.2 Likelihood

For the 352 time intervals in which data was observed, assuming a Poisson process, we have

$$p(\mathbf{x}|\lambda) = \prod_{i=1}^{352} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

For the 13 time intervals in which data is missing,

$$p(\mathbf{z}|\mathbf{x}, \lambda) = p(\mathbf{z}|\lambda) \sim \prod_{i=1}^{13} \frac{e^{-\lambda}\lambda^{z_i}}{z_i!}, \qquad z_i > 4.$$

This is sometimes called a **truncated Poisson distribution**. Thus, all the ingredients in eqn. 4 are completely specified.

The resulting joint probability distribution (eqn. 3) is therefore the product:

$$p(\mathbf{x}, \mathbf{z}|\lambda) \sim \frac{e^{-365\lambda}\lambda^{\sum x_i + \sum z_i}}{\prod x_i! \prod z_i!}.$$

With $p(\lambda) = 1/\lambda$, this implies that,

$$p(\lambda|\mathbf{x}, \mathbf{z}) \sim e^{-365\lambda}\lambda^{\sum x_i + \sum z_i - 1} \sim \text{Gamma}(\sum x_i + \sum z_i, 365).$$

Recall that

$$\text{Gamma}(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

Note that,

$$\sum x_i = \sum_{j=0}^{3} j n_j = 0 \times 142 + 1 \times 129 + 2 \times 56 + 3 \times 25 = 316.$$

This would suggest a failure rate of the order of $\lambda \approx 316/352 \approx 1$.

3 Strategy

We have (i) a (truncated Poisson) model for the process generating the missing data $p(\mathbf{z}|\lambda)$, and (ii) a (Gamma) model for the posterior from which the latent variables can be marginalized out. This gives us a Gibbs sampling scheme to sample λ.

- Given $\lambda^{(t-1)}$, pick 13 samples Z_i from the truncated Poisson distribution:

$$Z_i^{(t)} \sim \text{TruncPossion}(\lambda^{(t-1)}, 4)$$

- Sample $\lambda^{(t)}$ from Gamma distribution

$$\lambda^{(t)} \sim \text{Gamma}\left(\sum x_i + \sum Z_i^{(t)}, 365\right).$$

We will have to figure out a way to sample from the non-standard truncated Poisson distribution, and the Gamma distribution.[2]

4 Exercises

- Estimate and plot the distribution $p(\lambda|\mathbf{x})$, after discarding burn-in. Use the Gelman-Rubin diagnostic to estimate a suitable simulation length.
- Estimate and plot the distribution $p(\mathbf{z}|\mathbf{x}, \lambda)$.[3]
- Repeat the two steps above, if the last two rows of the data table were replaced by "3+ = 38" (instead of separate rows for "3" and "4+").

5 Appendix: Python Code

Here is python code that exploits built-in routines from the scipy.stats library to sample from the truncated Poisson distribution. In particular, it uses the functions `cdf` which yields the cumulative distribution function for a given value of x, and `ppf` which does the inverse;[4] it yields an x that corresponds to a certain value for the CDF.

The basic idea is the transformation method $x = F^{-1}(u)$.

```
import scipy.stats as stats

def TruncatedPoisson(mu, kmin, nsamples = 1):
    """ Truncated Poisson, values >=k; mu is same as lambda
    This effectively uses x = F^{-1}(u) technique;
    exploits built-in python functions"""
```

[2] If you cannot find built-in routines, these links may be useful: http://www.hongliangjie.com/2012/12/19/how-to-generate-gamma-random-variables/

[3] show the histogram of the values of Z sampled during the simulation.

[4] essentially F^{-1}

```
    # normalization factor. Subtract pbty of truncated part
    nrm = 1.0 - stats.poisson.cdf(kmin-1, mu)

    # u = values between cdf(k) and 1; the second term is the random part
    yr = stats.poisson.cdf(kmin-1, mu)  + np.random.rand(nsamples)*(nrm)

    # inverse CDF
    xr = stats.poisson.ppf(yr, mu)

    # maps them to integers
    return xr.astype(int)
```

In python, we can sample from Gamma(α, β) by using the function `rvs` as `gamma.rvs(alpha, scale=1/beta)`.