A computational framework for saddlepoint methods

Godrick Oketch

November 28, 2023

Saddlepoint approximation

Consider a θ-dependent random variable Y whose PDF/PMF $\boxed{f(y;\theta)}$ is unknown.

Known MGF: $M_Y(t;\theta) = \mathrm{E}(e^{tY})$

Saddlepoint approximation transforms the cumulant generating function (CGF) $\boxed{K_Y(t;\theta) = \log M_Y(t;\theta)}$

$$\hat{f}(y;\theta) = \frac{\exp(K_Y(\hat{t};\theta) - \hat{t}y)}{\sqrt{\det(2\pi K_Y''(\hat{t};\theta))}}, \text{ where}$$
$$K_Y'(\hat{t};\theta) = y$$

Saddlepoint approximation

Consider a θ-dependent random variable Y whose PDF/PMF $f(y;\theta)$ is unknown.

To estimate θ, using y

$L(\theta; y)$; Not possible

Known MGF: $M_Y(t;\theta)$

Saddlepoint approximation transforms the cumulant generating function (CGF)

$$K_Y(t;\theta) = \log M_Y(t;\theta)$$

$$\hat{f}(y;\theta) = \frac{\exp(K_Y(\hat{t};\theta) - \hat{t}y)}{\sqrt{\det(2\pi K_Y''(\hat{t};\theta))}}, \text{ where}$$

$$K_Y'(\hat{t};\theta) = y$$

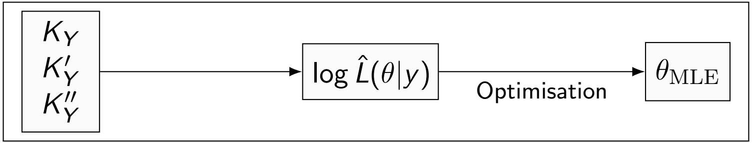
$$\hat{L}(\theta; y) = \hat{f}(y;\theta)$$ - "saddlepoint likelihood"

Saddlepoint likelihood

To estimate θ, we refer to

$$\log \hat{L}(\theta|y) = K_Y(\hat{t};\theta) - \hat{t}y - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log\det\{K_Y''(\hat{t};\theta)\},$$

where $\hat{t} = \hat{t}(\theta; y)$ is the solution of the saddlepoint equation, i.e., $K_Y'(t;\theta) = y$.

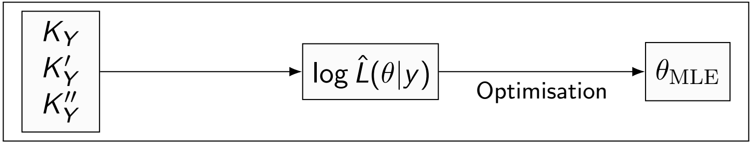


Saddlepoint likelihood

To estimate θ, we refer to

$$\log \hat{L}(\theta|y) = K_Y(\hat{t};\theta) - \hat{t}y - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log\det\{K''_Y(\hat{t};\theta)\},$$

where $\hat{t} = \hat{t}(\theta; y)$ is the solution of the saddlepoint equation, i.e., $K'_Y(t;\theta) = y$.



What if $Y = \sum_{i=1}^{N} \tilde{X}_i$

$\tilde{X}_i \sim \mathrm{Multinomial}(n*, \pi_i)$

$N = \mathrm{Binomial}(n_1, p_1) + \mathrm{Binomial}(n_2, p_2) + \mathrm{Binomial}(n_3, p_3)$

- Consider a random variable $Y = \sum_{i=1}^{n} X_i$, where X_i's are i.i.d copies of X.

> If we know the CGF of X, we can exploit these operations to obtain the CGF of Y.

- Essentially, we transform $\{K_X(t;\varphi), K'_X(t;\varphi), K''_X(t;\varphi)\}$ to $\{K_Y(t;\theta), K'_Y(t;\theta), K''_Y(t;\theta)\}$
- The *"distributional parameter"*, φ is modelled by θ(the *"model parameter"*).
- As a function, $\varphi = h(\theta)$ - *"adaptor"*

Building a model CGF -Multivariate Poisson r.v.

We observe vector $Y = (Y_1, \ldots, Y_d)$ which follow a multivariate Poisson distribution such that

$$\begin{aligned} Y_1 &= X_1 + Z_0 \\ Y_2 &= X_2 + Z_0 \\ &\vdots \\ Y_d &= X_d + Z_0. \end{aligned} \tag{1}$$

X_i's and Z_0 are unobservable and independent Poisson random variables with distributional parameters α and β.

> Goal: Estimate $\theta = (\alpha, \beta)$ using the observations - Y.

- We can represent (1) as $Y = X + Z$, where $X = (X_1, \ldots, X_d)$ is a vector of i.i.d Poisson random variables and Z is a vector with d repeated components of Z_0.
- Structurally, $Y = X + AZ_0$, A: a column vector of ones.

Building a model CGF -Multivariate Poisson r.v.

We observe vector $Y = (Y_1, \ldots, Y_d)$ which follow a multivariate Poisson distribution such that

$$
\begin{aligned}
Y_1 &= X_1 + Z_0 \\
Y_2 &= X_2 + Z_0 \\
&\vdots \\
Y_d &= X_d + Z_0.
\end{aligned}
\qquad (1) \qquad
Y = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} Z_0
$$

X_i's and Z_0 are unobservable and independent Poisson random variables with distributional parameters α and β.

Goal: Estimate $\theta = (\alpha, \beta)$ using the observations - Y.

- We can represent (1) as $Y = X + Z$, where $X = (X_1, \ldots, X_d)$ is a vector of i.i.d Poisson random variables and Z is a vector with d repeated components of Z_0.
- Structurally, $Y = X + AZ_0$, A: a column vector of ones.

Building a model CGF -Multivariate Poisson r.v.

Goal: Estimate $\theta = (\alpha, \beta)$ using $Y = (Y_1, \ldots, Y_d)$.

$\{Y = X + Z\} = \{Y = X + AZ_0\}$; X a vector of i.i.d $\mathrm{Poisson}(\alpha)$ and $Z_0 \sim \mathrm{Poisson}(\beta)$

```
# CGF of X(i.i.d Poisson random variable):
K_X <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 1))
```

Building a model CGF -Multivariate Poisson r.v.

Goal: Estimate $\theta = (\alpha, \beta)$ using $Y = (Y_1, \ldots, Y_d)$.

$\{Y = X + Z\} = \{Y = X + AZ_0\}$; X a vector of i.i.d $\text{Poisson}(\alpha)$ and $Z_0 \sim \text{Poisson}(\beta)$

```
# CGF of X(i.i.d Poisson random variable):
K_X <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 1))

# CGF of Z = A*Z_0
K_Z0 <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 2))
A <- matrix(1, nrow = d)
K_Z <- linearlyMappedCGF(baseCGF = K_Z0, matrix_A = A)
```

Building a model CGF -Multivariate Poisson r.v.

Goal: Estimate $\theta = (\alpha, \beta)$ using $Y = (Y_1, \ldots, Y_d)$.

$\{Y = X + Z\} = \{Y = X + AZ_0\}$; X a vector of i.i.d $\mathrm{Poisson}(\alpha)$ and $Z_0 \sim \mathrm{Poisson}(\beta)$

```
# CGF of X(i.i.d Poisson random variable):
K_X <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 1))

# CGF of Z = A*Z_0
K_Z0 <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 2))
A <- matrix(1, nrow = d)
K_Z <- linearlyMappedCGF(baseCGF = K_Z0, matrix_A = A)

# The CGF of Y
K_Y <- sumOfIndependentCGF(K_X, K_Z)
```

find.saddlepoint.MLE() for $\theta = (\alpha, \beta)$

$\{Y = X + Z\} = \{Y = X + AZ_0\}$; X a vector of i.i.d $\text{Poisson}(\alpha)$ and $Z_0 \sim \text{Poisson}(\beta)$

```
find.saddlepoint.MLE(observed.data = Y, model.cgf = K_Y,
                     starting.theta = c(1,1), std.error = TRUE)
```

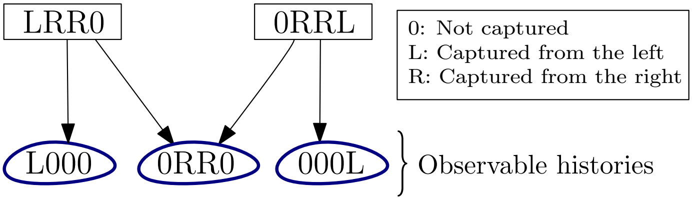
```
K_X <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 1))
K_Z0 <- PoissonModelCGF(lambda = adaptorUsingIndices(indices = 2))
K_Z <- linearlyMappedCGF(baseCGF = K_Z0, matrix_A = matrix(1, nrow = d))
K_Y <- sumOfIndependentCGF(K_X, K_Z)
```

Capture-recapture models with latent identities

$$Y = AX$$

- Latent identities in such models occur in such a way that X can be modelled but is not observable, and Y cannot be modelled but is observable.

Example: Two-source (Multimark) model



- There is no way of matching these observed histories to the animals that produced them.

Capture-recapture models with latent identities

$$Y = AX; \;\; X \sim \mathrm{Multinomial}(N, \pi); \;\; \theta = (N, p_L, p_R)$$

```
h <- function(theta) pi
grad.h <- function(theta) {...}
A <- ...
-----------------------------------
K.X <- MultinomialModelCGF(n = adaptorUsingIndices(indices = 1),
                  prob.vec = adaptorUsingRFunctions(h = h, grad_h = grad.h))
K.Y <- linearlyMappedCGF(baseCGF = K.X, matrix_A = A)
-----------------------------------
find.saddlepoint.MLE(observed.data = Y, model.cgf = K.Y,
                     starting.theta = ...)
```

Summary

- This framework allows us to easily create and compute MGFs/CGFs and their derivatives.
- We exploit "CGF-compatible" operations as our building blocks: linear mapping operation, sum of independent r.vs operation, operations involving compound distributions, e.t.c
- For estimation using the saddlepoint likelihood, the framework provides a streamlined and intuitive way of building CGFs. (The knowledge of the actual CGF of a observable random variable is unnecessary to obtain estimates. We can use the framework to directly build and utilise them.)
- The framework is extensible and allows for the addition of new operations and CGFs.

A different approach - MV Poisson problem

We observe vector $Y = (Y_1, \dots, Y_d)$ which follow a multivariate Poisson distribution such that

$$\begin{aligned} Y_1 &= X_1 + Z_0 \\ Y_2 &= X_2 + Z_0 \\ &\vdots \\ Y_d &= X_d + Z_0. \end{aligned} \qquad Y = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} Z_0$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ Z_0 \end{pmatrix}$$

CGF of Y will involve a *"linear mapping"* operation of a *"concatenated"* CGF.