cvLME

A multi-language library to perform cross-validated Bayesian model selection

Joram Soch, BCCN Berlin

Toolbox URL:	https://github.com/JoramSoch/cvLME
Support Contact:	Joram Soch ⟨joram.soch@bccn-berlin.de⟩
Current Version:	cvLME V0.3.3 [V0.languages.models]
Related Paper:	Soch J, Allefeld C (2018). MACS – a new SPM toolbox for model assessment, comparison and selection. *Journal of Neuroscience Methods*, vol. 306, pp. 19-31; DOI: 10.1016/j.jneumeth.2018.05.017.

Contents

0 General remarks

0.1 Mathematics

In the following sections, we are considering different model classes – such as linear regression for continuous data (see Section 2) or a Poisson model for count data (see Section 3). For each of these model classes – except for the section on model comparison where only some measures of model selection are introduced (see Section 1) –, we are going through several steps of mathematical derivation which are outlined here.
Step 1: First, the likelihood function is specified,

$$p(y|\theta, m) \; , \tag{0.1}$$

i.e. the probability of observing the data y, given a model m and parameters θ.
Step 2: Second, the maximum likelihood estimates are derived,

$$\hat{\theta} = \arg\max_{\theta} \log p(y|\theta, m) \; , \tag{0.2}$$

i.e. those model parameters that maximize the log-likelihood function $\mathrm{LL}(\theta)$.
Step 3: Third, a (conjugate) prior distribution is specified,

$$p(\theta|m) \; , \tag{0.3}$$

i.e. a distribution over parameters that can be applied to the likelihood $p(y|\theta, m)$.
Step 4: Then, the joint likelihood function is calculated,

$$p(y, \theta|m) = p(y|\theta) \, p(\theta|m) \; , \tag{0.4}$$

i.e. the product of likelihood function and prior distribution over model parameters.
Step 5: Then, the posterior distribution is obtained,

$$p(\theta|y, m) = \frac{p(y|\theta, m) \, p(\theta|m)}{p(y|m)} \propto p(y, \theta|m) \; , \tag{0.5}$$

i.e. a distribution over model parameters that is proportional to the joint likelihood.
Step 6: Then, the log model evidence is derived,

$$\mathrm{LME}(m) = \log \int p(y|\theta) \, p(\theta|m) \, \mathrm{d}\theta = \log \frac{p(y|\theta, m) \, p(\theta|m)}{p(\theta|y, m)} \; , \tag{0.6}$$

i.e. the logarithm of the marginal likelihood, expected over the prior distribution.
Step 7: Then, the cross-validated log model evidence is presented,

$$\mathrm{cvLME}(m) = \sum_{i=1}^{S} \log \int p(y_i|\theta, m) \, p(\theta| \cup_{j \neq i} y_j, m) \, \mathrm{d}\theta \tag{0.7}$$

which basically consists in describing what kind of non-informative prior is used for obtaining the informative posterior from the training data in each cross-validation fold.
Steps 8 and 9: Finally, special cases of each model class are considered and implementation in the different programming languages is described.

0.2 Implementation

The cross-validated log model evidence (cvLME; see eq. 0.7) is implemented within several programming languages and the respective implementational details are described here. In what follows, `<name-of-the-model-class>` is either "`GLM`" or "`Poiss`" (but more options are to follow in the future).

MATLAB: In MATLAB, the different methods for each model class (maximum likelihood estimation, Bayesian estimation, log model evidence, cross-validated log model evidence) are implemented as different functions called "`<name-of-the-model-class>_MLE/Bayes/LME/cvLME.m`" and these functions can be directly called with variables representing quantities in the model, e.g. measured data or experimental design information. Often, some quantities can be left empty.

Python: In Python, all methods for all model classes are implemented as a single module called "`cvBMS.py`" which can be simply imported at the beginning of a script using "`import cvBMS`". Then, a particular model is intialized by calling "`model = cvBMS.<name-of-the-model-class>`" with variables representing model quantities. When a model has been initialized, statistical operations can be performed in object-oriented fashion by typing "`model.MLE/Bayes/LME/cvLME()`" where sometimes, some more input variables can or must be provided.

Further details are provided in the implementation subsections of each model class individually (see Sections X.9).

1 Model spaces and model selection

1.1 Log model evidence

A model space is defined as a set of models. In the context of these tools, a model space is always initialized with a set of *log model evidences* (LME)

$$\mathrm{LME}(m) = \log p(y|m) = \log \int p(y|\theta,m)\, p(\theta|m)\, \mathrm{d}\theta \tag{1.1}$$

or *cross-validated log model evidences* (cvLMEs)

$$\mathrm{cvLME}(m) = \sum_{i=1}^{S} \log \int p(y_i|\theta,m)\, p(\theta| \cup_{j \neq i} y_j, m)\, \mathrm{d}\theta \tag{1.2}$$

where S is the number of data subsets.

1.2 Log Bayes factor

The *Bayes factor* (BF) is defined as the ratio of two model evidences,

$$\mathrm{BF}_{12} = \frac{p(y|m_1)}{p(y|m_2)} \;, \tag{1.3}$$

such that the *log Bayes factor* (LBF) is the difference of two log model evidences,

$$\mathrm{LBF}_{12} = \log \mathrm{BF}_{12} = \log \frac{p(y|m_1)}{p(y|m_2)} = \mathrm{LME}(m_1) - \mathrm{LME}(m_2) \; . \tag{1.4}$$

1.3 Posterior probabilities

Given more than two models, one can also calculate *posterior model probabilities* (PPs) by simply applying Bayes' theorem to the model evidences

$$p(m_i|y) = \frac{p(y|m_i)\, p(m_i)}{\sum_{j=1}^{M} p(y|m_j)\, p(m_j)} \tag{1.5}$$

or, equivalently, to the exponentiated log model evidences (LME)

$$p(m_i|y) = \frac{\exp[\mathrm{LME}(m_i)]\, p(m_i)}{\sum_{j=1}^{M} \exp[\mathrm{LME}(m_j)]\, p(m_j)} \tag{1.6}$$

where $p(m_i)$ are prior model probabilities and M is the number of models.

Note that posterior probabilities do not on depend on absolute LME values, but only on relative LME difference. For this reason, the mean LME over models is subtracted from all LMEs before PPs are calculated.

1.4 Log family evidence

The *family evidence* (FE) is obtained by marginalizing over "model" within "family", i.e. as the marginal probability over the model evidences from all models within one family

$$p(y|f) = \sum_{m \in f} p(y|m)\, p(m|f) \tag{1.7}$$

and the *log family evidence* (LFE) is the natural logarithm of this quantity

$$\mathrm{LFE}(f) = \log p(y|f) = \log \sum_{m \in f} p(y|m)\, p(m|f) \tag{1.8}$$

where $p(m|f)$ is a (most likely uniform) within-family prior distribution.

Note that, with a uniform within-family prior, the family evidence is the average of model evidences, but the log family evidence is not the average of the log model evidences! In particular, the problem is that we usually cannot access model evidences $p(y|m)$ directly, but only deal with log model evidences $\log p(y|m)$. LMEs are used to avoid computational problems with very small model evidences that could not be stored in standard computers, e.g. $p(y|m) = 10^{-100} \Rightarrow \log p(y|m) \approx -230$. However, just exponentiating LMEs does not work, because they often fall below a specific underflow threshold $-u$, e.g. $u = 745$, so that all model evidences would be 0.
The solution is to select the maximum LME within a family

$$\mathrm{L}^*(f) = \max_{m \in f} \left[\mathrm{LME}(m)\right] \tag{1.9}$$

and define differences between LMEs and maximum LME as

$$\mathrm{L}'(m) = \mathrm{LME}(m) - \mathrm{L}^*(f) \; . \tag{1.10}$$

Then, the log family evidence can be written as

$$\mathrm{LFE}(f) = \log p(y|f) = \log \left[\frac{1}{M_f} \sum_{i=1}^{M_f} \exp\left[\mathrm{LME}(m_i)\right] \right] \tag{1.11}$$

which can be further developed in the following way:

$$\begin{aligned}
\mathrm{LFE}(f) &= \log \left[\frac{1}{M_f} \sum_{i=1}^{M_f} \exp\left[\mathrm{L}'(m_i) + \mathrm{L}^*(f)\right] \right] \\
&= \log \left[\frac{1}{M_f} \exp \mathrm{L}^*(f) \sum_{i=1}^{M_f} \exp \mathrm{L}'(m_i) \right] \\
&= \mathrm{L}^*(f) + \log \sum_{i=1}^{M_f} \exp \mathrm{L}'(m_i) - \log M_f \; .
\end{aligned} \tag{1.12}$$

1.5 Implementation

In **MATLAB**, (log) Bayes factors, posterior model probabilities and log family evidences are implemented via the functions `MS_LBF`, `MS_PP` and `MS_LFE` which have to be called with an $M \times N$ matrix `LME` as input.
In **Python**, a model space object has to be initiated via `ms = cvBMS.MS(LME)` and (log) Bayes factors, posterior model probabilities and log family evidences are calculated via `ms.LBF`, `ms.BF`, `ms.PP`, and `ms.LFE`.

2 Univariate General Linear Model

2.1 Likelihood function

In the univariate general linear model (GLM), a single measured signal (y) is modelled as a linear combination (β) of predictor variables (X), where errors (ε) are assumed to be normally distributed around zero and to have a known covariance structure (V), but unknown variance factor (σ^2):

$$y = X\beta + \varepsilon, \; \varepsilon \sim \mathrm{N}(0, \sigma^2 V) \; . \tag{2.1}$$

In this equation, y is the $n \times 1$ measured signal, X is the $n \times p$ design matrix, β is a $p \times 1$ vector of regression coefficients, ε is an $n \times 1$ vector of errors, σ^2 is the variance of these errors and V is an $n \times n$ correlation matrix where n is the number of data points and p ist the number of regressors.
The GLM equation (2.1) implies the following *likelihood function*

$$p(y|\beta, \sigma^2) = \mathrm{N}(y; X\beta, \sigma^2 V) = \sqrt{\frac{1}{(2\pi)^n |\sigma^2 V|}} \, \exp\left[-\frac{1}{2\sigma^2} (y - X\beta)^T V^{-1} (y - X\beta) \right] \tag{2.2}$$

which, for mathematical convenience, can also be parametrized as

$$p(y|\beta, \tau) = \mathrm{N}(y; X\beta, (\tau P)^{-1}) = \sqrt{\frac{|\tau P|}{(2\pi)^n}} \, \exp\left[-\frac{\tau}{2} (y - X\beta)^T P (y - X\beta) \right] \tag{2.3}$$

using the residual precision $\tau = 1/\sigma^2$ and the $n \times n$ precision matrix $P = V^{-1}$.

2.2 Maximum likelihood

Classical model estimation proceeds by maximizing the *log-likelihood* (LL)

$$\mathrm{LL}(\beta, \sigma^2) = \log p(y|\beta, \sigma^2) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log \left|\sigma^2 V\right| - \frac{1}{2\sigma^2} (y - X\beta)^T V^{-1} (y - X\beta) \tag{2.4}$$

which gives rise to *maximum-likelihood* (ML) parameter estimates

$$\begin{aligned} \hat{\beta} &= (X^T V^{-1} X)^{-1} X^T V^{-1} y \\ \hat{\sigma}^2 &= \frac{1}{n} (y - X\hat{\beta})^T V^{-1} (y - X\hat{\beta}) \end{aligned} \tag{2.5}$$

that can be used to form t- and F-statistics

$$\begin{aligned} t &= \frac{c^T \hat{\beta}}{\sqrt{\hat{\sigma}^2 c^T \mathrm{cov}(\hat{\beta}) c}} \\ F &= (C^T \hat{\beta})^T (\hat{\sigma}^2 C^T \mathrm{cov}(\hat{\beta}) C)^{-1} (C^T \hat{\beta}) \end{aligned} \tag{2.6}$$

where c is a $p \times 1$ *contrast vector*, C is a $p \times q$ *contrast matrix* and

$$\mathrm{cov}(\hat{\beta}) = (X^T V^{-1} X)^{-1} \; . \tag{2.7}$$

2.3 Prior distribution

A conjugate prior distribution relative to the likelihood function given by (2.3) is the *normal-gamma distribution* over regression coefficients β and residual precision τ

$$p(\beta, \tau) = \mathrm{N}(\beta; \mu_0, (\tau \Lambda_0)^{-1}) \cdot \mathrm{Gam}(\tau; a_0, b_0) \tag{2.8}$$

which can be split into a conditional distribution and a marginal distribution

$$\begin{aligned} p(\beta|\tau) &= \mathrm{N}(\beta; \mu_0, (\tau \Lambda_0)^{-1}) = \sqrt{\frac{|\tau \Lambda_0|}{(2\pi)^p}} \exp\left[-\frac{\tau}{2}(\beta - \mu_0)^T \Lambda_0 (\beta - \mu_0)\right] \\ p(\tau) &= \mathrm{Gam}(\tau; a_0, b_0) = \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0 - 1} \exp[-b_0 \tau] \end{aligned} \tag{2.9}$$

where μ_0 and Λ_0 are the prior mean and the prior precision of β and a_0 and b_0 are the prior shape and rate parameters for τ.

2.4 Joint likelihood

Combining the likelihood function (2.3) with the prior distribution (2.9), the *joint likelihood function* of the general linear model with normal-gamma priors (GLM-NG) becomes

$$\begin{aligned} p(y, \beta, \tau) &= p(y|\beta, \tau) \, p(\beta, \tau) = p(y|\beta, \tau) \, p(\beta|\tau) \, p(\tau) \\ &= \sqrt{\frac{|\tau P|}{(2\pi)^n}} \, \exp\left[-\frac{\tau}{2}(y - X\beta)^T P (y - X\beta)\right] \cdot \\ &\quad \sqrt{\frac{|\tau \Lambda_0|}{(2\pi)^p}} \, \exp\left[-\frac{\tau}{2}(\beta - \mu_0)^T \Lambda_0 (\beta - \mu_0)\right] \cdot \\ &\quad \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0 - 1} \exp[-b_0 \tau] \; . \end{aligned} \tag{2.10}$$

Collecting identical variables gives:

$$\begin{aligned} p(y, \beta, \tau) = &\sqrt{\frac{\tau^{n+p}}{(2\pi)^{n+p}} |P| |\Lambda_0|} \, \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0 - 1} \exp[-b_0 \tau] \cdot \\ &\exp\left[-\frac{\tau}{2} \left((y - X\beta)^T P (y - X\beta) + (\beta - \mu_0)^T \Lambda_0 (\beta - \mu_0)\right)\right] \; . \end{aligned} \tag{2.11}$$

Completing the square over β gives:

$$
\begin{aligned}
p(y,\beta,\tau) = \sqrt{\frac{\tau^{n+p}}{(2\pi)^{n+p}}|P||\Lambda_0|}\, \frac{{b_0}^{a_0}}{\Gamma(a_0)}\, \tau^{a_0-1} \exp[-b_0\tau]\cdot \\
\exp\left[-\frac{\tau}{2}\left((\beta-\mu_n)^T\Lambda_n(\beta-\mu_n) + (y^TPy + \mu_0^T\Lambda_0\mu_0 - \mu_n^T\Lambda_n\mu_n)\right)\right] \; .
\end{aligned}
\tag{2.12}
$$

2.5 Posterior distribution

The *posterior distribution* in the GLM-NG can be evaluated using Bayes' theorem:

$$
p(\beta,\tau|y) = \frac{p(y|\beta,\tau)\,p(\beta,\tau)}{p(y)} \; . \tag{2.13}
$$

Since $p(y)$ is just a normalization factor, the posterior is proportional to the joint:

$$
p(\beta,\tau|y) \propto p(y|\beta,\tau)\,p(\beta,\tau) = p(y,\beta,\tau) \; . \tag{2.14}
$$

From the term in (2.12), we can isolate the posterior distribution over β:

$$
\begin{aligned}
p(\beta|\tau,y) &= \mathrm{N}(\beta;\mu_n,(\tau\Lambda_n)^{-1}) \\
\mu_n &= \Lambda_n^{-1}(X^TPy + \Lambda_0\mu_0) \\
\Lambda_n &= X^TPX + \Lambda_0 \; .
\end{aligned}
\tag{2.15}
$$

From the remaining term, we can isolate the posterior distribution over τ:

$$
\begin{aligned}
p(\tau|y) &= \mathrm{Gam}(\tau;a_n,b_n) \\
a_n &= a_0 + \frac{n}{2} \\
b_n &= b_0 + \frac{1}{2}(y^TPy + \mu_0^T\Lambda_0\mu_0 - \mu_n^T\Lambda_n\mu_n) \; .
\end{aligned}
\tag{2.16}
$$

2.6 Log model evidence

According to the law of marginal probability, the *model evidence* of the GLM-NG is:

$$
p(y|m) = \iint p(y|\beta,\tau)\,p(\beta|\tau)\,p(\tau)\,\mathrm{d}\beta\,\mathrm{d}\tau \; . \tag{2.17}
$$

According to the law of conditional probability, the integrand is equivalent to the joint:

$$
p(y|m) = \iint p(y,\beta,\tau)\,\mathrm{d}\beta\,\mathrm{d}\tau \; . \tag{2.18}
$$

In (2.12), we have already evaluated this term as

$$\begin{split} p(y, \beta, \tau) = \sqrt{\frac{\tau^n |P|}{(2\pi)^n}} \sqrt{\frac{\tau^p |\Lambda_0|}{(2\pi)^p}} \, \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0-1} \exp[-b_0 \tau] \cdot \\ \exp\left[-\frac{\tau}{2} \left((\beta - \mu_n)^T \Lambda_n (\beta - \mu_n) + (y^T P y + \mu_0^T \Lambda_0 \mu_0 - \mu_n^T \Lambda_n \mu_n) \right) \right] \; . \end{split} \tag{2.19}$$

Using the posterior distribution over β, we can rewrite this as

$$\begin{split} p(y, \beta, \tau) = \sqrt{\frac{\tau^n |P|}{(2\pi)^n}} \sqrt{\frac{\tau^p |\Lambda_0|}{(2\pi)^p}} \sqrt{\frac{(2\pi)^p}{\tau^p |\Lambda_n|}} \, \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0-1} \exp[-b_0 \tau] \cdot \\ \mathrm{N}(\beta; \mu_n, (\tau \Lambda_n)^{-1}) \, \exp\left[-\frac{\tau}{2} (y^T P y + \mu_0^T \Lambda_0 \mu_0 - \mu_n^T \Lambda_n \mu_n) \right] \; . \end{split} \tag{2.20}$$

Now, β can be integrated out easily:

$$\begin{split} \int p(y, \beta, \tau) \, \mathrm{d}\beta = \sqrt{\frac{\tau^n |P|}{(2\pi)^n}} \sqrt{\frac{|\Lambda_0|}{|\Lambda_n|}} \, \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \tau^{a_0-1} \exp[-b_0 \tau] \cdot \\ \exp\left[-\frac{\tau}{2} (y^T P y + \mu_0^T \Lambda_0 \mu_0 - \mu_n^T \Lambda_n \mu_n) \right] \; . \end{split} \tag{2.21}$$

Using the posterior distribution over τ, we can rewrite this as

$$\int p(y, \beta, \tau) \, \mathrm{d}\beta = \sqrt{\frac{|P|}{(2\pi)^n}} \sqrt{\frac{|\Lambda_0|}{|\Lambda_n|}} \, \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \frac{\Gamma(a_n)}{{b_n}^{a_n}} \, \mathrm{Gam}(\tau; a_n, b_n) \; . \tag{2.22}$$

Finally, τ can also be integrated out:

$$\iint p(y, \beta, \tau) \, \mathrm{d}\beta \, \mathrm{d}\tau = \sqrt{\frac{|P|}{(2\pi)^n}} \sqrt{\frac{|\Lambda_0|}{|\Lambda_n|}} \, \frac{\Gamma(a_n)}{\Gamma(a_0)} \, \frac{{b_0}^{a_0}}{{b_n}^{a_n}} = p(y|m) \; . \tag{2.23}$$

Thus, the *log model evidence* of the GLM-NG is given by

$$\begin{split} \log p(y|m) = \frac{1}{2} \log |P| - \frac{n}{2} \log(2\pi) + \frac{1}{2} \log |\Lambda_0| - \frac{1}{2} \log |\Lambda_n| + \\ \log \Gamma(a_n) - \log \Gamma(a_0) + a_0 \log b_0 - a_n \log b_n \; . \end{split} \tag{2.24}$$

2.7 Cross-validated LME

For calculation of the *cross-validated log model evidence* (cvLME), the data are splitted into S subsets. In the training phase, all except one subset of the data are analyzed using a non-informative prior $p_{\mathrm{ni}}(\beta, \tau)$ with the prior parameters

$$\mu_0 = 0_p, \; \Lambda_0 = 0_{pp} \quad \text{and} \quad a_0 = 0, \; b_0 = 0 \tag{2.25}$$

to obtain an informative posterior $p(\beta, \tau | \cup_{j \neq i} y_j)$ using equations (2.15) and (2.16). In the testing phase, this informative posterior is then applied as a prior distribution to obtain the out-of-sample log model evidence $\log p(y_i | \cup_{j \neq i} y_j)$ via equation (2.24). Summing up over data subsets yields the cvLME according to equation (1.2).
As one can see from equations (2.15) and (2.16), the priors in (2.25) are non-informative in the sense that only the data remain to influence the posteriors.

2.8 Special cases

The *univariate Gaussian with unknown variance* (UGuv) is a special case in which

$$X = 1_n, \quad \beta = \mu \quad \text{and} \quad P = I_n \; . \tag{2.26}$$

Furthermore, *simple linear regression* (SLR) is a special case of the GLM-NG where

$$X = [1_n, x], \quad \beta = [\beta_0, \beta_1]^T \quad \text{and} \quad V = I_n \; . \tag{2.27}$$

The *one-sample t-test*, the *two-sample t-test*, the *paired t-test* and the *omnibus F-test* can all be emulated as comparisons of general linear models with specific design matrices.

2.9 Implementation

In **MATLAB**, maximum likelihood estimates and Bayesian posterior distributions can be obtained via the functions `GLM_MLE` and `GLM_Bayes` while log model evidence and cross-validated LME can be calculated using the functions `GLM_LME` and `GLM_cvLME`. Given an $n \times v$ data matrix Y, an $n \times p$ design matrix X, an $n \times n$ precision matrix P and a number of data subsets S, the cvLME for a GLM-NG is calculated as

```
cvLME = GLM_cvLME(Y, X, P, S);
```
(2.28)

In **Python**, a GLM object has to be initiated via `glm = cvBMS.GLM(Y, X, V)` and maximum likelihood estimates, Bayesian posterior distributions, log model evidence and cross-validated LME are evaluated via `glm.MLE`, `glm.Bayes`, `glm.LME`, and `glm.cvLME`. Given Y, X, V and S as above, the cvLME for a GLM-NG is calculated as

```
cvLME = cvBMS.GLM(Y, X, V).cvLME(S);
```
(2.29)

In all of the above, V and P default to I_n whereas S defaults to 2 when left empty.

3 Poisson Distribution with Exposures

3.1 Likelihood function

Let $y = \{y_1, \ldots, y_n\}$ with $y_i \in \mathbb{N}$ be a series of observed *counts* and let $x = \{x_1, \ldots, x_n\}$ with $x_i \in \mathbb{R}$ be a series of concurrent *exposures*, some quantity that might or might not influence the measured counts. Then, according to a relatively simple model, each observation (y) would be Poisson-distributed with the Poisson rate being a product of the concurrent exposure (x) and some unknown constant (λ):

$$p(y_i|\lambda) = \mathrm{Poiss}(y_i; \lambda x_i) = \frac{(\lambda x_i)^{y_i} \cdot \exp[-\lambda x_i]}{y_i!} \; . \tag{3.1}$$

Assuming independence between individual observations, i.e. factorization of individual likelihoods, this would imply the following *likelihood function*:

$$p(y|\lambda) = \prod_{i=1}^{n} p(y_i|\lambda) = \prod_{i=1}^{n} \mathrm{Poiss}(y_i; \lambda x_i) = \prod_{i=1}^{n} \frac{(\lambda x_i)^{y_i} \cdot \exp[-\lambda x_i]}{y_i!} \; . \tag{3.2}$$

3.2 Maximum likelihood

Classical model estimation proceeds by maximizing the *log-likelihood* (LL)

$$\mathrm{LL}(\lambda) = \log p(y|\lambda) = \sum_{i=1}^{n} \left[y_i \log(\lambda x_i) - \lambda x_i - \log \Gamma(y_i + 1) \right] \tag{3.3}$$

which gives rise to *maximum-likelihood* (ML) parameter estimates

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i} = \frac{n\bar{y}}{n\bar{x}} = \frac{\bar{y}}{\bar{x}} \; . \tag{3.4}$$

3.3 Prior distribution

A conjugate prior distribution relative to the likelihood function given by (3.2) is the *gamma distribution* over the Poisson rate λ which is given by

$$p(\lambda) = \mathrm{Gam}(\lambda; a_0, b_0) = \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \lambda^{a_0 - 1} \exp[-b_0 \lambda] \tag{3.5}$$

where a_0 and b_0 are the prior shape and rate parameters for λ.

3.4 Joint likelihood

Combining the likelihood function (3.2) with the prior distribution (3.5), the *joint likelihood function* of the Poisson distribution with exposures (Poiss-exp) becomes

$$\begin{aligned} p(y, \lambda) &= p(y|\lambda) \, p(\lambda) \\ &= \prod_{i=1}^{n} \left(\frac{(\lambda x_i)^{y_i} \, \exp[-\lambda x_i]}{y_i!} \right) \cdot \frac{{b_0}^{a_0}}{\Gamma(a_0)} \, \lambda^{a_0 - 1} \exp[-b_0 \lambda] \; . \end{aligned} \tag{3.6}$$

Multiplying out the product gives:

$$p(y, \lambda) = \prod_{i=1}^{n} \left(\frac{x_i^{y_i}}{y_i!} \right) \lambda^{n\bar{y}} \exp[-n\bar{x}\lambda] \cdot \frac{{b_0}^{a_0}}{\Gamma(a_0)} \lambda^{a_0-1} \exp[-b_0\lambda] \; . \tag{3.7}$$

Collecting identical variables gives:

$$p(y, \lambda) = \prod_{i=1}^{n} \left(\frac{x_i^{y_i}}{y_i!} \right) \frac{{b_0}^{a_0}}{\Gamma(a_0)} \lambda^{a_0+n\bar{y}-1} \exp[-(b_0 + n\bar{x})\lambda] \; . \tag{3.8}$$

3.5 Posterior distribution

The *posterior distribution* of the Poisson can be evaluated using Bayes' theorem:

$$p(\lambda|y) = \frac{p(y|\lambda)\, p(\lambda)}{p(y)} \; . \tag{3.9}$$

Since $p(y)$ is just a normalization factor, the posterior is proportional to the joint:

$$p(\lambda|y) \propto p(y|\lambda)\, p(\lambda) = p(y, \lambda) \; . \tag{3.10}$$

From the term in (3.8), we can isolate the posterior distribution over λ:

$$\begin{aligned} p(\lambda|y) &= \mathrm{Gam}(\lambda; a_n, b_n) \\ a_n &= a_0 + n\bar{y} \\ b_n &= b_0 + n\bar{x} \; . \end{aligned} \tag{3.11}$$

Note that $\bar{y}$ and $\bar{x}$ are the averages of y and x and therefore $n\bar{y}$ and $n\bar{x}$ are the sums of all elements in y and x, respectively.

3.6 Log model evidence

According to the law of marginal probability, the *model evidence* of the Poisson is:

$$p(y|m) = \int p(y|\lambda)\, p(\lambda) \, \mathrm{d}\lambda \; . \tag{3.12}$$

According to the law of conditional probability, the integrand is equivalent to the joint:

$$p(y|m) = \int p(y, \lambda) \, \mathrm{d}\lambda \; . \tag{3.13}$$

In (3.8), we have already evaluated this term as

$$p(y, \lambda) = \prod_{i=1}^{n} \left(\frac{x_i^{y_i}}{y_i!} \right) \frac{{b_0}^{a_0}}{\Gamma(a_0)} \lambda^{a_0+n\bar{y}-1} \exp[-(b_0 + n\bar{x})\lambda] \; . \tag{3.14}$$

Using the posterior distribution over λ, we can rewrite this as

$$p(y,\lambda) = \prod_{i=1}^{n} \left(\frac{x_i^{y_i}}{y_i!} \right) \frac{{b_0}^{a_0}}{\Gamma(a_0)} \frac{\Gamma(a_n)}{{b_n}^{a_n}} \, \mathrm{Gam}(\lambda; a_n, b_n) \; . \tag{3.15}$$

Now, λ can be integrated out easily:

$$\int p(y,\lambda) \, \mathrm{d}\lambda = \prod_{i=1}^{n} \left(\frac{x_i^{y_i}}{y_i!} \right) \frac{{b_0}^{a_0}}{\Gamma(a_0)} = p(y|m) \; . \tag{3.16}$$

Thus, the *log model evidence* of the Poisson is given by

$$\begin{split} \log p(y|m) = & \sum_{i=1}^{n} y_i \log(x_i) - \sum_{i=1}^{n} \log \Gamma(y_i+1) + \\ & \log \Gamma(a_n) - \log \Gamma(a_0) + a_0 \log b_0 - a_n \log b_n \; . \end{split} \tag{3.17}$$

3.7 Cross-validated LME

For calculation of the *cross-validated log model evidence* (cvLME), the data are splitted into S subsets. In the training phase, all except one subset of the data are analyzed using a non-informative prior $p_{\mathrm{ni}}(\lambda)$ with the prior parameters

$$a_0 = 0 \quad \text{and} \quad b_0 = 0 \tag{3.18}$$

to obtain an informative posterior $p(\lambda| \cup_{j \neq i} y_j)$ using equation (3.11). In the testing phase, this informative posterior is then applied as a prior distribution to obtain the out-of-sample log model evidence $\log p(y_i| \cup_{j \neq i} y_j)$ via equation (3.17). Summing up over data subsets yields the cvLME according to equation (1.2).
As one can see from equation (3.11), the priors in (3.18) are non-informative in the sense that only the data remain to influence the posteriors.

3.8 Special cases

The *Poisson distribution without exposures* (Poiss) is a special case in which

$$x = 1_n \; , \tag{3.19}$$

i.e. the exposures x are constant and one, such that $\bar{x} = 1$ and $n\bar{x} = n$.

3.9 Implementation

In **MATLAB**, maximum likelihood estimates and Bayesian posterior distributions can be obtained via the functions `Poiss_MLE` and `Poiss_Bayes` while log model evidence and cross-validated LME can be calculated using the functions `Poiss_LME` and `Poiss_cvLME`. Given an $n \times v$ data matrix Y, an $n \times 1$ design vector x and a number of data subsets S, the cvLME for the Poisson is calculated as

```
cvLME = Poiss_cvLME(Y, x, S);
```

(3.20)

In **Python**, a Poisson object has to be initiated via `poiss = cvBMS.Poiss(Y, x)` and maximum likelihood estimates, Bayesian posterior distributions, log model evidence and cross-validated LME are evaluated via `poiss.MLE`, `poiss.Bayes`, `poiss.LME`, and `poiss.cvLME`. Given Y, x and S as above, the cvLME for the Poisson is calculated as

```
cvLME = cvBMS.Poiss(Y, x).cvLME(S);
```

(3.21)

In all of the above, x defaults to 1_n whereas S defaults to 2 when left empty.