Journal of Machine Learning Research 13 (2012) 1699-1703 Submitted 11/10; Revised 10/11; Published 5/12

glm-ie: Generalised Linear Models Inference & Estimation Toolbox

Hannes Nickisch HANNES@NICKISCH.ORG
Max Planck Institute for Biological Cybernetics
Spemannstraße 38
72076 Tübingen, Germany

Editor: Mikio Braun

Abstract

The glm-ie toolbox contains functionality for estimation and inference in generalised linear models over continuous-valued variables. Besides a variety of penalised least squares solvers for estimation, it offers inference based on (convex) variational bounds, on expectation propagation and on factorial mean field. Scalable and efficient inference in fully-connected undirected graphical models or Markov random fields with Gaussian and non-Gaussian potentials is achieved by casting all the computations as matrix vector multiplications. We provide a wide choice of penalty functions for estimation, potential functions for inference and matrix classes with lazy evaluation for convenient modelling. We designed the glm-ie package to be simple, generic and easily expansible. Most of the code is written in Matlab including some MEX files to be fully compatible to both Matlab 7.x and GNU Octave 3.3.x. Large scale probabilistic classification as well as sparse linear modelling can be performed in a common algorithmical framework by the glm-ie toolkit.

Keywords: sparse linear models, generalised linear models, Bayesian inference, approximate inference, probabilistic regression and classification, penalised least squares estimation, lazy evaluation matrix class

1. Introduction

Generalised Linear Models (GLMs) are a widely used class of probabilistic graphical models over continuous variables allowing a unified treatment of linear, logistic and Poisson regression and applications range from simple binary pattern classification over continuous regression to image reconstruction. GLMs combine the computational and analytical simplicity of linear functions with the expressivity of pointwise nonlinear link functions.

Estimation of the unknown parameters by maximum likelihood and penalised variants thereof can be done by iteratively reweighted least squares (IRLS). Penalised least squares (PLS) corresponds to maximum a posteriori estimation (MAP) in a Bayesian model.

(Approximate) Bayesian inference as opposed to MAP places the parameter estimate at the centre of mass rather than at the mode of the posterior distribution.

- We support Expectation Propagation (EP) or TAP (Minka, 2001; Opper and Winther, 2001) using parallel moment matching (van Gerven et al., 2010),
- Variational Bounding (VB) (Seeger and Nickisch, 2011) based on decoupling (Wipf and Nagarajan, 2008) and (convex) (Nickisch and Seeger, 2009) optimisation, and
- Mean field (MF) (Miskin and MacKay, 2000) factorial inference.

While EP yields very accurate approximations for small models, VB allows for efficient computations in large-scale models by a sequence of variance-smoothed PLS problems, which makes experimental design for imaging applications (Seeger et al., 2010) possible.

Related codes are the GPML library and the Infer.NET framework both of which do not offer scalable matrix vector multiplication (MVM) based approximate inference.

2. Implementation and Model Class

The `glm-ie` toolbox can be obtained from `http://mloss.org/software/view/269/` and from `http://hannes.nickisch.org/code/glm-ie/` under the FreeBSD license. Based on simple interfaces for *potential*, *penalty*, and *estimation* functions as well as *inference* methods and *matrix* classes, we offer full compatibility to Matlab 7.x[1] and GNU Octave 3.3.x[2].

We provide modular, extensible and tested code. The algorithms rely on PLS estimations based on MVMs in turn. Our documentation comes in two parts: (i) a hypertext document[3] `doc/index.html` with detailed examples and (ii) a technical documentation[4] `doc/manual.pdf` explaining the interfaces to allow inclusion of new functionality.

The `glm-ie` toolbox deals with inference and estimation in GLMs of unknown hidden parameters $\mathbf{u} \in \mathbb{R}^n$, Gaussian observations $\mathbf{y} \in \mathbb{R}^m$ and non-Gaussian potentials $\mathcal{T}_j(s_j)$

$$\mathbf{y} = \mathbf{X}\mathbf{u} + \varepsilon,\ \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \quad \mathbf{s} = \mathbf{B}\mathbf{u} \in \mathbb{R}^q,$$

leading to a posterior of the form

$$\mathbb{P}(\mathbf{u}|\mathcal{D}) = \mathbb{P}(\mathbf{u}|\mathbf{X},\mathbf{y},\sigma) = \frac{1}{Z}\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{u},\sigma^2\mathbf{I})\prod_{j=1}^{q}\mathcal{T}_j(s_j), \quad Z = \int \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{u},\sigma^2\mathbf{I})\prod_{j=1}^{q}\mathcal{T}_j(s_j)\mathrm{d}\mathbf{u}. \tag{1}$$

A MAP estimate $\hat{\mathbf{u}}_{\mathrm{MAP}} \in \mathbb{R}^n$ is the parameter value with highest posterior density; finding $\hat{\mathbf{u}}_{\mathrm{MAP}}$ is equivalent to solving the PLS problem

$$\hat{\mathbf{u}}_{\mathrm{MAP}} = \arg\max_{\mathbf{u}} \mathbb{P}(\mathbf{u}|\mathcal{D}) = \arg\min_{\mathbf{u}} \|\mathbf{X}\mathbf{u}-\mathbf{y}\|^2 + 2\lambda\cdot\rho(\mathbf{s}),\ \mathbf{s} = \mathbf{B}\mathbf{u},\ \lambda \in \mathbb{R}_+ \tag{2}$$

with *penalty function* $\rho(\mathbf{s}) = -\sum_{j=1}^{q}\ln\mathcal{T}_j(s_j)$ derived from the *potential function* $\mathcal{T}(\mathbf{s})$ and weight $\lambda = \sigma^2$. The normalisation constant Z is called the model evidence or equivalently the marginal likelihood and can be used to compare models and adjust free parameters such as σ.

Our approximate inference algorithms replace the non-Gaussian potentials $\mathcal{T}_j(s_j)$ by Gaussians $\mathcal{N}(s_j|\beta_j\gamma_j,\gamma_j) \propto \exp(-s_j^2/(2\gamma_j)+\beta_j s_j)$ resulting in an overall Gaussian approximation $\mathbb{P}(\mathbf{u}|\mathcal{D}) \approx \mathbb{Q}(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m},\mathbf{V})$, where $\mathbf{V}^{-1} = \mathbf{A} = \mathbf{X}^\top\mathbf{X}/\sigma^2 + \mathbf{B}^\top\boldsymbol{\Gamma}^{-1}\mathbf{B}$ and $\mathbf{m} = \mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{y}/\sigma^2 + \mathbf{B}^\top\boldsymbol{\beta})$ correspond to the mean and (co)variance and $\boldsymbol{\Gamma} := \mathrm{dg}(\gamma_1,..,\gamma_q)$.

Overall, a GLM can be specified by three kinds of objects: (i) *potentials* $\mathcal{T}(s)$ and *penalties* $\rho(s)$, (ii) *matrices* $\mathbf{X}$, $\mathbf{B}$ and (iii) *PLS algorithms*. Together with the responses $\mathbf{y}$, scalar parameters and optimisation options, these three constituents serve as inputs to the double loop inference engine `dli` computing approximations to $\ln Z$, $\mathbf{m}$ and $\mathbf{V}$.

1. Matlab is available from MathWorks, `http://www.mathworks.com/`.
2. Octave is available from the Free Software Foundation, `http://www.gnu.org/software/octave/`.
3. Documentation can be found at `http://people.kyb.tuebingen.mpg.de/hn/glm-ie/doc/index.html`.
4. Technical docs are available at `http://people.kyb.tuebingen.mpg.de/hn/glm-ie/manual.pdf`.

2.1 Potential and Penalty Functions

Several *non-Gaussian potentials* $\mathcal{T}(s)$[5] can be used to shape the posterior distribution of Equation (1). In addition to the Gaussian potential `potGauss`, we provide several sparse potentials (exponential power `potExpPow`, Laplace `potLaplace`, Sech-squared[6] `potSech2`, Student's t `potT`) and the logistic potential `potLogistic` for binary classification.

In MAP estimation, we most naturally use the *penalty function* $\rho(s) = -\ln \mathcal{T}(s)$ in Equation (2) which includes penalty functions like p-norms `penAbs`, `penQuad`, `penPow`. Approximate inference by variational bounding requires *penalty functions* derived from a *potential function*, for example, `penVB` or `penVBNorm`.

2.2 General Matrix Class and Implementations

To facilitate the specification and composition of system matrices $\mathbf{X}$, $\mathbf{B}$ (and their respective transposes) in a GLM, the `glm-ie` toolbox contains a specialised matrix class `mat`. As MVMs form the most important computations in both estimation and inference, the class `mat` provides addition, transposition, scaling, composition, and concatenation. Thus, expressions like `A+B'`, `A*B`, `A*x`, `a*A`, `[A,B]`, `[A;B]`, `A(:,1)`, `kron(A,B)`, `repmat(A,[2,3])` are possible even though `A` or `B` are of type `mat` and have a size that would not fit into memory if stored as a dense array. Combinations are also possible. We provide several matrix classes that are derived from `mat` and implement their own MVM. Besides 2d convolution `matConv2`, diagonal `matDiag`, finite difference `matFD2` and (quadrature mirror) wavelet matrices `matWav`, we offer three kinds of Fourier matrices `matFFT2line`, `matFFTNmask` and `matFFT2nu` allowing for nonuniform spacing. Computations only take place through MVMs; the other operations (addition, composition etc.) are pure bookkeeping.

2.3 Penalised Least Squares Solvers

The `glm-ie` toolbox contains several solvers for the PLS estimation problem of Equation (2):

- `plsCG`: Conjugate Gradients (CG) using a standalone solver (Rasmussen, 2006),
- `plsCGBT`: CG with an Armijo backtracking rule (Lustig et al., 2007),
- `plsTN`: Truncated Newton or IRLS (Seeger et al., 2009),
- `plsLBFGS`: uses a wrapper for the famous LBFGSB code (L-BFGS-B, 1997),
- `plsBB`: first order two-point step size rule (Barzilai and Borwein, 1988), and
- `plsSB`: Bregman Splitting (Goldstein and Osher, 2009).

The solvers can be used for a standalone estimation task or—together with the `penVB` penalty function—as the inner loop of the double loop variational inference algorithm.

5. We use an additional scale parameter τ, that is, the rescaled potential $\mathcal{T}(\tau s)$.
6. The sech-square distribution is another name for the logistic distribution. We use sech-square to avoid a name clash with the logistic classification potential.

3. Example and Code

To illustrate the modular structure of the `glm-ie` toolbox, we provide a simple code example[7]. We use a sparse linear model to describe Fourier measurements $\mathbf{y} = \mathbf{X}\mathbf{u} + \boldsymbol{\varepsilon} \in \mathbb{C}^m$ of an unknown pixel image $\mathbf{u} \in \mathbb{R}^n$ where the design matrix $\mathbf{X}$ contains a subset of the rows of the Fourier matrix $\mathbf{F}$, that is, $\mathbf{X} = \mathbf{M}\mathbf{F}$ as specified by a measurement masking matrix $\mathbf{M} \in \{0,1\}^{m \times n}$. As a prior, we employ the knowledge that the filter responses $\mathbf{s} = \mathbf{B}\mathbf{u}$ of natural images with zero mean filters $\mathbf{b}_j$, $j{=}1..q$ follow a sparse distribution imitated by the Laplace potential $\mathcal{T}_j(s_j) = e^{-\tau|s_j|}$. More specifically, the filter matrix $\mathbf{B}$ contains multiscale derivatives; it is a concatenation of finite differences in both image directions and wavelet coefficients. This model allows to reconstruct images from undersampled magnetic resonance imaging scanner measurements, where $m < n$.

These steps are illustrated below, and following the code, we discuss in some detail, the meaning and role of each line, and mention in passing some of the alternative possibilities. Note that, full specification and prediction can be done in as little as nine lines of code:

```
1 [y,mask] = read_data; su = size(mask);          % load measurements y and mask
2 X    = matFFTNmask(mask);                    % construct a partial Fourier matrix
3 s2   = 1e-5;                               % define the observation noise variance
4 B    = [matWav(su); matFD2(su)];  % conc. wavelet and finite difference matrix
5 pen  = @(s) penAbs(s);                 % define l1-norm penalty function (LASSO)
6 [u,phi]  = plsLBFGS(u0,X,y,B,opt,s2,pen);             % perform PLS estimation
7 pot = @potLaplace;       % define a Laplace potential (corresponds to l1-norm)
8 tau = 15;                                % declare width of the Laplace potential
9 [m,ga,be,z,zu,nlZ] = dli(X,y,s2,B,pot,tau,opts);   % double loop VB inference
```

Data $\mathbf{y}$ is loaded in **line 1**. The observation noise variance σ^2 and the design matrix $\mathbf{X}$ as declared in **lines 2-3** form the Gaussian part. The filter matrix $\mathbf{B}$ for the non-Gaussian part is constructed in **line 4** as the concatenation of two matrices. More involved compositions such as `[W; a*D]` are possible. In the next two lines, we perform estimation by optimising equation (2) using the `plsLBFGS` solver in **line 6** for the penalty function $\rho(\mathbf{s}) = \sum_j |s_j|$ as declared in **line 5**. Variational inference requires a potential $\mathcal{T}(\tau\mathbf{s})$; here $\mathcal{T}(\tau s) = \exp(-\tau|s|)$, with scale τ, is defined in **lines 7-8**. The engine for double loop inference `dli` is finally called in **line 9** yielding the posterior mean estimate $\mathbf{m}$, the variational parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$, the marginal variances $\mathbf{z} = \mathrm{var}(\mathbf{B}\mathbf{u})$, $\mathbf{z}_\mathbf{u} = \mathrm{var}(\mathbf{u})$ and the negative log evidence $-\ln Z \equiv$ `nlZ` of the model.

Acknowledgments

Thanks go to George Papandreou for contributing his marginal variance estimator (Papandreou and Yuille, 2011), to Matthias Seeger for helpful suggestions, Michael Hirsch for testing and contributing the convolution code and Wolf Blecher for testing.

References

Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

7. The example is a shortened version of one the detailed application examples, which are part of the documentation of the `glm-ie` package.

Tom Goldstein and Stanley Osher. The split Bregman method for l1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. URL ftp://ftp.math.ucla.edu/pub/camreport/cam08-29.pdf.

Michael Lustig, David L. Donoho, and John M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Tom Minka. Expectation propagation for approximate Bayesian inference. In *UAI*, 2001.

James Miskin and David J.C. MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in Independent Component Analysis*, 2000.

Hannes Nickisch and Matthias Seeger. Convex variational Bayesian inference for large scale generalized linear models. In *ICML*, 2009.

Manfred Opper and Ole Winther. Adaptive and selfaveraging Thouless-Anderson-Palmer mean field theory for probabilistic modeling. *Physical Review E*, 64:056131, 2001.

George Papandreou and Alan Yuille. Efficient variational inference in large-scale Bayesian compressed sensing. In *Proc. IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition (in conjunction with ICCV)*, 2011.

Carl E. Rasmussen. minimize.m, September 2006. URL http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/.

Matthias W. Seeger and Hannes Nickisch. Large scale variational inference and experimental design for sparse generalized linear models. *SIAM Journal on Imaging Sciences*, 4(1):166–199, 2011.

Matthias W. Seeger, Hannes Nickisch, Rolf Pohmann, and Bernhard Schölkopf. Bayesian experimental design of magnetic resonance imaging sequences. In *NIPS*, 2009.

Matthias W. Seeger, Hannes Nickisch, Rolf Pohmann, and Bernhard Schölkopf. Optimization of k-space trajectories for compressed sensing by bayesian experimental design. *Magnetic Resonance in Medicine*, 63(1):116–126, 2010.

L-BFGS-B. by Ciyou Zhu, Richard Byrd and Jorge Nocedal., 1997. URL http://www.eecs.northwestern.edu/~nocedal/lbfgsb.html.

Marcel van Gerven, Botond Cseke, Floris de Lange, and Tom Heskes. Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *Neuroimage*, 50:150–161, 2010.

David Wipf and Srikantan Nagarajan. A new view of automatic relevance determination. In *NIPS*, 2008.