SCGLR: An \texttt{R} package for generalized linear regression on supervised components.

Catherine Trottier & Guillaume Cornu
with Frédéric Mortier & Xavier Bry

\textit{3èmes Rencontres R}
Montpellier
25-27 juin 2014
⇒ Question: What in X may predict what in Y?

→ Approach: Dimension reduction by construction of components
Collinearities:
to avoid overfitting, search for components. Components must:
- capture enough variance in $X$,
- model and predict $y$.

Several components:
to avoid redundancy, search for uncorrelation. → constraint of construction: orthogonality.

Multiple $y$:
same components,
but each $y$ with its own regression coefficients.

Exponential family distributed
→ generalized linear regression.
PLS1: single $y$

- **First component** is a **compromise** between the direction of $X$ that best predicts $y$ and the first principal component (PC) of $X$.

$$\rightarrow \text{Criterion: } \max_{||u||^2=1} [\text{cov}(y, Xu)]$$

$$\max_{||u||^2=1} \left[ \sqrt{\text{var}(y)} \sqrt{\text{var}(Xu)} \text{corr}(y, Xu) \right]$$

$$\rightarrow \text{Program to solve: } P_1 : \max_{||u||^2=1} [< y, Xu >_W]$$

- **Further components:** $W$-orthogonality of components is ensured using the part of $X$ that is not yet used, i.e. the residuals of $X$ regressed on previous components.
PLS2: multiple $y$

- **First component** can be obtained using several equivalent programs:

\[
P_2 : \max_{\|u\|^2, \|v\|^2 = 1} [\langle Xu, Yv \rangle_W]
\]

\[
P_3 : \max_{\|u\|^2 = 1} \left[ \sum_{k=1}^{q} \langle Xu, y^k \rangle^2_W \right]
\]

$P_3$ is adapted to the case of multiple weighting:

\[
P_4 : \max_{\|u\|^2 = 1} \left[ \sum_{k=1}^{q} \langle Xu, y^k \rangle^2_{W_k} \right]
\]

\[\Rightarrow \text{Solution: eigenvector associated to largest eigenvalue of:} \]

\[A = X'\Omega X \text{ with } \Omega = \sum_{k=1}^{q} W_k y^k y'^k W_k\]

- **Further components**: idem, subject to constraint of orthogonality to previous components.
In the GLM, linear predictors are constrained to be collinear to one another:

\[ \forall k = 1, q : \eta^k = X\beta_k + T\delta_k = X\gamma_k u + T\delta_k \]

\[ u \text{ and } \gamma = (\gamma_k)_{k=1,q} \text{ estimated iterating an alternated least squares two steps sequence:} \]

(1) Given \( \gamma \), working data \( (z^k)_k \) is regressed on matrix \([\gamma \otimes X, 1_q \otimes T]\) with respect to working matrix \( W = \text{diag}[W_k]_k \)

\[ \rightarrow \text{coefficient vectors } \hat{u}, \hat{\delta} = (\hat{\delta}_k)_k \]

\[ \rightarrow \hat{u} \text{ made unit norm} \quad \rightarrow \text{updated } u \]

(2) Given \( Xu \), each working vector \( z^k \) is regressed on \([Xu, T]\) with respect to working matrix \( W_k \)

\[ \rightarrow \text{updated } \gamma_k, \delta_k \]
Step \( t \) of the FSA:

\[
\min_{\gamma, u: u'=1} \left[ \sum_k \| z[k]^{t} - X \gamma u \|_{W_k}^2 \right] \\
\iff \min_{u: u'=1} \left[ \sum_k \| z[k]^{t} - \Pi_{Xu} z[k]^{t} \|_{W_k}^2 \right] \\
\iff \max_{u: u'=1} \left[ \sum_k \| z[k]^{t} \|_{W_k}^2 \cos^2_{W_k}(z[k]^{t}, Xu) \right] \\
\]

is replaced by: \( \max_{u: u'=1} \left[ \sum_k \| z[k]^{t} \|_{W_k}^2 \cos^2_{W_k}(z[k]^{t}, Xu) \right] \|Xu\|_{W_k}^2 \]
equivalent to:

\[
\max_{u: u'=1} \left[ \sum_k \langle z[k]^{t}, Xu \rangle_{W_k}^2 \right] \\
= \text{local extended PLS2} \\
\implies \text{Solution: eigenvector associated to largest eigenvalue of:} \\
A = X' \Omega^{[t]} X \text{ with } \Omega^{[t]} = \sum_{k=1}^{q} W_k^{[t]} z[k]^{t} z'[k]^{t} W_k^{[t]} \]
• Tuning the attraction of components towards principal components:

\[ A_s = (X' WX)^s A \]

The larger the value of \( s \), the closer the components to PC’s

\[ \Rightarrow \text{if } s = +\infty, \text{ SCGLR components } = \text{ PC’s}. \]

• Choice of the number of components:

Cross-validation subsampling \( \rightarrow \) prediction error

\[ \rightarrow \text{ model selection} \]
Functions and methods

1. **Main functions**
   - `scglr()`, `scglrCrossVal()`

2. **Utility functions**
   - `multivariateFormula()`, `multivariateGlm()`, `infoCriterion()`

3. **Methods**
   - `print()`, `summary()`
   - `barplot()`, `plot()`, `pairs()`

'genus' sample dataset

1. **Samples**: 1000 plots (8 by 8 km laid on a grid)
2. **Y**: abundance of 27 common tree genera in the tropical forest
3. **X**: 40 environmental variables
First Example, known number of components

Model building

```r
> library(SCGLR)
>
> # load sample data
> data(genus)
> # get variable names from dataset
> n <- names(genus)
> ny <- n[grep("^gen", n)]  # Y <- names that begins with "gen"
> nx <- n[-grep("^gen", n)]  # X <- remaining names
> # remove "geology" and "surface" from nx
> # as surface is offset and we want to use geology as additional covariate
> nx <- nx[!nx %in% c("geology", "surface")]
> # define family
> fam <- rep("poisson", length(ny))
>
> # build multivariate formula
> # we also add "lat*lon" as computed covariate
> form <- multivariateFormula(ny, c(nx, "I(lat*lon)"), c("geology"))
```

`form` is a `Formula` object:

\[ y_1 + y_2 + \ldots \sim x_1 + x_2 + \ldots | t_1 + \ldots \]
First Example, known number of components

Model fitting

```r
> genus.scglr <- scglr(formula=form, data = genus, family=fam, K=2,
+ offset=genus$surface)
> str(genus.scglr, max.level=1)
```

List of 11

$ call : language scglr(formula = form, data = genus, family = fam, K = 2, offset = genus$surface)

$ u : 'data.frame': 46 obs. of 2 variables:

$ comp : 'data.frame': 1000 obs. of 2 variables:

$ compr : 'data.frame': 1000 obs. of 2 variables:

$ gamma : List of 27

$ beta : 'data.frame': 51 obs. of 27 variables:

$ lin.pred : 'data.frame': 1000 obs. of 27 variables:

$ xFactors : 'data.frame': 1000 obs. of 1 variable:

$ xNumeric : 'data.frame': 1000 obs. of 40 variables:

$ inertia : Named num [1:2] 0.227 0.315

..- attr(*, "names")= chr [1:2] "cr1" "cr2"

$ deviance : Named num [1:27] 2307 2790 1632 1479 1468 ...

..- attr(*, "names")= chr [1:27] "gen1" "gen2" "gen3" "gen4" ...

- attr(*, "class")= chr "SCGLR"
First Example, known number of components

Graphics outputs

```r
> barplot(genus.scglr)
> plot(genus.scglr, style="simple, threshold")
```
Second example, unknown number of components

Cross-validation

```r
> genus.cv <- scglrCrossVal(formula=form, data=genus, family=fam, K=12, + offset=genus$surface)
> mean.crit <- t(apply(genus.cv, 1, function(x) x/mean(x)))
> mean.crit <- apply(mean.crit, 2, mean)
> K.cv <- which.min(mean.crit)-1
> cat("Best number of components: ", K.cv)

Best number of components: 8
```
Second example, unknown number of components

Fitting and pairs-Plot

```r
> genus.scglr <- scglr(formula=form, data=genus, family=fam, K=K.cv,
+ offset=genus$surface)
> pairs(genus.scglr, components=c(1:3,5), style="simple,threshold")
```
Ongoing works

SCGLR 1.3 version, soon on CRAN
- with new alternate optimization algorithms: Eigen vector and Iterative Normalized Gradient
- Enhancements for plot customization

SCGLR 2 version, in progress
- Multiple explanatory theme support
- New distribution families (Negative-Binomial, Exponential, Inverse Gaussian)