mRMRe: an R package for parallelized mRMR ensemble feature selection

Nicolas De Jay[1], Simon Papillon-Cavanagh[1], Catharina Olsen[2], Gianluca Bontempi[2], and Benjamin Haibe-Kains[1]

[1]Bioinformatics and Computational Biology Laboratory, Institut de recherches cliniques de Montréal, Montreal, Quebec, Canada
[2]Machine Learning Group, Université Libre de Bruxelles, Brussels, Belgium

August 22, 2012

Contents

1 Introduction

mRMRe is an R package for parallelized mRMR ensemble feature selection.

1.1 Installation

mRMRe requires that *Rcpp* is installed. These should be installed automatically when you install *mRMRe*. Install *mRMRe* from CRAN or Bioconductor using *biocLite* function.

```
> install.packages("mRMRe")
```

Load *mRMRe* into your current workspace:

```
> library(mRMRe)
```

Load the example dataset *cgps* into your current workspace:

```
> data(cgps)
> data_cgps <- data.frame(cgps_ic50, cgps_ge)
```

1.2 Known Issues

mRMRe has only been tested on Mac OS X 10.6.8 and on Linux platforms. Due to the use of the *openMP* library, users may encounter problems when trying to install this package on Mac OS X 10.6.8. To fix this issue, it is recommended to add the *-fopenmp* to the CC, CXX and LDFLAGS flags in the *~/.R/Makevars* configuration file.

2 Measures of Associtation

2.1 Mutual Information Matrix

mRMRe offers a fully parallelized implementation to compute the Mutual Information Matrix (MIM). The object *data_cgps* should be a dataframe with samples/observations in rows and features/variables in columns. The method

supports the following column types: "numeric" ("integer" or "double"), "ordered factor" and "Surv". Mutual information (MI) between two columns is estimated using a linear approximation based on correlation such that MI is estimated as $I(x,y) = -\frac{1}{2}\ln(1-\rho(x,y)^2)$, where I and ρ respectively represent the MI and correlation coefficient between features x and y. Correlation between continuous variables can be computed using either Pearson's or Spearman's estimators, while Cramer's V and Somers' Dxy index are used for correlation between discrete variables and between continuous variables and survival data, respectively.

```
> ## Test on a dummy dataset
> library(survival)
> dd <- data.frame("surv1"=Surv(runif(100), sample(0:1, 100, replace=TRUE)),
+         "cont1"=runif(100),
+         "cat1"=factor(sample(1:5, 100, replace=TRUE), ordered=TRUE),
+         "surv2"=Surv(runif(100), sample(0:1, 100, replace=TRUE)),
+         "cont2"=runif(100),
+         "cont3"=runif(100),
+         "surv3"=Surv(runif(100),
+                 sample(0:1, 100, replace=TRUE)),
+         "cat2"=factor(sample(1:5, 100, replace=TRUE), ordered=TRUE))

> message("Dummy dataframe:")
> print(dd[1:5,1:5])

       surv1     cont1 cat1      surv2     cont2
1 0.6410303  0.9672470    3 0.5591143  0.7449125
2 0.3787347+ 0.9221511    3 0.1102520+ 0.5263561
3 0.4004970  0.2316712    2 0.8510946+ 0.4606690
4 0.5629020  0.9104971    4 0.5462024  0.8989798
5 0.3369667  0.7166005    1 0.7477541  0.3907519

> message("Resulting MIM:")
> mim <- build.mim(data=dd)
> print(mim[1:5,1:5])

            surv1        cont1         cat1        surv2        cont2
surv1         Inf 0.0020741269 7.759248e-02          NaN 8.978724e-03
cont1 0.002074127          Inf 4.178243e-04 3.377074e-03 7.408064e-03
```

```
cat1  0.077592485 0.0004178243          Inf 2.178909e-02 8.214195e-05
surv2         NaN 0.0033770744 2.178909e-02          Inf 9.507031e-06
cont2 0.008978724 0.0074080643 8.214195e-05 9.507031e-06          Inf
```

```
> ## Test on the 'cgps' dataset
> ## The variables are all of continuous type
>
> # Uses Spearman as correlation estimator
> message("MIM with Pearson estimator:")
> mim <- build.mim(data_cgps)
> print(mim[1:5,1:5])
```

```
             cgps_ic50  geneid_3310  geneid_2978  geneid_6352 geneid_2621
cgps_ic50          Inf 0.0080503821 0.0011476923 0.0017245614 0.007620051
geneid_3310 0.008050382          Inf 0.0004934719 0.0129489908 0.048101708
geneid_2978 0.001147692 0.0004934719          Inf 0.0003044936 0.007718133
geneid_6352 0.001724561 0.0129489908 0.0003044936          Inf 0.001368217
geneid_2621 0.007620051 0.0481017083 0.0077181333 0.0013682174         Inf
```

```
> # Uses Pearson as correlation estimator
> message("MIM with Spearman estimator:")
> mim <- build.mim(data_cgps, uses_ranks=FALSE)
> print(mim[1:5,1:5])
```

```
              cgps_ic50  geneid_3310  geneid_2978  geneid_6352  geneid_2621
cgps_ic50           Inf 6.961090e-03 2.409113e-03 8.431788e-05 0.0062324093
geneid_3310 6.961090e-03          Inf 6.488387e-05 5.094406e-03 0.0668067709
geneid_2978 2.409113e-03 6.488387e-05          Inf 2.678583e-03 0.0074338657
geneid_6352 8.431788e-05 5.094406e-03 2.678583e-03          Inf 0.0000941246
geneid_2621 6.232409e-03 6.680677e-02 7.433866e-03 9.412460e-05          Inf
```

2.2 Correlations

The mRMRe package offers an efficient, stratified and weighted implementation of the major correlation estimators: Cramer's V, Somers Dxy index (based on the concordance index), Pearson, Spearman correlation coefficients.

```
> # Compute c-index between feature 1 and 2
> correlate(cgps_ge[,1],cgps_ge[,2], method="cindex")
```

```
> # Compute Cramer's V
> x <- sample(c(0, 1, 2), 100, replace=TRUE)
> y <- sample(c(0, 1), 100, replace=TRUE)
> correlate(x, y, method="cramer")
> # Compute Pearson coefficient with random strata and sample weights
> # between feature 1 and 2
> strata <- sample(as.factor(c("STRATUM_1","STRATUM_2","STRATUM_3")),
+         nrow(cgps_ge), replace=TRUE)
> weights <- runif(nrow(cgps_ge))
> correlate(cgps_ge[, 1], cgps_ge[, 2], strata=strata, weights=weights,
+         method="pearson", bootstrap_count=1000)
```

3 mRMR Feature Selection

mRMRe offers a highly efficient implementation of the mRMR feature selection [2, 4]. The two crucial aspects of our implementation consists first, in parallelizing the key steps of the algorithm and second, in using a lazy procedure to compute only the part of the MIM that is required during the search for the best set of features (instead of estimating the full MIM).

3.1 Classic mRMR

Here is an example of the classic mRMR feature selection [2].

```
> mRMR.classic(data_cgps, 1, 30)
```

3.2 Ensemble mRMR

Our ensemble approach allows to create a tree-like set of solutions of non redundant mRMR solutions. The topology of the ensemble tree is user defined throught the *levels* parameter. A binary tree of depth 5 can be generated with *levels=rep(2,5)*, therefore creating 2^5 mRMR solutions.

```
> mRMR.ensemble(data_cgps, target_index=1, levels=rep.int(1, 30)) # For mRMR.cla
> mRMR.ensemble(data_cgps, target_index=1, levels=rep(2,5))
```

4 Causality Inference

The mRMRe package allows one to infer causality through the use of the Co-information lattice method [1, 3].

```
> ensemble <- mRMR.ensemble(data_cgps, target_index=1, c(10, 5, rep(1, 5)))
> compute.causality(ensemble)
> compute.causality(data=data_cgps, target_index=1, mim=NULL, solutions=ensemble
```

5 Utilities

In order to allow for full user control, mRMRe allows its users to set the number of threads it will use for computations. One should consider using the following method to avoid crowding computing clusters.

```
> set.thread.count(3)
> mim <- build.mim(data_cgps)
> head(mim)
```

6 Session Info

- R Under development (unstable) (2012-06-27 r59668), x86_64-unknown-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=C, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, grDevices, graphics, methods, splines, stats, utils
- Other packages: mRMRe 1.0.1, survival 2.36-14
- Loaded via a namespace (and not attached): tools 2.16.0

References

[1] A J Bell. The co-information lattice. In S Amari, A Cichocki, S Makino, and N Murata, editors, *Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation: ICA 2003*, 2003.

[2] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(2):185–205, April 2005.

[3] W McGill. Multivariate information transmission. *IEEE Transactions on Information Theory*, 4(4):93–111, September 1954.

[4] P. E. Meyer, C. Schretter, and Gianluca Bontempi. Information-Theoretic Feature Selection in Microarray Data Using Variable Complementarity. *Selected Topics in Signal Processing, IEEE Journal of*, 2(3):261–274, 2008.