A compariative study of variable selection methods on diverse nature of data

Compulsory Assignment (STAT 370)

Raju Rimal

May 7, 2016

Contents

Introduction

Simple and informative models is always preferred to complex model. Different variable selection techniques are compared on the basis of their performance on various nature of data. Table-1 presents an overview of the datasets used in this study. All the data are pre-processed using different techniques such as baseline correction and smoothing before they are used in this study.

Datasets	Types	Response Variables	Predictor Variables
`MALDITOF_Milk`	Spectrometric	`cow, goat, ewe`	Mass Spectra (`MS`)
`NIR_Raman_PUFA`	Spectroscopic	`PUFA_total, PUFA_fat`	`NIR` and `Raman`
`GeneExpr_Prostate`	Microarray	`tumor`	Gene Expression (`GeneExpr`)

Table 1: Overview of Datasets

MALDI-TOF (Matrix-assisted laser ionization - Time of Flight) is a mass spectrometric technique known for identifying proteins, peptides and some other ionisable compounds in samples, as explained in Liland et al. (2009) and (Fig: 1; top-left).

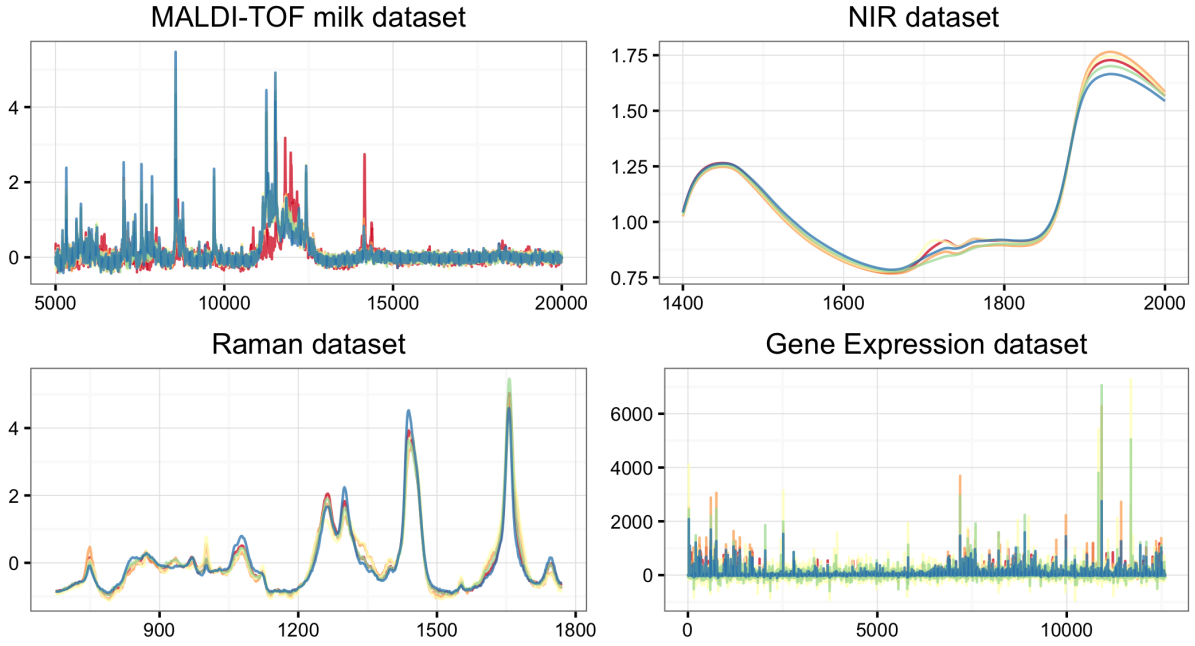

Figure 1: Sample observations from a) MALDI-TOF (top-left) b) NIR (top-right) c) Raman (bottom-left) and d) GeneExpression (bottom-right)

Near Infrared (NIR) and **Raman** are spectroscopic techniques where the frequencies of reflected light from a vibrating molecules are measured. In the case of Raman spectroscopy, measurement is made only on raman shift, it is able to detect wider range of molecules than NIR (Fig: 1; top-right and bottom-left). Here, the techniques are used to measure the percentage of polyunsaturated fatty acids in a) total sample weight b) total fat content (Næs et al., 2013). **Gene Expression** is a microarray data which contains a Gene Expression Matrix and a response vector which identify whether the sample contains tumor or not. A raw microarray data are images which are transformed into gene expression matrix that contains gene for each sample (fig-1 (bottom-right)).

Statistical Models

Principal Component Analysis (PCA)

PCA enables the underlying structure present in a dataset by transforming variables into a new set of uncorrelated variables. PCA is used to reduce the dimension of dataset consisting of correlated variables, retaining most of the variation present in it on first few PCs.

In this study, the first and second principal components are plotted, in figure - 2 (top-left), which are linear combinations of original variables. The PC of MALDI-TOF are colored according to milk proportion of cow, goat and ewe which revels clusters present in the mass-spectra at three different corners. A similar grouping is visible on gene expression data in fig-2 (top-right) where the points are colored according to there state of having tumor or not.

Further, the principal components of NIR and Raman datasets in figure - 2 (bottom) shows that higher concentration of PUFA are separated by first principal component in NIR and second principal component in Raman dataset.

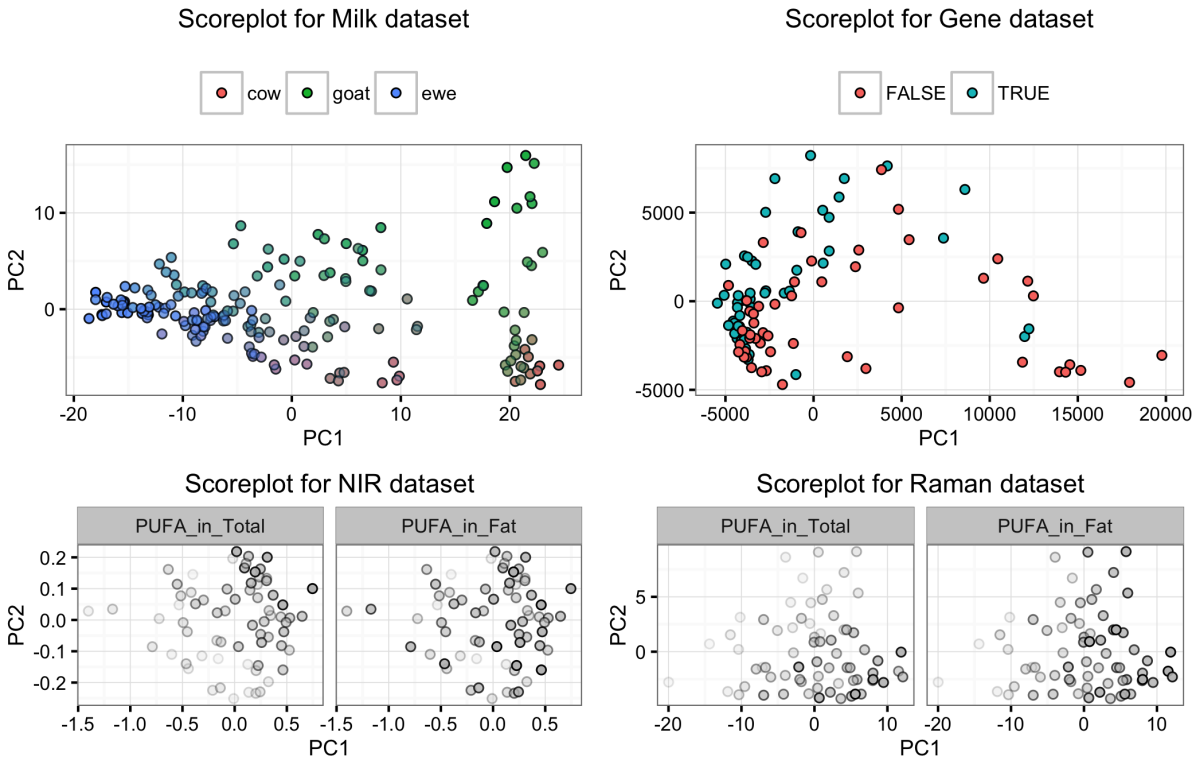

Figure 2: Principal components of (top-left) MALDI-TOF dataset with colored proportion of cow, goat and ewe milk, (top-right) Gene Expression dataset with presence and absense of tumer on the sample, (bottom) NIR and Raman dataset with opacity factor with percentage of (left) PUFA in total sample and (right) PUFA in total fat content

Calibration and Validation

Model validation ensures how the model will perform on completely new environment, i.e. with observations on included for its calibration. For *external validity*[1], datasets are splitted into training and test sets with 70 percent of observations taken as training set while rest as test set. However, due to replications, MALDI-TOF milk data is splitted into 3:2 ratio such that the replications are bind together. An

[1]Martens and Martens, 2001, ch.10

internal validation[2] is done on the training set with 10 fold random cross-validation except on MALDI-TOF milk. On MALDI-TOF data a consecutive splitting is made with 30 folds which hence ensure each set of replication creates a fold. Further, the models calibrated with cross-validation from training set are used for predicting test set.

Partial Least Square (PLS) Regression

PLS uses the latent structure for modeling the relation between matrices **X** and **Y**. Latent variables of both **X** and **Y** are modeled to explain the variance structure (direction) present in **Y**. Flawless performance on wide matrices (where variables are more than observations) and correction of multicollinearity are the advantages of Partial Least Square Regression.

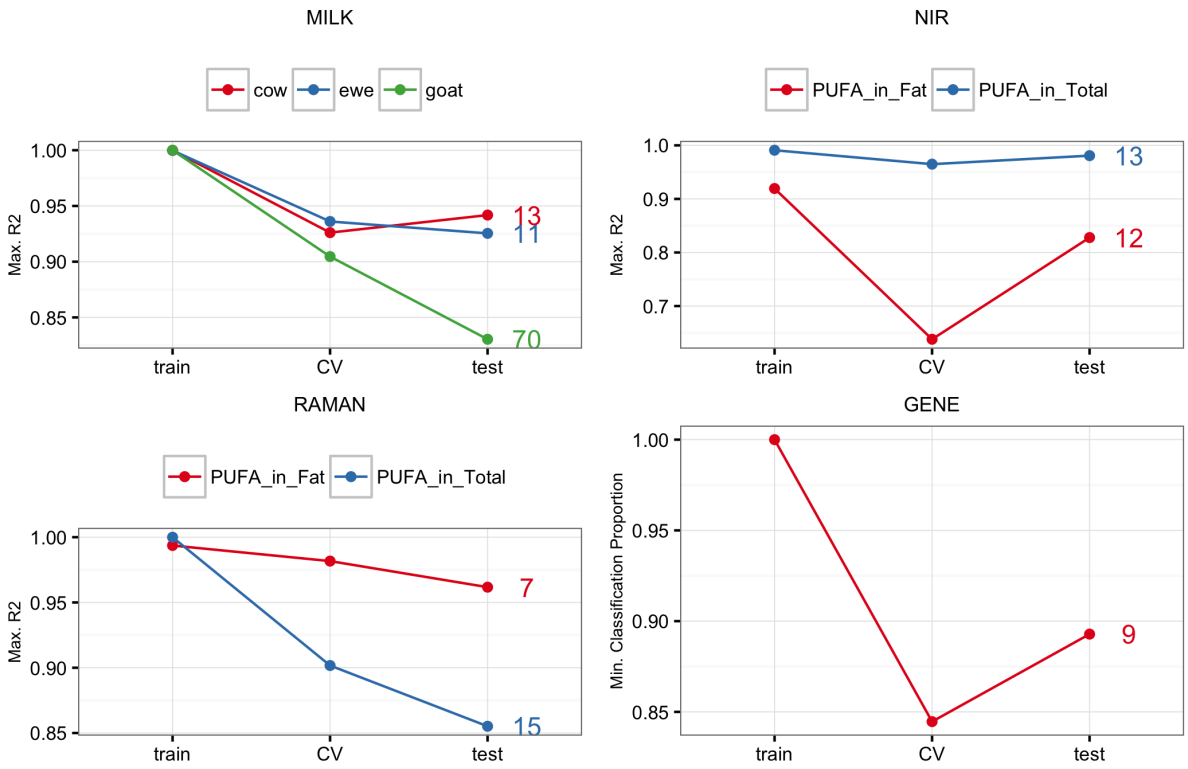

Figure 3: Maximum variation explained on response in Milk, NIR and Raman dataset and maximium correct classification proportion for gene Expression dataset. Each line represent a model for which the R2 predicted is calculated with number of components chosen by cross-validation (end of each line).

A PLS model with cross-validation is fitted on training set of all the datasets with individual responses (i.e. one response per model). A measure of maximum R^2 predicted is used as a validation tool. The number of components, needed for explaining most of the variation present in the response, obtained from cross-validation is used for test prediction. For classification of tumor or non-tumor in Gene Expression dataset, a modified PLS model is used where the scores from PLS model, with number of components needed for minimum RMSECV, is used as predictor variable in Linear Discriminant Analysis (LDA) model. A final model, with maximum average correct classification proportion, is used to predict the test set.

Mostly, the variation explained in test are smaller than cross-validation in fig - 3 while on few of them it is larger than cross-validation. Plots from spectroscopic data shows that NIR data are able to explain *PUFA present in total* better than Raman while Raman is better is explaining *PUFA in fat*. Since

[2]Martens and Martens, 2001, ch.10

number of components from cross-validation are used, the model fitted for goat response in MALDI-TOF data has 70 components and is performing poor in test data for which the presence of noise with inclusion of more components may be the reason behind. However, in all the cases, more than 65% of the variation on test is explained by the chosen PLS model.

Linear Discriminent Analysis

Assuming equal variance for each group, LDA attempts to classify / discriminate the response **y** using a log-odd function, linear on **X**, i.e, LDA decision boundaries are linear (Friedman, Hastie, and Tibshirani, 2001). Subjects with and without tumor, in Gene Expression data are classified using a LDA model from PLS scores. The number of components is obtained from a cross-validation technique, as in figure - **??**, which integrates LDA and PCA to obtain optimized number of components that minimize the misclassification error. The first Linear Discriminant score plotted for training and testset for their prediction (fig - **??** (right)) shows that the model has classified the presence of tumor with high accuracy.

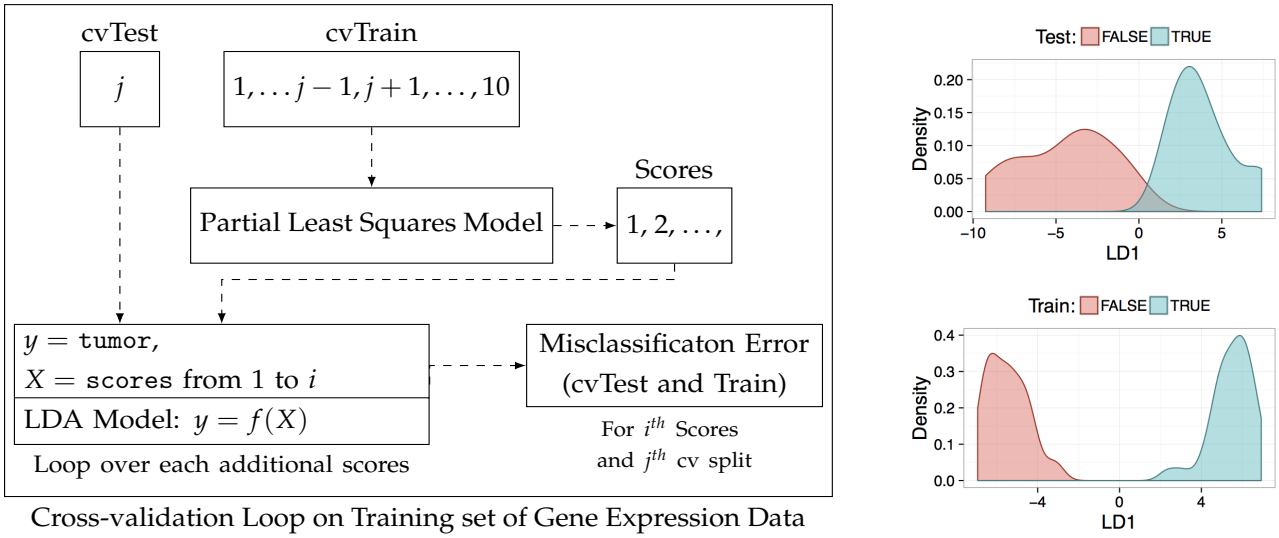

Figure 4: (left) Cross Validation with LDA integrated within PLS model for classification problem in Gene Expression Data (right) Test and Training classification

Variable Selection

Variable selection techniques as suggested in Mehmood et al. (2012) are implemented on each dataset and a comparison is made with the model fitted with complete set. Mehmood et al. (2012) have categorized the variable selection technique into three different categories: Filter, Wrapper and Embedded methods (fig. - 5).

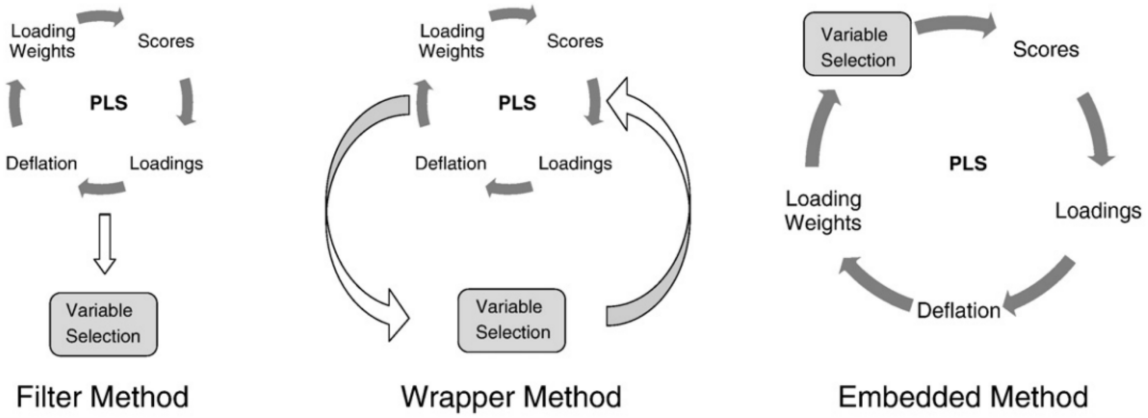

Figure 5: Illustration of Variation Selection Techniques

Filter Method

Filter method selects variables from a fitted PLS model introducing a threshold on some measure such as Loading Weights, Regression Coefficients and Variable Importance in Projection (VIP). VIP is used as a filter method in this study with a threshold value of 1. Since, only the important variables are selected and with reduced noise, in many occasions, PLS model fitted with only selected variables predicts better than the full Model. For instance, with cow milk as response in the case of MALDI-TOF model, figure - 6 shows that the model from VIP filter with only 1305 variables, has explained more than the Full Model on both CV and testset.

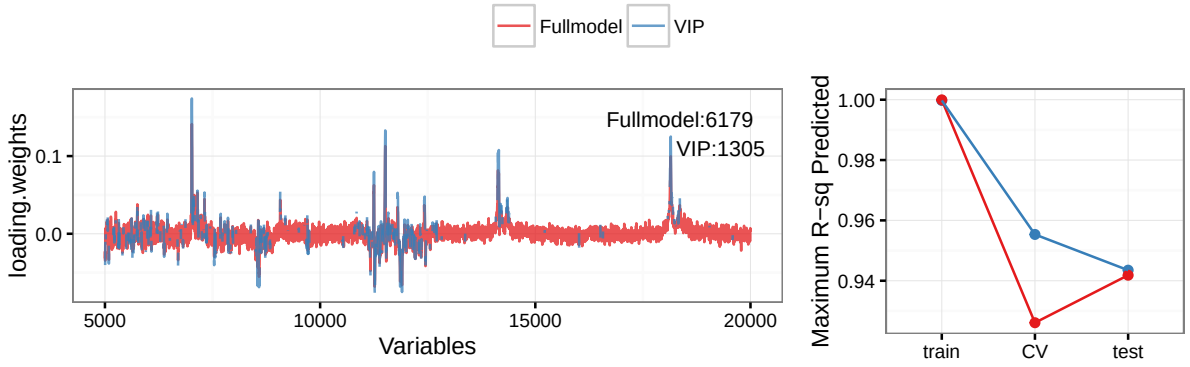

Figure 6: Comparison of Models with complete set of variables and variables obtained from VIP filter

Wrapper Method

Wrapper Method uses filter method in an iterative way. MCUVE (Monte-Carlo based Uninformative Variable Elimination) and Shaving methods are used in this study. The comparison

MCUVE method is used which splits the sample set into test and train and runs random cross-validation on training and selects variables based on final performance test on test data. Despite decrease the risk of over-fitting (Mehmood et al., 2012), MCUVE PLS adopted here has chosen very few variables resulting low R-Sq predicted in all nature of datasets.

Shaving method (*Package 'plsVarSel'*) first arrange the variables using some filter methods. A subset of least information variables are eliminated using a threshold value and a model is again fitted and model performance is measured. The procedure is repeated to achieve maximum model performance.

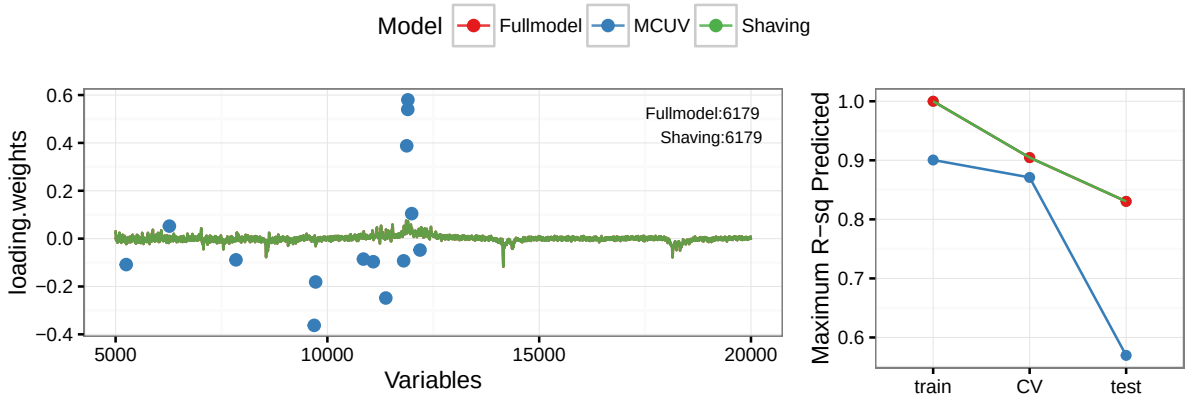

Figure 7: (left) Variables selected on MALDI-TOF Milk Spectra data where goat is a response. (right) Maximum R^2 Predicted from model with variables selected from Fullmodel, MCUV, Shaving method.

MCUV method is able to explain only 20% of variation present in goat percent in the milk mixture of MALDI-TOF data (fig - 7 (right)). Selection of few variables by the method has removed many more information which results in this poor performance. While the shaving method with less than 5000 variables has almost performed similarly as complete model with more than 6000 variables. The comparison between these methods will be performed on next section of this paper.

Embedded Method

Embedded Method integrate variable selection procedure in a single run of model fitting. Since the procedure is wrapped within PLS algorithm, it is less time consuming than wrapper method where double iteration occurs. Truncation, an embedded method, is used in this study where, loading weights are truncated around their median based on their confidence intervals. For Instance, in the case of MALDI-TOF milk data fitted for response `cow` and Gene Expression data (Figure - 8), the loading weights closer to zero with 95% confidence level around median are termed as uninformative and set to zero or truncated. For each additional components, the truncation method truncates more loading weights since the higher order components contains extra noise.

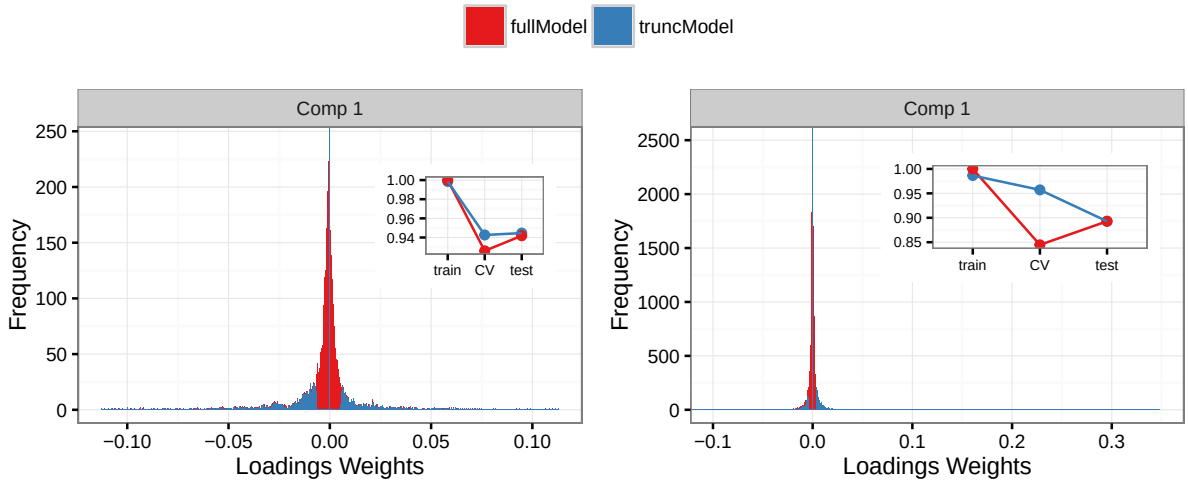

Figure 8: Histogram of Loading Weights for full and truncated model for (left) MALDI-TOF milk data with response cow (right) Gene Expression data. Corresponding inset shows the maximum R-sq predicted for train, CV and test.

Analysis and Discussion

Small degree of freedom arises difficulties in studying the interaction between dataset and variable selection methods implemented on them. So, replications of complete study are created specifying different test and training sets. Since, 30 percent of observations are sampled randomly as a test set in NIR, Raman and Gene datasets, replications 2 and 3 has also implemented the same strategy. However, in the case of MALDI-TOF milk dataset, with constrained to keep replicates together, the second and third replicates are created by keeping 120 observations as training set starting from position 41 and 61 respectively. R^2 predicted for test observation from all replicated are used for further analysis (Figure - 9).

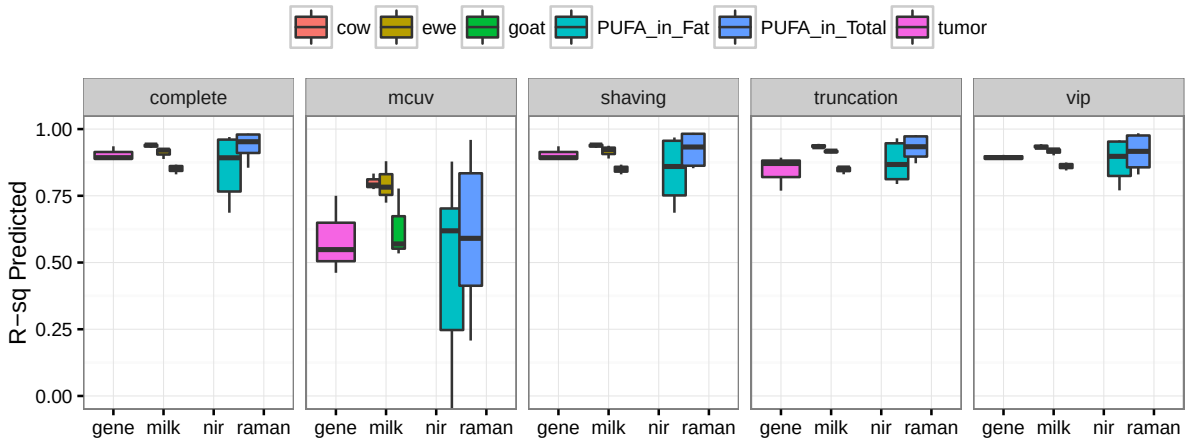

Figure 9: Boxplot for each data and method combination subdivided into their respective responses. Model from MCUVE in NIR and Raman data have drastically poor performance than other.

Since each responses are confined to their respective datasets and we are specifically interested on the methods, a nested mixed effect model is adopted in this study as eq-1 where methods are kept fixed

and the data and response nested on them are considered random, i.e each data and response nested on their respective dataset have different intercept. The model is written as,

$$y_{ijkl} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \gamma_{k(j)} + \epsilon_{ik(j)l} \tag{1}$$

Where, $\epsilon \sim N(0, \sigma^2)$ and the random and fixed effects are assumed to satisfy following assumptions:

$$\sum_{i=1}^{5} \alpha_i = 0 \qquad \beta_j \sim N(0, \sigma^2_\beta) \qquad \gamma_{k(j)} \sim N(0, \sigma^2_{\gamma(\beta)})$$

Here, $i = 1, \ldots 5$ (Methods), $j = 1, \ldots, 4$ (Data), $k = 1, \ldots, n(j)$ (Response nested under data) and $l = 1, 2, 3$ (Replication)

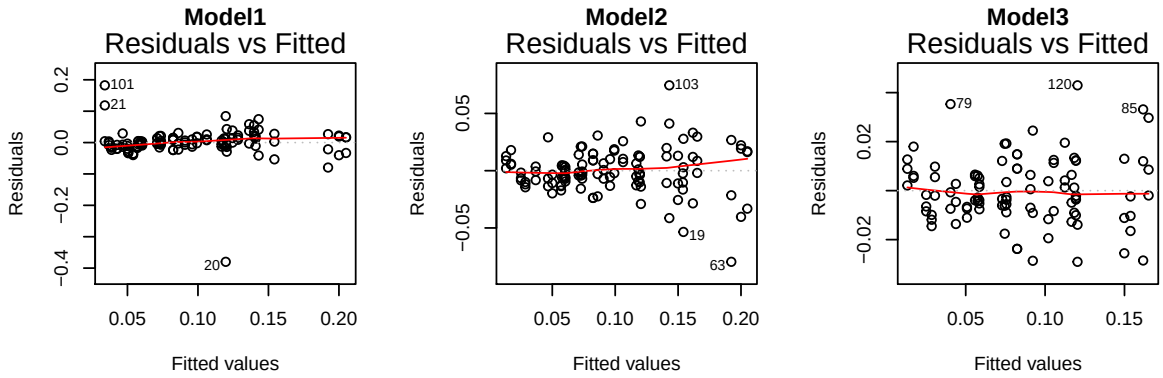

Figure 10: Residuals vs Fitted Plot

On fitting the model in eq-1, due to non homogeneous residuals and negative values in the response variable R^2 predicted, we transformed the response as, $\log[(p * (1-p)) + 1]$. The transformed model shows that methods are significant while data are not however the residual vs fitted plot suggested three (20, 21 and 101) models all from using MCUVE method on NIR data as outlier. However, the significance of the factors did not change even without these outlier. The residual vs fitted plot from the second model suggested 3 models (63, 19 and 103) which still are from using MCUVE method. Finally, a model without MCUVE method is fitted which has suggested that there is not any significance difference between different data and different methods but there is significant interaction between them.

Figure 11: Plot from Anova table. Number on top of each bars are p-value

Conclusion

Results from ANOVA, plotted in figure - 11, was expected since the MCUVE method of variables selection has selected few variables and has lost considerable amount of information. However, Model 3, without MCUVE method, shows that the variables selection methods performs differently on different nature of dataset (significant interaction between method and data). Furthermore, in all three models, the response which is nested under data are significant, i.e. responses for given data are significantly different.

On this comparison of variable selection methods, we found that there is no significance difference in variable selection methods apart from MCUVE which is rather conservative in selecting variables. However, the methods can perform differently according the nature of data. So, it is desirable to test the performance of variable selection before implementing them on any analysis.

Codes in Use

```
# PCA model Fitting
pca$milk <- prcomp(data$milk$MS)
## PLS model for MALDI-TOF milk Dataset
pls$milk <- lapply(colnames(data$milk$milk), function(rsp) {
  plsr(milk[, rsp] ~ MS, data = data$milk[data$milk$train, ], validation = "CV", segments = 30, segment.
     type = "consecutive", jackknife = TRUE)
})
names(pls$milk) <- colnames(data$milk$milk)
## PLS with integrated LDA for GeneExpression
gene$plda <- plda(data$gene$GeneExpr[data$gene$train, ], cat.resp = data$gene$tumor[data$gene$train],
                  fn = "plsr", fitComp = 25, split = 10)
## Getting Validation (R-sq predicted)
r2 <- lapply(names(pls), function(mdls) {
  getValidation(pls[[mdls]], newdata = data[[mdls]][!data[[mdls]]$train, ])
})
names(r2) <- names(pls)
# Variable Selection
subset <- subvaridx <- subdata <- submodel <- sub_r2 <- list()
```

```
## Filter with VIP
subset$vip <- lapply(names(pls), function(dta, vald = r2) {
  mdls <- lapply(names(pls[[dta]]), function(mdl) {
    opt.comp <- vald[[dta]][[2]][response == mdl & estimate == "CV", comp]
    vip <- VIP(pls[[dta]][[mdl]], opt.comp = opt.comp)
    vip.idx <- as.numeric(which(vip > 1))
    subdata <- within(data[[dta]], {assign(xy.info[[dta]][["x"]], data[[dta]][, xy.info[[dta]]["x"]][, vip
     .idx])})
    subMdl <- update(pls[[dta]][[mdl]], . ~ ., data = subdata[subdata$train, ])
    subvaridx$vip[[dta]][[mdl]] <<- vip.idx
    subdata$vip[[dta]][[mdl]] <<- subdata
    submodel$vip[[dta]][[mdl]] <<- subMdl
    return(vip)})
  names(mdls) <- names(pls[[dta]])
  sub_r2$vip[[dta]] <<- getValidation(submodel$vip[[dta]], subdata$vip[[dta]])
  return(mdls)
})
names(subset$vip) <- names(pls)
# Model Fitting for model comparison with data as random factor and response nested on it
mdl <- lm(value ~ method * r(data) + response %in% r(data), data = test.r2)
```

References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin, 2001.

[2] Kristian Hovde Liland. *Package 'plsVarSel'*. URL: `https://cran.r-project.org/web/packages/plsVarSel/plsVarSel.pdf`.

[3] Kristian Hovde Liland et al. "Quantitative whole spectrum analysis with MALDI-TOF MS, Part I: Measurement optimisation". In: *Chemometrics and Intelligent Laboratory Systems* 96.2 (2009), pp. 210–218.

[4] Harald Martens and Magni Martens. *Multivariate analysis of quality. An introduction*. 2001.

[5] Tahir Mehmood et al. "A review of variable selection methods in partial least squares regression". In: *Chemometrics and Intelligent Laboratory Systems* 118 (2012), pp. 62–69.

[6] *Microarray*. 2009. URL: `http://www.nature.com/scitable/content/microarray-6656746`.

[7] Tormod Næs et al. "Multi-block regression based on combinations of orthogonalisation, PLS-regression and canonical correlation analysis". In: *Chemometrics and Intelligent Laboratory Systems* 124 (2013), pp. 32–42.