Utility-Based Learning with UBL package

Paula Branco, Rita Ribeiro and Luís Torgo

LIAAD-INESC TEC
DCC-FCUP

19-Nov-2015

Overview

1. Problem Definition
2. Main Challenges and Solutions
   - Performance Assessment
   - Modelling Approaches
3. UBL Package
   - Introduction
   - Approaches available for classification tasks
   - Approaches available for regression tasks

What is Utility-based Predictive Analytics?

Context

- Predictive tasks
- Goal: obtain a good approximation of an unknown function $Y = f(X_1, X_2, \cdots, X_p)$
- This function maps a set of p predictor variables into a target variable Y which may be numeric (regression) or nominal (classification)
- Use a training set $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$ to obtain an approximation of $f()$

What is Utility-based Predictive Analytics?

Context

- Predictive tasks
- Goal: obtain a good approximation of an unknown function $Y = f(X_1, X_2, \cdots, X_p)$
- This function maps a set of p predictor variables into a target variable Y which may be numeric (regression) or nominal (classification)
- Use a training set $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$ to obtain an approximation of $f()$

User Preferences Biases

- Accurate predictions do not have the same benefit for the user and/or
- The different errors have differentiated costs.

What is Utility-based Predictive Analytics?

Context

- Predictive tasks
- Goal: obtain a good approximation of an unknown function $Y = f(X_1, X_2, \cdots, X_p)$
- This function maps a set of p predictor variables into a target variable Y which may be numeric (regression) or nominal (classification)
- Use a training set $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$ to obtain an approximation of $f()$

User Preferences Biases

- Accurate predictions do not have the same benefit for the user and/or
- The different errors have differentiated costs.
- The goal of the user is to maximize the utility (net balance between benefits and costs) of the predictions

Challenges in Learning Models for these Tasks

- when there is a **mismatch** between the more extreme situations in terms of utility (higher benefits or costs) and the distribution of Y on the training data;
- **standard evaluation criteria** (used for both learning and evaluating) take into account only the distribution of Y.
  - the feedback of these criteria does not reflect the preference biases of the user in terms of utility and thus can be misleading
  - models are not learned with the goal of maximizing utility

Main Challenges

Performance Assessment Measures: How can we evaluate the performance of the models considering the user preferences?

Modelling Approaches: How can we build models that take into consideration these preferences?

Performance Assessment Measures

Classification tasks

- precision, recall, F_β, geometric mean, dominance, index of balanced accuracy, optimized precision, adjusted geometric mean, H-measure, B42
- ROC curve, AUC, Precision-recall curves, Cost Curves, Lift Charts

Regression tasks

- LIN-LIN, QUAD-EXP, precision/recall, Mean Utility, Normalized Mean Utility
- RROC, AOC, REC curves, RECS

Evaluation taking into account the user preference biases

Assuming that we have domain knowledge on these biases, the best evaluation procedure is to maximize the utility.

$$U = \sum_{i=1}^{n} u(y_i, \hat{y}_i)$$

What is the domain knowledge?

- If we are considering a classification task, then $u(y_i, \hat{y}_i)$ is the **cost/benefit matrix** (Elkan, 2001)
- If we are considering a regression task, then we can use **utility surfaces** (Torgo and Ribeiro, 2007; Ribeiro, 2011)

Charles Elkan. 2001. The Foundations of Cost-Sensitive Learning. In *IJCAI'01: Proc. of 17th Int. Joint Conf. of Artificial Intelligence*, Vol. 1. Morgan Kaufmann Publishers, 973–978.

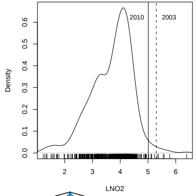
Luís Torgo and Rita P Ribeiro. 2007. Utility-Based Regression. In *PKDD'07: Proc. of 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases*. Springer, 597–604.

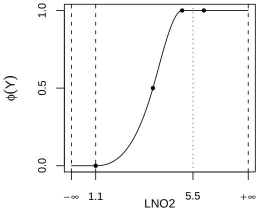
Rita P Ribeiro. 2011. Utility-based Regression. Ph.D. Dissertation. Dep. Computer Science, Faculty of Sciences - University of Porto.

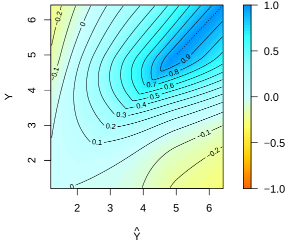
An example of Utility Surfaces

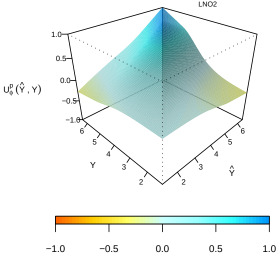
Prediction of Outdoor Air Pollution

- Positive Utility:
  - near main diagonal, growing towards top right corner.
- Negative Utility:
  - closer to top left and bottom right corners.







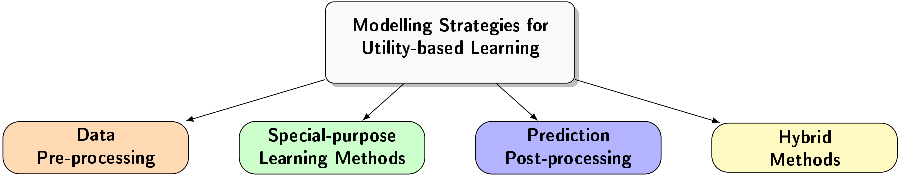


Modelling Strategies

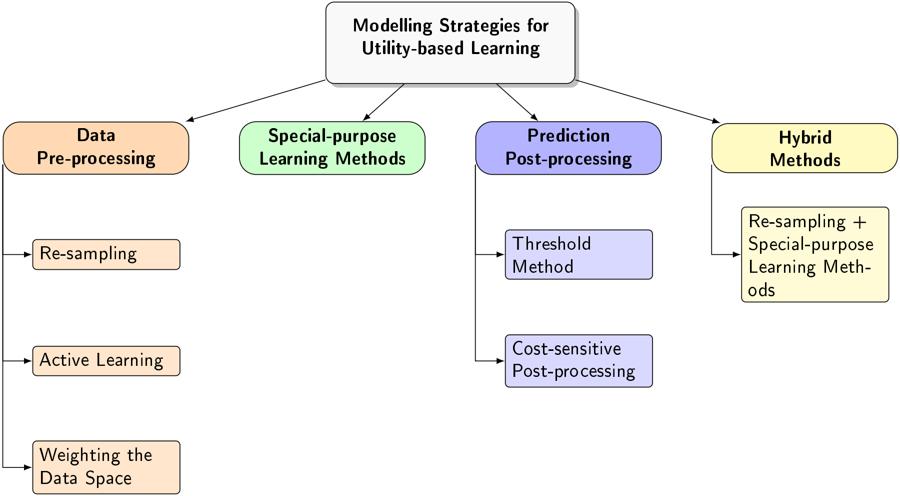
Utility Maximization can be achieved using one of the following main types of strategies:

- **Data Pre-processing**: change the original data distribution;
- **Special-purpose Learning Methods**: modify the internal preference criteria of models;
- **Prediction Post-processing**: post-process the model predictions; and
- **Hybrid Methods**: combination of the above strategies.

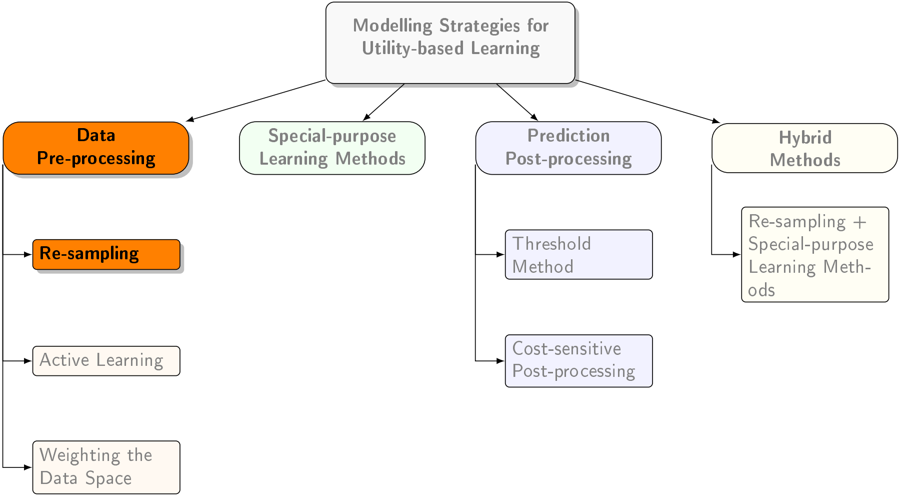
Modelling Strategies



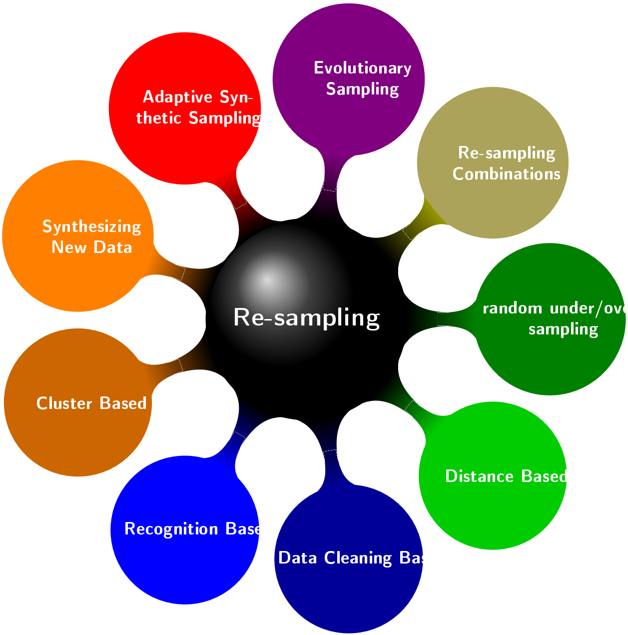
Modelling Strategies



Modelling Strategies



Re-sampling


Re-sampling
Evolutionary Sampling
Re-sampling Combinations
random under/ove sampling
Distance Based
Data Cleaning Bas
Recognition Base
Cluster Based
Synthesizing New Data
Adaptive Synthetic Sampling

Utility-based Learning in R - **UBL Package**

- Available on **github**: `https://github.com/paobranco/UBL`;
- **Implements different approaches** for addressing Utility-based learning problems (for the moment only of the re-sampling type);
- Suitable for **classification and regression** tasks;
- All the approaches implemented were adapted for dealing with **multiclass** problems;
- Includes a **package vignette** with detailed explanation of each approach, examples and analysis of the impact on the domain distribution.
- The user can choose from a set of **distance functions** to use (allows to deal with data sets containing nominal and numeric features).

UBL package installation and dependencies

Intallation

- `library(devtools)`
- `install_github("paobranco/UBL",ref="development")`

Dependencies

Package uba available at `http://www.dcc.fc.up.pt/~rpribeiro/uba`
`install.packages("uba_0.7.5.tar.gz",repos=NULL,dependencies=T)`

Approaches for classification

Functions named "⋆Classif"

- Random under/over-sampling
- Importance Sampling
- Tomek links
- Condensed Nearest Neighbors
- One-Sided Selection
- Edited Nearest Neighbors
- Neighborhood CLeaning rule
- Synthetic examples generation using Gaussian Noise
- Smote

Random Undersampling for classification tasks

```
data(iris)
data <- iris[-c(91:130),]
table(data$Species)

##
##     setosa versicolor  virginica
##         50         40         20

newDataB <- randUnderClassif(Species ~ ., data, C.perc ="balance")
newDataE <- randUnderClassif(Species ~ ., data, C.perc = "extreme")
newDataU <- randUnderClassif(Species ~ ., data,
                            C.perc= list(setosa=0.3, versicolor=0.8, virginica=1))
```

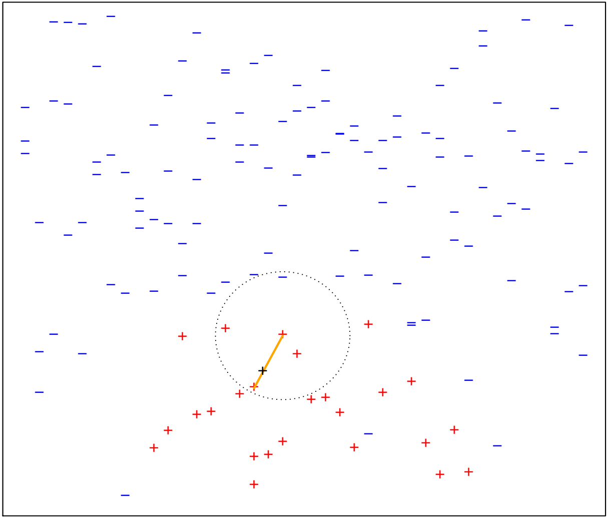
	setosa	versicolor	virginica
Original	50	40	20
newDataB	20	20	20
newDataE	8	10	20
newDataU	15	32	20

Tomek Links for classification tasks

```
  ir <- TomekClassif(Species~., data)
# use chebyshev distance, and select only two classes to under-sample
  irCheb <- TomekClassif(Species~., data, dist="Chebyshev",
                         Cl=c("virginica", "setosa"))
# use Manhattan distance, enable under-sampling in all classes, and
# select to break the link by only removing the example from the majority class
  irManM <- TomekClassif(Species~., data, dist="Manhattan", Cl="all", rem="maj")
  irManB <- TomekClassif(Species~., data, dist="Manhattan", Cl="all", rem="both")
```

	setosa	versicolor	virginica
Original	50	40	20
ir	50	38	18
irCheb	50	40	19
irManM	50	38	20
irManB	50	38	18

Smote Algorithm for classification tasks



Smote Algorithm for classification tasks

```
mysmote1 <- smoteClassif(Species~., data,
                         C.perc=list(setosa=0.6, virginica=1.5))
mysmote2 <- smoteClassif(Species~., data,
                         C.perc=list(setosa=0.2, versicolor=4), repl=TRUE)
mysmote3 <- smoteClassif(Species~., data,
                         C.perc=list(virginica=6, versicolor=2))
smoteB <- smoteClassif(Species~., data,
                         C.perc="balance")
smoteE <- smoteClassif(Species~., data,
                         C.perc="extreme")
```

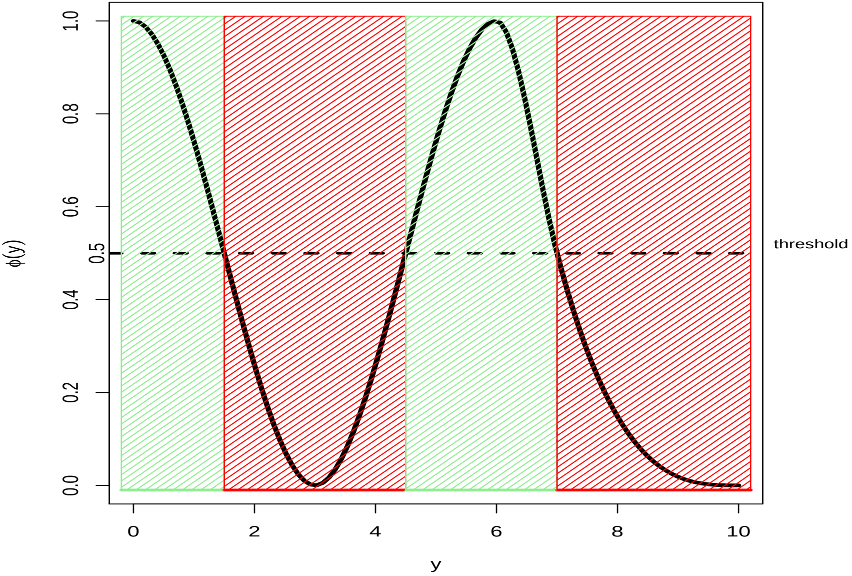
	setosa	versicolor	virginica
Original	50	40	20
mysmote1	30	40	30
mysmote2	10	160	20
mysmote3	50	80	120
smoteB	37	37	37
smoteE	23	29	58

Approaches for regression

Functions named "$\star$Regress"

- Random under/over-sampling
- Synthetic examples generation using Gaussian Noise
- SmoteR
- Importance Sampling

Relevance function with **uba package**



Random undersampling for regression tasks

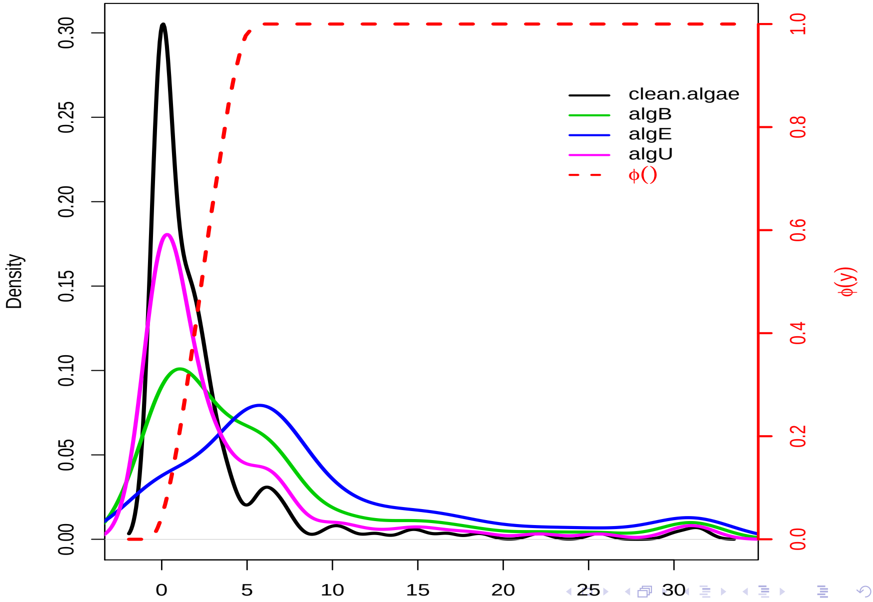
```r
# use algae data set with NA's removed
library(DMwR)
data(algae)
clean.algae <- algae[complete.cases(algae),]

# We start by using the automatic method for the relevance function
# Since this is the default behaviour, we can simply not mention the
# "rel" parameter

algB <- randUnderRegress(a7~., clean.algae, C.perc="balance")
algE <- randUnderRegress(a7~., clean.algae, C.perc="extreme")

# the automatic method for the relevance function provides only one bump
# with values to be under-sampled, thus we only need to indicate one percentage
algU <- randUnderRegress(a7~., clean.algae, C.perc=list(0.5))
```

Random undersampling for regression tasks

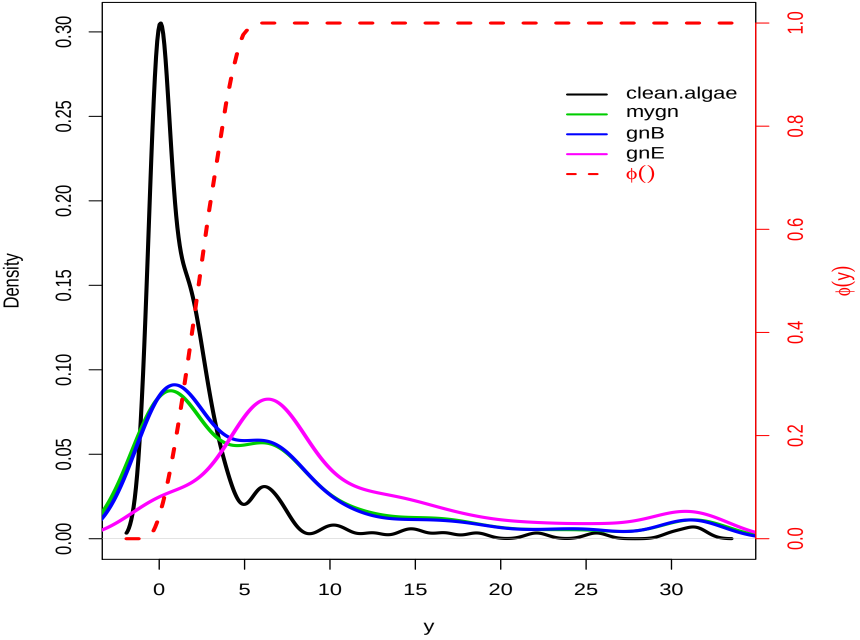


Synthetic examples with Gaussian Noise for regression tasks

```
# relevance function estimated automatically has two bumps
# defining the desired percentages of under and over-sampling to apply
C.perc=list(0.5, 3)

# define the relevance threshold
thr.rel=0.8

mygn <- gaussNoiseRegress(a7~., clean.algae, thr.rel=thr.rel, C.perc=C.perc)
gnB <- gaussNoiseRegress(a7~., clean.algae, thr.rel=thr.rel, C.perc="balance")
gnE <- gaussNoiseRegress(a7~., clean.algae, thr.rel=thr.rel, C.perc="extreme")
```


clean.algae
mygn
gnB
gnE
$\phi()$
Density
y
$\phi(y)$
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0

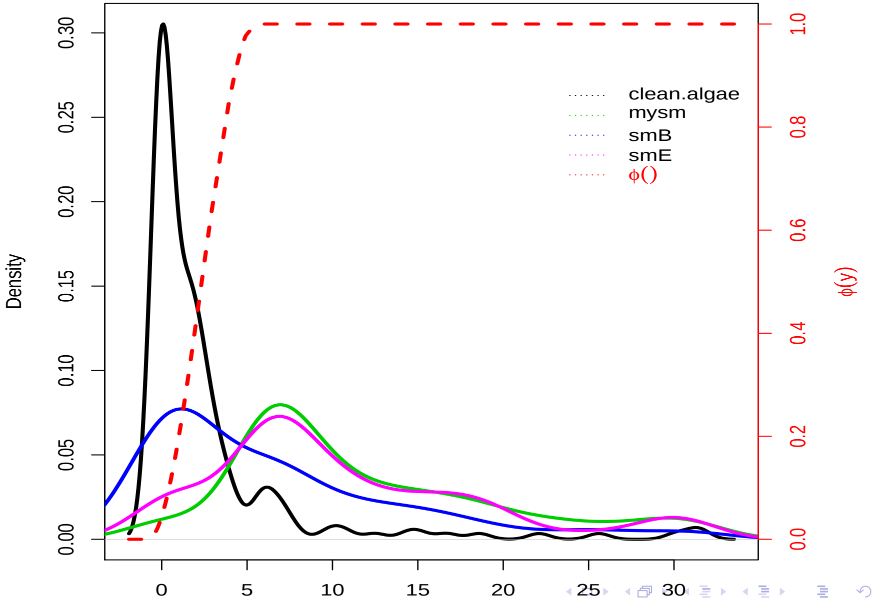
Smote for Regression (Torgo et al., 2013)

```
thr.rel=0.8
C.perc=list(0.1, 8)

mysm <- smoteRegress(a7~., clean.algae, thr.rel=thr.rel,
                     dist="HEOM", C.perc=C.perc)
smB <- smoteRegress(a7~., clean.algae, thr.rel=thr.rel,
                    dist="HEOM", C.perc="balance")
smE <- smoteRegress(a7~., clean.algae, thr.rel=thr.rel,
                    dist="HEOM", C.perc="extreme")
```

Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. 2013. SMOTE for Regression. In *Progress in Artificial Intelligence*. Springer, 378–389.

Smote for Regression



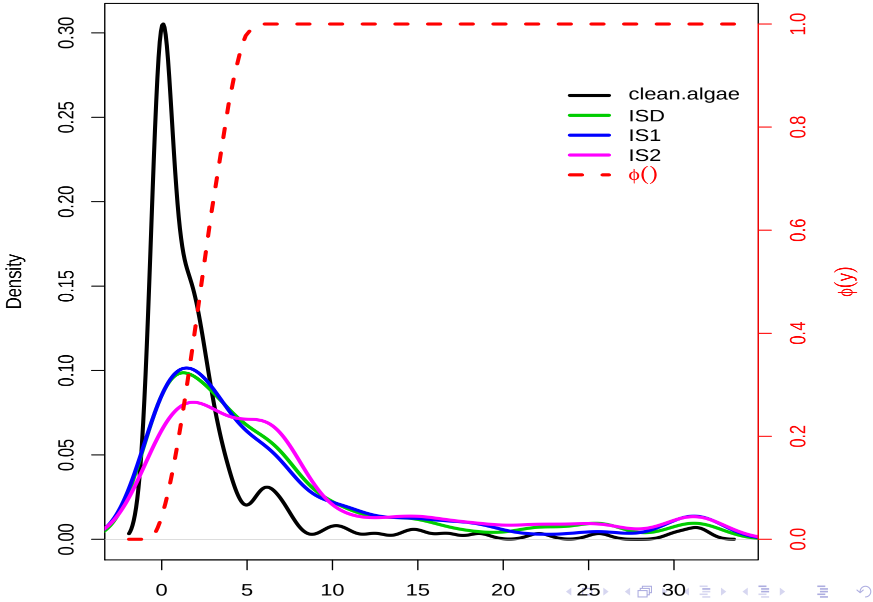
Importance Sampling for Regression Tasks

```
# relevance function is estimated automatically

# the default is not to use a relevance threshold and to assign equal
# importance to under and over-sampling, i.e., U=0.5 and O=0.5

ISD <- ImpSampRegress(a7~., clean.algae)
IS1 <- ImpSampRegress(a7~., clean.algae, U=0.9, O=0.2)
IS2 <- ImpSampRegress(a7~., clean.algae, U=0.5, O=0.8)
```

Importance Sampling for Regression Tasks



Summary

Main Challenges Utility-based Predictive Analytics

- Mismatch between the more extreme situations in terms of utility (higher benefits or costs) and the distribution of Y;
- Standard evaluation criteria are biased solely towards the distribution of Y.
- Solutions:
  - performance assessment based on utility maximization
  - modelling approaches:
    - pre-preprocessing
    - special purpose learning methods
    - post-processing
    - hybrid

UBL R Package

- Aims at providing a toolbox of methods for adressing utility-based predictive analytics tasks
- Currently implements re-sampling approaches (for both classification and regression tasks)

Utility-Based Learning with UBL package

Paula Branco, Rita Ribeiro and Luís Torgo

LIAAD-INESC TEC
DCC-FCUP

19-Nov-2015