Package: rcccd (via r-universe)

September 16, 2024

Type Package

Title Class Cover Catch Digraph Classification

Version 0.3.3

Description Fit Class Cover Catch Digraph Classification models that can be used in machine learning. Pure and proper and random walk approaches are available. Methods are explained in Priebe et al. (2001) <doi:10.1016/S0167-7152(01)00129-8>, Priebe et al. (2003) <doi:10.1007/s00357-003-0003-7>, and Manukyan and Ceyhan (2016) <doi:10.48550/arXiv.1904.04564>.

Depends R (>= 4.2)

License MIT + file LICENSE

Encoding UTF-8

LazyData false

RoxygenNote 7.3.0

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, RANN, Rfast, proxy

Repository https://fatihsaglam.r-universe.dev

RemoteUrl https://github.com/fatihsaglam/rcccd

RemoteRef HEAD

RemoteSha 177f0d7ce1ffd1a357ff7f351a4375b0916d689d

Contents

pcccd_classifier *Pure and Proper Class Cover Catch Digraph Classifier*

Description

pcccd_classifier fits a Pure and Proper Class Cover Catch Digraph (PCCCD) classification model.

Usage

```
pcccd_classifier(x, y, proportion = 1)
```

Arguments

x	feature matrix or dataframe.
y	class factor variable.
proportion	proportion of covered samples. A real number between $(0, 1]$. 1 by default. Smaller numbers results in less dominant samples.

Details

Multiclass framework for PCCCD. PCCCD determines target class dominant points set S and their circular cover area by determining balls $B(x^{\text{target}}, r_i)$ with radii r using minimum amount of dominant point which satisfies $X^{\text{non-target}} \cap \bigcup_i B_i = \varnothing$ (pure) and $X^{\text{target}} \subset \bigcup_i B_i$ (proper).

This guarantees that balls of target class never covers any non-target samples (pure) and balls cover all target samples (proper).

For detail, please refer to Priebe et al. (2001), Priebe et al. (2003), and Manukyan and Ceyhan (2016).

Note: Much faster than cccd package.

Value

an object of "cccd_classifier" which includes:

i_dominant_list	dominant sample indexes.
x_dominant_list	dominant samples from feature matrix, x
radii_dominant_list	Radiuses of the circle for dominant samples
class_names	class names
k_class	number of classes
proportions	proportions each class covered

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Priebe, C. E., DeVinney, J., & Marchette, D. J. (2001). On the distribution of the domination number for random class cover catch digraphs. Statistics & Probability Letters, 55(3), 239–246. https://doi.org/10.1016/s0167-7152(01)00129-8

Priebe, C. E., Marchette, D. J., DeVinney, J., & Socolinsky, D. A. (2003). Classification Using Class Cover Catch Digraphs. Journal of Classification, 20(1), 3–23. https://doi.org/10.1007/s00357-003-0003-7

Manukyan, A., & Ceyhan, E. (2016). Classification of imbalanced data with a geometric digraph family. Journal of Machine Learning Research, 17(1), 6504–6543. https://jmlr.org/papers/volume17/15-604/15-604.pdf

Examples

```
n <- 1000
x1 <- runif(n, 1, 10)
x2 <- runif(n, 1, 10)
x <- cbind(x1, x2)
y <- as.factor(ifelse(3 < x1 & x1 < 7 & 3 < x2 & x2 < 7, "A", "B"))

m_pcccd <- pcccd_classifier(x = x, y = y)

# dataset
plot(x, col = y, asp = 1)

# dominant samples of first class
x_center <- m_pcccd$x_dominant_list[[1]]

# radii of balls for first class
radii <- m_pcccd$radii_dominant_list[[1]]

# balls
for (i in 1:nrow(x_center)) {
xx <- x_center[i, 1]
yy <- x_center[i, 2]
r <- radii[i]
theta <- seq(0, 2*pi, length.out = 100)
xx <- xx + r*cos(theta)
yy <- yy + r*sin(theta)
lines(xx, yy, type = "l", col = "green")
}

# testing the performance
i_train <- sample(1:n, round(n*0.8))

x_train <- x[i_train,]
y_train <- y[i_train]
```

```
x_test <- x[-i_train,]
y_test <- y[-i_train]

m_pcccd <- pcccd_classifier(x = x_train, y = y_train)
pred <- predict(object = m_pcccd, newdata = x_test)

# confusion matrix
table(y_test, pred)

# test accuracy
sum(y_test == pred)/nrow(x_test)
```

---

pcccd_ensemble_classifier

Pure and Proper Class Cover Catch Digraph Ensemble Classifier

---

Description

pcccd_ensemble_classifier fits an Ensemble Pure and Proper Class Cover Catch Digraph (PCCCD) classification model.

Usage

```
pcccd_ensemble_classifier(
  x,
  y,
  n_model = 30,
  n_var = ncol(x),
  replace = FALSE,
  prop_sample = ifelse(replace, 1, 0.67),
  min_proportion = 0.7,
  max_proportion = 1,
  verbose = TRUE
)
```

Arguments

x	feature matrix or dataframe.
y	class factor variable.
n_model	an integer. Number of weak classifiers.
n_var	an integer. number of variables in weak classifiers.
replace	a bool. Should replacement be used in data sampling
prop_sample	a value between 0 and 1. Proportion the number of resampled samples to the number of samples in x.
min_proportion	Minimum proportion of cover proportion in weak classifiers.
max_proportion	Maximum proportion of cover proportion in weak classifiers.

Details

Bagging framework for PCCCD.

Value

an object of "cccd_classifier" which includes:

`i_dominant_list`
: dominant sample indexes.

`x_dominant_list`
: dominant samples from feature matrix, x

`radii_dominant_list`
: Radiuses of the circle for dominant samples

`class_names`
: class names

`k_class`
: number of classes

`proportions`
: proportions each class covered

Author(s)

Fatih Saglam, saglamf89@gmail.com

Examples

```
n <- 1000
```

---

predict.pcccd_classifier

Pure and Proper Class Cover Catch Digraph Prediction

Description

`predict.pcccd_classifier` makes prediction using `pcccd_classifier` object.

Usage

```
## S3 method for class 'pcccd_classifier'
predict(object, newdata, type = "pred", ...)
```

Arguments

`object`
: a `pcccd_classifier` object

`newdata`
: newdata as matrix or dataframe.

`type`
: "pred" or "prob". Default is "pred". "pred" is class estimations, "prob" is $n \times k$ matrix of class probabilities.

`...`
: not used.

Details

Estimations are based on nearest dominant neighbor in radius unit.

For detail, please refer to Priebe et al. (2001), Priebe et al. (2003), and Manukyan and Ceyhan (2016).

Value

a vector of class predictions (if type is "pred") or a $n \times p$ matrix of class probabilities (if type is "prob").

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Priebe, C. E., DeVinney, J., & Marchette, D. J. (2001). On the distribution of the domination number for random class cover catch digraphs. Statistics & Probability Letters, 55(3), 239–246. https://doi.org/10.1016/s0167-7152(01)00129-8

Priebe, C. E., Marchette, D. J., DeVinney, J., & Socolinsky, D. A. (2003). Classification Using Class Cover Catch Digraphs. Journal of Classification, 20(1), 3–23. https://doi.org/10.1007/s00357-003-0003-7

Manukyan, A., & Ceyhan, E. (2016). Classification of imbalanced data with a geometric digraph family. Journal of Machine Learning Research, 17(1), 6504–6543. https://jmlr.org/papers/volume17/15-604/15-604.pdf

Examples

```
n <- 1000
x1 <- runif(n, 1, 10)
x2 <- runif(n, 1, 10)
x <- cbind(x1, x2)
y <- as.factor(ifelse(3 < x1 & x1 < 7 & 3 < x2 & x2 < 7, "A", "B"))

# testing the performance
i_train <- sample(1:n, round(n*0.8))

x_train <- x[i_train,]
y_train <- y[i_train]

x_test <- x[-i_train,]
y_test <- y[-i_train]

m_pcccd <- pcccd_classifier(x = x_train, y = y_train)
pred <- predict(object = m_pcccd, newdata = x_test)

# confusion matrix
table(y_test, pred)

# test accuracy
```

```
sum(y_test == pred)/nrow(x_test)
```

---

predict.pcccd_ensemble_classifier
Pure and Proper Class Cover Catch Digraph Ensemble Prediction

---

Description

predict.pcccd_ensemble_classifier makes prediction using pcccd_ensemble_classifier object.

Usage

```
## S3 method for class 'pcccd_ensemble_classifier'
predict(object, newdata, type = "pred", ...)
```

Arguments

object	a rwcccd_classifier object
newdata	newdata as matrix or dataframe.
type	"pred" or "prob". Default is "pred". "pred" is class estimations, "prob" is $n \times k$ matrix of class probabilities.
...	not used.

Value

a vector of class predictions (if type is "pred") or a $n \times p$ matrix of class probabilities (if type is "prob").

Author(s)

Fatih Saglam, saglamf89@gmail.com

Examples

```
n <- 1000
```

predict.rwcccd_classifier
Random Walk Class Cover Catch Digraph Prediction

Description

predict.rwcccd_classifier makes prediction using rwcccd_classifier object.

Usage

```
## S3 method for class 'rwcccd_classifier'
predict(object, newdata, type = "pred", e = 0, ...)
```

Arguments

object	a rwcccd_classifier object
newdata	newdata as matrix or dataframe.
type	"pred" or "prob". Default is "pred". "pred" is class estimations, "prob" is $n \times k$ matrix of class probabilities.
e	0 or 1. Default is 0. Penalty based on T scores in rwcccd_classifier object.
...	not used.

Details

Estimations are based on nearest dominant neighbor in radius unit. e argument is used to penalize estimations based on T scores in rwcccd_classifier object.

For detail, please refer to Priebe et al. (2001), Priebe et al. (2003), and Manukyan and Ceyhan (2016).

Value

a vector of class predictions (if type is "pred") or a $n \times p$ matrix of class probabilities (if type is "prob").

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Priebe, C. E., DeVinney, J., & Marchette, D. J. (2001). On the distribution of the domination number for random class cover catch digraphs. Statistics & Probability Letters, 55(3), 239–246. https://doi.org/10.1016/s0167-7152(01)00129-8

Priebe, C. E., Marchette, D. J., DeVinney, J., & Socolinsky, D. A. (2003). Classification Using Class Cover Catch Digraphs. Journal of Classification, 20(1), 3–23. https://doi.org/10.1007/s00357-003-0003-7

Manukyan, A., & Ceyhan, E. (2016). Classification of imbalanced data with a geometric digraph family. Journal of Machine Learning Research, 17(1), 6504–6543. https://jmlr.org/papers/volume17/15-604/15-604.pdf

Examples

```
n <- 1000
x1 <- runif(n, 1, 10)
x2 <- runif(n, 1, 10)
x <- cbind(x1, x2)
y <- as.factor(ifelse(3 < x1 & x1 < 7 & 3 < x2 & x2 < 7, "A", "B"))

# testing the performance
i_train <- sample(1:n, round(n*0.8))

x_train <- x[i_train,]
y_train <- y[i_train]

x_test <- x[-i_train,]
y_test <- y[-i_train]

m_rwcccd <- rwcccd_classifier(x = x_train, y = y_train)
pred <- predict(object = m_rwcccd, newdata = x_test, e = 0)

# confusion matrix
table(y_test, pred)

# test accuracy
sum(y_test == pred)/nrow(x_test)
```

---

rwcccd_classifier *Random Walk Class Cover Catch Digraph Classifier*

Description

rwcccd_classifier and rwcccd_classifier_2 fits a Random Walk Class Cover Catch Digraph (RWCCCD) classification model. rwcccd_classifier uses C++ for speed and rwcccd_classifier_2 uses R language to determine balls.

Usage

```
rwcccd_classifier(x, y, method = "default", m = 1, proportion = 0.99)

rwcccd_classifier_2(
  x,
  y,
  method = "default",
  m = 1,
```

```
  proportion = 0.99,
  partial_ordering = FALSE
)
```

Arguments

x	feature matrix or dataframe.
y	class factor variable.
method	"default" or "balanced".
m	penalization parameter. Takes value in $[0, \infty)$.
proportion	proportion of covered samples. A real number between $(0, 1]$.
partial_ordering	TRUE or FALSE Default is FALSE TRUE uses partial ordering in determining dominant points. It orders incompletely but faster. Only for rwcccd_classifier_2.

Details

Random Walk Class Cover Catch Digraphs (RWCCD) are determined by calculating T_{target} score for each class as target class as

$$T_{\text{target}} = R_{\text{target}}(r_{\text{target}}) - \frac{r_{\text{target}} n_u}{2 d_m(x)}.$$

Here, r_{target} is radius and determined by maximum $R_{\text{target}}(r) - P_{\text{target}}(r)$ calculated for each target sample. $R_{\text{target}}(r)$ is

$$R_{\text{target}}(r) := w_{target} | z \in X^{\text{target}}_{n_{\text{target}}} : d(x^{\text{target}}, z) \leq r | - w_{non-target} | z \in X^{\text{non-target}}_{n_{\text{non-target}}} : d(x^{\text{target}}, z) \leq r |$$

and $P_{\text{target}}(r)$ is

$$P_{\text{target}}(r) = m \times d(x^{\text{target}}, z)^p.$$

$m = 0$ removes penalty. $w_{target} = 1$ for default and $w_{target} = n_{\text{target}/n_{\text{non-target}}}$ for balanced method. n_u is the number of uncovered samples in the current iteration and $d_m(x)$ is $\max d(x^{\text{target}}, x^{\text{uncovered}})$.

This method is more robust to noise compared to PCCCD However, balls covers classes improperly and $r = 0$ can be selected.

For detail, please refer to Priebe et al. (2001), Priebe et al. (2003), and Manukyan and Ceyhan (2016).

Value

a rwcccd_classifier object

i_dominant_list	dominant sample indexes.
x_dominant_list	dominant samples from feature matrix, x
radii_dominant_list	Radiuses of the circle for dominant samples

class_names	class names
k_class	number of classes
proportions	proportions each class covered

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Priebe, C. E., DeVinney, J., & Marchette, D. J. (2001). On the distribution of the domination number for random class cover catch digraphs. Statistics & Probability Letters, 55(3), 239–246. https://doi.org/10.1016/s0167-7152(01)00129-8

Priebe, C. E., Marchette, D. J., DeVinney, J., & Socolinsky, D. A. (2003). Classification Using Class Cover Catch Digraphs. Journal of Classification, 20(1), 3–23. https://doi.org/10.1007/s00357-003-0003-7

Manukyan, A., & Ceyhan, E. (2016). Classification of imbalanced data with a geometric digraph family. Journal of Machine Learning Research, 17(1), 6504–6543. https://jmlr.org/papers/volume17/15-604/15-604.pdf

Examples

```
n <- 500
x1 <- runif(n, 1, 10)
x2 <- runif(n, 1, 10)
x <- cbind(x1, x2)
y <- as.factor(ifelse(3 < x1 & x1 < 7 & 3 < x2 & x2 < 7, "A", "B"))

# dataset
m_rwcccd_1 <- rwcccd_classifier(x = x, y = y, method = "default", m = 1)

plot(x, col = y, asp = 1, main = "default")
# dominant samples of second class
x_center <- m_rwcccd_1$x_dominant_list[[2]]
# radii of balls for second class
radii <- m_rwcccd_1$radii_dominant_list[[2]]

# balls
for (i in 1:nrow(x_center)) {
  xx <- x_center[i, 1]
  yy <- x_center[i, 2]
  r <- radii[i]
  theta <- seq(0, 2*pi, length.out = 100)
  xx <- xx + r*cos(theta)
  yy <- yy + r*sin(theta)
  lines(xx, yy, type = "l", col = "green")
}

# dataset
m_rwcccd_2 <- rwcccd_classifier_2(x = x, y = y, method = "default", m = 1, partial_ordering = TRUE)
```

```
plot(x, col = y, asp = 1, main = "default, prartial_ordering = TRUE")
# dominant samples of second class
x_center <- m_rwcccd_2$x_dominant_list[[2]]
# radii of balls for second class
radii <- m_rwcccd_2$radii_dominant_list[[2]]

# balls
for (i in 1:nrow(x_center)) {
  xx <- x_center[i, 1]
  yy <- x_center[i, 2]
  r <- radii[i]
  theta <- seq(0, 2*pi, length.out = 100)
  xx <- xx + r*cos(theta)
  yy <- yy + r*sin(theta)
  lines(xx, yy, type = "l", col = "green")
}

# dataset
m_rwcccd_3 <- rwcccd_classifier(x = x, y = y, method = "balanced", m = 1, proportion = 0.5)

plot(x, col = y, asp = 1, main = "balanced, proportion = 0.5")
# dominant samples of second class
x_center <- m_rwcccd_3$x_dominant_list[[2]]
# radii of balls for second class
radii <- m_rwcccd_3$radii_dominant_list[[2]]

# balls
for (i in 1:nrow(x_center)) {
  xx <- x_center[i, 1]
  yy <- x_center[i, 2]
  r <- radii[i]
  theta <- seq(0, 2*pi, length.out = 100)
  xx <- xx + r*cos(theta)
  yy <- yy + r*sin(theta)
  lines(xx, yy, type = "l", col = "green")
}

# testing the performance
i_train <- sample(1:n, round(n*0.8))

x_train <- x[i_train,]
y_train <- y[i_train]

x_test <- x[-i_train,]
y_test <- y[-i_train]

m_rwcccd <- rwcccd_classifier(x = x_train, y = y_train, method = "balanced")
pred <- predict(object = m_rwcccd, newdata = x_test)

# confusion matrix
table(y_test, pred)
```

```
# accuracy
sum(y_test == pred)/nrow(x_test)
```

Index