Package: handwriterRF (via r-universe)

November 6, 2024

Type Package

Title Handwriting Analysis with Random Forests

Version 1.0.2.9000

Maintainer Stephanie Reinders <reinders.stephanie@gmail.com>

Description Perform forensic handwriting analysis of two scanned handwritten documents. This package implements the statistical method described by Madeline Johnson and Danica Ommen (2021) <doi:10.1002/sam.11566>. Similarity measures and a random forest produce a score-based likelihood ratio that quantifies the strength of the evidence in favor of the documents being written by the same writer or different writers.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), tibble

VignetteBuilder knitr

Depends R (>= 2.10)

Imports dplyr, handwriter, magrittr, purrr, ranger, reshape2, tidyr, tidyselect

Config/testthat/edition 3

URL https://github.com/CSAFE-ISU/handwriterRF

BugReports https://github.com/CSAFE-ISU/handwriterRF/issues

Config/pak/sysreqs cmake libglpk-dev make libgsl0-dev libmagick++-dev gsfonts jags libicu-dev libpng-dev libxml2-dev libssl-dev

Repository https://csafe-isu.r-universe.dev

RemoteUrl https://github.com/csafe-isu/handwriterrf

RemoteRef HEAD

RemoteSha b543658113a385d2ae62cb0f96bda76bc4191a25

Contents

`calculate_percent_rank` *Calculate Percent Ranks*

Description

Compares two handwriting samples scanned and saved a PNG images with the following steps:

1. `processDocument` splits the writing in both samples into component shapes, or graphs.
2. `get_clusters_batch` groups the graphs into clusters of similar shapes.
3. `get_cluster_fill_counts` counts the number of graphs assigned to each cluster.
4. `get_cluster_fill_rates` calculates the proportion of graphs assigned to each cluster. The cluster fill rates serve as a writer profile.
5. A similarity score is calculated between the cluster fill rates of the two documents using a random forest trained with **ranger**.
6. The similarity score is compared to reference samples of same writer and different writer similarity scores. The percent rank of the observed similarity score is returned for each sample. The percent rank for score x is calculated as the number of scores in the sample less than or equal to x divided by the total number of scores.

Usage

```
calculate_percent_rank(
  sample1_path,
  sample2_path,
  rforest = random_forest,
  project_dir = NULL
)
```

Arguments

sample1_path	A file path to a handwriting sample saved in PNG file format.
sample2_path	A file path to a second handwriting sample saved in PNG file format.
rforest	Optional. A random forest trained with **ranger**. If rforest is not given, the data object random_forest is used.
project_dir	Optional. A path to a directory where helper files will be saved. If no project directory is specified, the helper files will be saved to tempdir() and deleted before the function terminates.

Value

A list of two numbers

Examples

```
# Compare two samples from the same writer
s1 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r01.png"),
                  package = "handwriterRF")
s2 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r02.png"),
                  package = "handwriterRF")
calculate_slr(s1, s2)

# Compare samples from two writers
s1 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r01.png"),
                  package = "handwriterRF")
s2 <- system.file(file.path("extdata", "docs", "w0238_s01_pWOZ_r02.png"),
                  package = "handwriterRF")
calculate_slr(s1, s2)
```

---

calculate_score_with_clusters

Calculate a Similarity Score from Cluster Assignments

Description

Calculates a similarity score between the cluster assignments of two handwriting samples with these steps:

1. get_cluster_fill_counts counts the number of graphs assigned to each cluster.
2. get_cluster_fill_rates calculates the proportion of graphs assigned to each cluster. The cluster fill rates serve as a writer profile.
3. A similarity score is calculated between the cluster fill rates of the two documents using a random forest trained with **ranger**.

Usage

```
calculate_score_with_clusters(
  sample1_clusters,
  sample2_clusters,
  rforest = random_forest,
  project_dir = NULL
)
```

Arguments

sample1_clusters	A file path to cluster assignments created with [handwriter::get_clusters_batch()]
sample2_clusters	A file path to cluster assignments created with [handwriter::get_clusters_batch()]
rforest	Optional. A random forest trained with **ranger**. If rforest is not given, the data object random_forest is used.
project_dir	Optional. A path to a directory where the output data frame will be saved. If no project directory is specified, the output data frame will be returned but not saved.

Details

This function is primarily useful for users who want to calculate similarity scores between large numbers of handwriting samples. Follow these steps:

1. Run process_batch_dir on the folder containing the scanned handwriting samples. This splits the writing in the samples into component shapes, or graphs.
2. Run get_clusters_batch on the output folder that contains the graphs. This groups the graphs from each sample into clusters of similar shapes.
3. Run ‘calculate_score_with_clusters()‘ on pairs of files output in the previous step.

Value

A number between 0 and 1

Examples

```
# Compare two samples from the same writer
c1 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r01.rds"),
                  package = "handwriterRF")
c2 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r02.rds"),
                  package = "handwriterRF")
calculate_score_with_clusters(c1, c2)

# Compare samples from two writers
c1 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r01.rds"),
                  package = "handwriterRF")
c2 <- system.file(file.path("extdata", "clusters", "w0238_s01_pWOZ_r02.rds"),
                  package = "handwriterRF")
```

```
calculate_score_with_clusters(c1, c2)
```

`calculate_slr` *Calculate a Score-Based Likelihood Ratio*

Description

Compares two handwriting samples scanned and saved a PNG images with the following steps:

1. `processDocument` splits the writing in both samples into component shapes, or graphs.
2. `get_clusters_batch` groups the graphs into clusters of similar shapes.
3. `get_cluster_fill_counts` counts the number of graphs assigned to each cluster.
4. `get_cluster_fill_rates` calculates the proportion of graphs assigned to each cluster. The cluster fill rates serve as a writer profile.
5. A similarity score is calculated between the cluster fill rates of the two documents using a random forest trained with **ranger**.
6. The similarity score is compared to reference distributions of same writer and different writer similarity scores. The result is a score-based likelihood ratio that conveys the strength of the evidence in favor of same writer or different writer. For more details, see Madeline Johnson and Danica Ommen (2021) <doi:10.1002/sam.11566>.

Usage

```
calculate_slr(
  sample1_path,
  sample2_path,
  rforest = random_forest,
  project_dir = NULL
)
```

Arguments

`sample1_path`	A file path to a handwriting sample saved in PNG file format.
`sample2_path`	A file path to a second handwriting sample saved in PNG file format.
`rforest`	Optional. A random forest trained with **ranger**. If rforest is not given, the data object random_forest is used.
`project_dir`	Optional. A path to a directory where helper files will be saved. If no project directory is specified, the helper files will be saved to tempdir() and deleted before the function terminates.

Value

A number

Examples

```
# Compare two samples from the same writer
s1 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r01.png"),
                  package = "handwriterRF")
s2 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r02.png"),
                  package = "handwriterRF")
calculate_slr(s1, s2)

# Compare samples from two writers
s1 <- system.file(file.path("extdata", "docs", "w0030_s01_pWOZ_r01.png"),
                  package = "handwriterRF")
s2 <- system.file(file.path("extdata", "docs", "w0238_s01_pWOZ_r02.png"),
                  package = "handwriterRF")
calculate_slr(s1, s2)
```

---

calculate_slr_with_clusters

Calculate a Score-Based Likelihood Ratio from Cluster Assignments

---

Description

Calculates a score-based likelihood ratio between the cluster assignments of two handwriting samples with these steps:

1. `get_cluster_fill_counts` counts the number of graphs assigned to each cluster.
2. `get_cluster_fill_rates` calculates the proportion of graphs assigned to each cluster. The cluster fill rates serve as a writer profile.
3. A similarity score is calculated between the cluster fill rates of the two documents using a random forest trained with **ranger**.
4. The similarity score is compared to reference distributions of same writer and different writer similarity scores. The result is a score-based likelihood ratio that conveys the strength of the evidence in favor of same writer or different writer. For more details, see Madeline Johnson and Danica Ommen (2021) <doi:10.1002/sam.11566>.

Usage

```
calculate_slr_with_clusters(
  sample1_clusters,
  sample2_clusters,
  rforest = random_forest,
  project_dir = NULL
)
```

Arguments

`sample1_clusters`	A file path to cluster assignments created with [handwriter::get_clusters_batch()]
`sample2_clusters`	A file path to cluster assignments created with [handwriter::get_clusters_batch()]
`rforest`	Optional. A random forest trained with **ranger**. If rforest is not given, the data object random_forest is used.
`project_dir`	Optional. A path to a directory where helper files will be saved. If no project directory is specified, the helper files will be saved to tempdir() and deleted before the function terminates.

Details

This function is primarily useful for users who want to calculate score-based likelihood ratios between large numbers of handwriting samples. Follow these steps:

1. Run `process_batch_dir` on the folder containing the scanned handwriting samples. This splits the writing in the samples into component shapes, or graphs.
2. Run `get_clusters_batch` on the output folder that contains the graphs. This groups the graphs from each sample into clusters of similar shapes.
3. Run 'calculate_slr_with_clusters()' on pairs of files output in the previous step.

Value

A number great than or equal to zero

Examples

```
# Compare two samples from the same writer
c1 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r01.rds"),
                  package = "handwriterRF")
c2 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r02.rds"),
                  package = "handwriterRF")
calculate_slr_with_clusters(c1, c2)

# Compare samples from two writers
c2 <- system.file(file.path("extdata", "clusters", "w0030_s01_pWOZ_r01.rds"),
                  package = "handwriterRF")
c2 <- system.file(file.path("extdata", "clusters", "w0238_s01_pWOZ_r02.rds"),
                  package = "handwriterRF")
calculate_slr_with_clusters(c1, c2)
```

cfc *Cluster Fill Counts for 1200 CSAFE Handwriting Database Samples*

Description

A dataset containing cluster fill counts for for 1,200 handwriting samples from the CSAFE Handwriting Database. The documents were split into graphs with `process_batch_dir`. The graphs were grouped into clusters with `get_clusters_batch`. The cluster fill counts were calculated with `get_cluster_fill_counts`.

Usage

```
cfc
```

Format

A data frame with 1200 rows and 41 variables:

docname The file name of the handwriting sample. The file name includes the writer ID, the writing session, prompt, and repetition number of the handwriting sample. There are 1,200 handwriting samples.

writer Writer ID. There are 100 distinct writer ID's. Each writer has 12 documents.

doc A document code that records the writing session, prompt, and repetition number of the handwriting sample. There are 12 distinct document codes. Each writer has a writing sample for each of the 12 document codes.

1 The number of graphs in cluster 1

2 The number of graphs in cluster 2

3 The number of graphs in cluster 3

4 The number of graphs in cluster 4

5 The number of graphs in cluster 5

6 The number of graphs in cluster 6

7 The number of graphs in cluster 7

8 The number of graphs in cluster 8

9 The number of graphs in cluster 9

10 The number of graphs in cluster 10

11 The number of graphs in cluster 11

12 The number of graphs in cluster 12

13 The number of graphs in cluster 13

14 The number of graphs in cluster 14

15 The number of graphs in cluster 15

16 The number of graphs in cluster 16

17 The number of graphs in cluster 17
18 The number of graphs in cluster 18
19 The number of graphs in cluster 19
20 The number of graphs in cluster 20
21 The number of graphs in cluster 21
22 The number of graphs in cluster 22
23 The number of graphs in cluster 23
24 The number of graphs in cluster 24
25 The number of graphs in cluster 25
26 The number of graphs in cluster 26
27 The number of graphs in cluster 27
28 The number of graphs in cluster 28
29 The number of graphs in cluster 29
30 The number of graphs in cluster 30
31 The number of graphs in cluster 31
32 The number of graphs in cluster 32
33 The number of graphs in cluster 33
34 The number of graphs in cluster 34
35 The number of graphs in cluster 35
36 The number of graphs in cluster 36
37 The number of graphs in cluster 37
38 The number of graphs in cluster 38
39 The number of graphs in cluster 39
40 The number of graphs in cluster 40

Source

<https://forensicstats.org/handwritingdatabase/>

`cfr` *Cluster Fill Rates for 1200 CSAFE Handwriting Database Samples*

Description

A dataset containing cluster fill rates for for 1,200 handwriting samples from the CSAFE Handwriting Database. The dataset was created by running `get_cluster_fill_rates` on the cluster fill counts data frame cfc. Cluster fill rates are the proportion of total graphs assigned to each cluster.

Usage

```
cfr
```

Format

A data frame with 1200 rows and 42 variables:

docname file name of the handwriting sample

total_graphs The total number of graphs in the handwriting sample

cluster1 The number of graphs in cluster 1

cluster2 The number of graphs in cluster 2

cluster3 The number of graphs in cluster 3

cluster4 The number of graphs in cluster 4

cluster5 The number of graphs in cluster 5

cluster6 The number of graphs in cluster 6

cluster7 The number of graphs in cluster 7

cluster8 The number of graphs in cluster 8

cluster9 The number of graphs in cluster 9

cluster10 The number of graphs in cluster 10

cluster11 The number of graphs in cluster 11

cluster12 The number of graphs in cluster 12

cluster13 The number of graphs in cluster 13

cluster14 The number of graphs in cluster 14

cluster15 The number of graphs in cluster 15

cluster16 The number of graphs in cluster 16

cluster17 The number of graphs in cluster 17

cluster18 The number of graphs in cluster 18

cluster19 The number of graphs in cluster 19

cluster20 The number of graphs in cluster 20

cluster21 The number of graphs in cluster 21

cluster22 The number of graphs in cluster 22

cluster23 The number of graphs in cluster 23

cluster24 The number of graphs in cluster 24

cluster25 The number of graphs in cluster 25

cluster26 The number of graphs in cluster 26

cluster27 The number of graphs in cluster 27

cluster28 The number of graphs in cluster 28

cluster29 The number of graphs in cluster 29

cluster30 The number of graphs in cluster 30

cluster31 The number of graphs in cluster 31

cluster32 The number of graphs in cluster 32

cluster33 The number of graphs in cluster 33

cluster34 The number of graphs in cluster 34

cluster35 The number of graphs in cluster 35

cluster36 The number of graphs in cluster 36

cluster37 The number of graphs in cluster 37

cluster38 The number of graphs in cluster 38

cluster39 The number of graphs in cluster 39

cluster40 The number of graphs in cluster 40

Source

<https://forensicstats.org/handwritingdatabase/>

`get_cluster_fill_rates` *Get Cluster Fill Rates*

Description

Calculate cluster fill rates from a data frame of cluster fill counts created with `get_cluster_fill_counts`.

Usage

```
get_cluster_fill_rates(df)
```

Arguments

`df`	A data frame of cluster fill rates created with `get_cluster_fill_counts`.

Value

A data frame of cluster fill rates.

Examples

```
rates <- get_cluster_fill_rates(df = cfc)
```

get_csafe_train_set *Get Training Set*

Description

Create a training set from a data frame of cluster fill rates from the CSAFE Handwriting Database.

Usage

```
get_csafe_train_set(df, train_prompt_codes)
```

Arguments

df	A data frame of cluster fill rates created with get_cluster_fill_rates
train_prompt_codes	A character vector of which prompt(s) to use in the training set. Available prompts are 'pLND', 'pPHR', 'pWOZ', and 'pCMB'.

Value

A data frame

Examples

```
train <- get_csafe_train_set(df = cfr, train_prompt_codes = 'pCMB')
```

get_distances *Get Distances*

Description

Calculate distances using between all pairs of cluster fill rates in a data frame using one or more distance measures. The available distance measures absolute distance, Manhattan distance, Euclidean distance, maximum distance, and cosine distance.

Usage

```
get_distances(df, distance_measures)
```

Arguments

df	A data frame of cluster fill rates created with get_cluster_fill_rates
distance_measures	A vector of distance measures. Use 'abs' to calculate the absolute difference, 'man' for the Manhattan distance, 'euc' for the Euclidean distance, 'max' for the maximum absolute distance, and 'cos' for the cosine distance. The vector can be a single distance, or any combination of these five distance measures.

Details

The absolute distance between two n-length vectors of cluster fill rates, a and b, is a vector of the same length as a and b. It can be calculated as abs(a-b) where subtraction is performed element-wise, then the absolute value of each element is returned. More specifically, element i of the vector is $|a_i - b_i|$ for $i = 1, 2, ..., n$.

The Manhattan distance between two n-length vectors of cluster fill rates, a and b, is $\sum_{i=1}^{n} |a_i - b_i|$. In other words, it is the sum of the absolute distance vector.

The Euclidean distance between two n-length vectors of cluster fill rates, a and b, is $\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$. In other words, it is the sum of the elements of the absolute distance vector.

The maximum distance between two n-length vectors of cluster fill rates, a and b, is $\max_{1 \leq i \leq n} \{|a_i - b_i|\}$. In other words, it is the sum of the elements of the absolute distance vector.

The cosine distance between two n-length vectors of cluster fill rates, a and b, is $\sum_{i=1}^{n}(a_i - b_i)^2 / (\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2})$.

Value

A data frame of distances

Examples

```
# calculate maximum and Euclidean distances between the first 3 documents in cfr.
distances <- get_distances(df = cfr[1:3, ], distance_measures = c('max', 'euc'))

distances <- get_distances(df = cfr, distance_measures = c('man'))
```

---

interpret_slr *Interpret an SLR Value*

Description

Verbally interprent an SLR value.

Usage

```
interpret_slr(df)
```

Arguments

df	A data frame created by calculate_slr.

Value

A string

Examples

```
df <- data.frame("score" = 5, "slr" = 20)
interpret_slr(df)

df <- data.frame("score" = 0.12, "slr" = 0.5)
interpret_slr(df)

df <- data.frame("score" = 1, "slr" = 1)
interpret_slr(df)

df <- data.frame("score" = 0, "slr" = 0)
interpret_slr(df)
```

---

plot_histograms *Plot Histograms*

Description

Plot histograms of same writer and different writers reference similarity scores from a random forest created with [train_rf()]. Plot a vertical, dashed line at a similarity score calculated with [calculate_slr()] to see whether the score is more typical of the same writer or different writers reference scores.

Usage

```
plot_histograms(rforest, score = NULL)
```

Arguments

rforest	A random forest created with [train_rf()]
score	A similarity score calculated with [calculate_slr()]

Value

A ggplot2 plot of histograms

Examples

```
plot_histograms(rforest = random_forest)

# Add a vertical line 0.1 on the horizontal axis.
plot_histograms(rforest = random_forest, score = 0.1)
```

`random_forest` *A* **ranger** *Random Forest, Distances, and Similarity Scores*

Description

A list that contains a trained random forest created with **ranger**, the data frame of distances used to train the random forest, and similarity scores calculated from the training data.

Usage

```
random_forest
```

Format

A list with the following components:

dists The data frame used to train the random forest. The data frame has 600 rows. Each row contains the absolute and Euclidean distances between the cluster fill rates of two handwriting samples. If both handwriting samples are from the same writer, the class is 'same'. If the handwriting samples are from different writers, the class is 'different'. There are 300 'same' distances and 300 'different' distances in the data frame.

rf A random forest created with **ranger** with settings: importance = 'permutation', scale.permutation.importance = TRUE, and num.trees = 200.

scores A similarity score was obtained for each pair of handwriting samples in the training data frame, dists, by calculating the proportion of decision trees that voted 'same' class for the pair.

Examples

```
# view the random forest
random_forest$rf

# view the distances data frame
random_forest$dists

# plot histograms of the similarity scores and place a vertical
# line at similarity score 0.9.
plot_histograms(random_forest, 0.9)
```

templateK40 *Cluster Template with 40 Clusters*

Description

A cluster template created by **handwriter** with 40 clusters. This template was created from 120 handwriting samples from the CSAFE Handwriting Database.

Usage

```
templateK40
```

Format

A list containing the contents of the cluster template.

centers_seed An integer for the random number generator use to select the starting cluster centers for the K-Means algorithm.

cluster A vector of cluster assignments for each graph used to create the cluster template. The clusters are numbered sequentially 1, 2,...,K.

centers The final cluster centers produced by the K-Means algorithm.

K The number of clusters in the template.

n The number of training graphs to used to create the template.

docnames A vector that lists the training document from which each graph originated.

writers A vector that lists the writer of each graph.

iters The maximum number of iterations for the K-means algorithm.

changes A vector of the number of graphs that changed clusters on each iteration of the K-means algorithm.

outlierCutoff A vector of the outlier cutoff values calculated on each iteration of the K-means algorithm.

stop_reason The reason the K-means algorithm terminated.

wcd The within cluster distances on the final iteration of the K-means algorithm. More specifically, the distance between each graph and the center of the cluster to which it was assigned on each iteration. The output of `make_clustering_template`' stores the within cluster distances on each iteration, but the previous iterations were removed here to reduce the file size.

wcss A vector of the within-cluster sum of squares on each iteration of the K-means algorithm.

Details

handwriter splits handwriting samples into component shapes called graphs. The graphs are sorted into 40 clusters with a K-Means algorithm.

Examples

```
# view number of clusters
templateK40$K

# view number of iterations
templateK40$iters

# view cluster centers
templateK40$centers
```

`train_rf` *Train a Random Forest*

Description

Train a random forest with **ranger** from a data frame of cluster fill rates.

Usage

```
train_rf(
  df,
  ntrees,
  distance_measures,
  output_dir = NULL,
  run_number = 1,
  downsample = TRUE
)
```

Arguments

`df`	A data frame of cluster fill rates created with `get_cluster_fill_rates`
`ntrees`	An integer number of decision trees to use
`distance_measures`	A vector of distance measures. Any combination of 'abs', 'euc', 'man', 'max', and 'cos' may be used.
`output_dir`	A path to a directory where the random forest will be saved.
`run_number`	An integer used for both the set.seed function and to distinguish between different runs on the same input data frame.
`downsample`	Whether to downsample the number of different writer distances before training the random forest. If TRUE, the different writer distances will be randomly sampled, resulting in the same number of different writer and same writer pairs.

Value

A random forest

Examples

```
train <- get_csafe_train_set(df = cfr, train_prompt_code = 'pCMB')
rforest <- train_rf(
  df = train,
  ntrees = 200,
  distance_measures = c('euc'),
  run_number = 1,
  downsample = TRUE
)
```

Index