Package: handwriterRF (via r-universe)

October 26, 2026

Type Package

Title Handwriting Analysis with Random Forests

Version 1.1.1

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

Depends R (>= 3.5.0)

Imports dplyr, handwriter (>= 3.2.4), lifecycle, magrittr, purrr, ranger, reshape2, stringr, tidyr, tidyselect

Config/testthat/edition 3

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

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

VignetteBuilder knitr

NeedsCompilation no

Author Iowa State University of Science and Technology on behalf of its Center for Statistics and Applications in Forensic Evidence [aut, cph, fnd], Stephanie Reinders [aut, cre]

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

Date/Publication 2025-01-29 01:53:13 UTC

RemoteUrl https://github.com/cran/handwriterRF

RemoteRef HEAD

RemoteSha 9d016a6d75e26133f5e7664eaba1ffcf03a47bcb

Contents

calculate_slr . . . 2
cfc . . . 4
compare_documents . . . 5
compare_writer_profiles . . . 6
get_cluster_fill_rates . . . 7
get_distances . . . 8
get_rates_of_misleading_slrs . . . 9
get_ref_scores . . . 10
interpret_slr . . . 10
plot_scores . . . 11
random_forest . . . 12
ref_scores . . . 13
templateK40 . . . 14
test . . . 14
train . . . 16
train_rf . . . 18
validation . . . 20

Index **22**

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

Description

[Superseded] `calculate_slr` has been superseded in favor of `compare_documents()` which offers more functionality.

Usage

```
calculate_slr(
  sample1_path,
  sample2_path,
  rforest = NULL,
  reference_scores = NULL,
  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 no random forest is specified, `random_forest` will be used.
`reference_scores`	Optional. A dataframe of reference similarity scores. If reference scores is not specified, `ref_scores` will be used.
`project_dir`	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

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.

Value

A dataframe

Examples

```
# Compare two samples from the same writer
s1 <- system.file(file.path("extdata", "docs", "w0005_s01_pLND_r03.png"),
  package = "handwriterRF"
)
s2 <- system.file(file.path("extdata", "docs", "w0005_s02_pWOZ_r02.png"),
  package = "handwriterRF"
)
calculate_slr(s1, s2)

# Compare samples from two writers
s1 <- system.file(file.path("extdata", "docs", "w0005_s02_pWOZ_r02.png"),
  package = "handwriterRF"
)
```

```
s2 <- system.file(file.path("extdata", "docs", "w0238_s01_pWOZ_r02.png"),
  package = "handwriterRF"
)
calculate_slr(s1, s2)
```

`cfc` *A Dataframe of Cluster Fill Counts*

Description

The cfc dataframe contains cluster fill counts for two documents from the CSAFE Handwriting Database: w0238_s01_pWOZ_r02.rds and w0238_s01_pWOZ_r03.rds.

Usage

```
cfc
```

Format

A dataframe with 2 rows and 15 variables:

docname The file name of the handwriting sample.

writer Writer ID.

doc The name of the handwriting prompt.

3 The number of graphs in cluster 3.

10 The number of graphs in cluster 10.

12 The number of graphs in cluster 12.

15 The number of graphs in cluster 15.

16 The number of graphs in cluster 16.

17 The number of graphs in cluster 17.

19 The number of graphs in cluster 19.

20 The number of graphs in cluster 20.

23 The number of graphs in cluster 23.

25 The number of graphs in cluster 25.

27 The number of graphs in cluster 27.

29 The number of graphs in cluster 29.

Details

The documents were split into graphs with `process_batch_dir`. The graphs were grouped into clusters with `get_clusters_batch` and the cluster template `templateK40`. The number of graphs in each cluster, the cluster fill counts, were counted with `get_cluster_fill_counts`. The dataframe cfc has a column for each cluster in `templateK40` that has at least one graph from w0238_s01_pWOZ_r02.rds or w0238_s01_pWOZ_r03.rds assigned to it. Empty clusters do not have columns in cfc, so cfc only has 12 cluster columns instead of 40.

Source

https://forensicstats.org/handwritingdatabase/

`compare_documents` *Compare Documents*

Description

Compare two handwritten documents to predict whether they were written by the same person. Use either a similarity score or a score-based likelihood ratio as a comparison method.

Usage

```
compare_documents(
  sample1,
  sample2,
  score_only = TRUE,
  rforest = NULL,
  project_dir = NULL,
  reference_scores = NULL
)
```

Arguments

`sample1`	A filepath to a handwritten document scanned and saved as a PNG file.
`sample2`	A filepath to a handwritten document scanned and saved as a PNG file.
`score_only`	TRUE returns only the similarity score. FALSE returns the similarity score and a score-based likelihood ratio for that score, calculated using `reference_scores`.
`rforest`	Optional. A random forest created with `ranger::ranger()`. If a random forest is not supplied, `random_forest` will be used.
`project_dir`	Optional. A folder in which to save helper files and a CSV file with the results. If no project directory is supplied. Helper files will be saved to tempdir() > comparison but deleted before the function terminates. A CSV file with the results will not be saved, but a dataframe of the results will be returned.
`reference_scores`	Optional. A list of same writer and different writer similarity scores used for reference to calculate a score-based likelihood ratio. If reference scores are not supplied, `ref_scores` will be used only if `score_only` is FALSE. If score only is true, reference scores are unnecessary because a score-based likelihood ratio will not be calculated. If reference scores are supplied, `score_only` will automatically be set to FALSE.

Value

A dataframe

Examples

```
# Compare two documents from the same writer with a similarity score
s1 <- system.file(file.path("extdata", "docs", "w0005_s01_pLND_r03.png"),
  package = "handwriterRF"
)
s2 <- system.file(file.path("extdata", "docs", "w0005_s02_pWOZ_r02.png"),
  package = "handwriterRF"
)
compare_documents(s1, s2, score_only = TRUE)

# Compare two documents from the same writer with a score-based
# likelihood ratio
s1 <- system.file(file.path("extdata", "docs", "w0005_s01_pLND_r03.png"),
  package = "handwriterRF"
)
s2 <- system.file(file.path("extdata", "docs", "w0005_s02_pWOZ_r02.png"),
  package = "handwriterRF"
)
compare_documents(s1, s2, score_only = FALSE)
```

`compare_writer_profiles` *Compare Writer Profiles*

Description

Compare the writer profiles from two handwritten documents to predict whether they were written by the same person. Use either a similarity score or a score-based likelihood ratio as a comparison method.

Usage

```
compare_writer_profiles(
  writer_profiles,
  score_only = TRUE,
  rforest = NULL,
  reference_scores = NULL
)
```

Arguments

`writer_profiles`	A dataframe of writer profiles or cluster fill rates calculated with get_cluster_fill_rates
`score_only`	TRUE returns only the similarity score. FALSE returns the similarity score and a score-based likelihood ratio for that score, calculated using `reference_scores`.

rforest — Optional. A random forest created with ranger::ranger(). If a random forest is not supplied, random_forest will be used.

reference_scores — Optional. A list of same writer and different writer similarity scores used for reference to calculate a score-based likelihood ratio. If reference scores are not supplied, ref_scores will be used only if score_only is FALSE. If score only is true, reference scores are unnecessary because a score-based likelihood ratio will not be calculated. If reference scores are supplied, score_only will automatically be set to FALSE.

Value

A dataframe

Examples

```
compare_writer_profiles(test[1:2, ], score_only = TRUE)

compare_writer_profiles(test[1:2, ], score_only = FALSE)
```

---

get_cluster_fill_rates

Get Cluster Fill Rates

Description

[Deprecated] get_cluster_fill_rates is deprecated. Use get_cluster_fill_rates instead.

Usage

```
get_cluster_fill_rates(df)
```

Arguments

df — A dataframe of cluster fill rates created with get_cluster_fill_counts.

Value

A dataframe of cluster fill rates.

Examples

```
## Not run:
rates <- get_cluster_fill_rates(df = cfc)

## End(Not run)
```

`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 dataframe of cluster fill rates created with `get_cluster_fill_rates` and an added column that contains a writer ID.
`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 dataframe of distances

Examples

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

# calculate maximum and distances between all documents in test.
distances <- get_distances(df = test, distance_measures = c("man"))
```

`get_rates_of_misleading_slrs` *Get Rates of Misleading Evidence for SLRs*

Description

Calculate the rates of misleading evidence for score-based likelihood ratios (SLRs) when the ground truth is known.

Usage

```
get_rates_of_misleading_slrs(df, threshold = 1)
```

Arguments

`df`	A dataframe of SLRs from `compare_writer_profiles` with `score_only = FALSE`.
`threshold`	A number greater than zero that serves as a decision threshold. If the ground truth for two documents is that they came from the same writer and the SLR is less than the decision threshold, this is misleading evidence that incorrectly supports the defense (false negative). If the ground truth for two documents is that they came from different writers and the SLR is greater than the decision threshold, this is misleading evidence that incorrectly supports the prosecution (false positive).

Value

A list

Examples

```
comparisons <- compare_writer_profiles(test, score_only = FALSE)
get_rates_of_misleading_slrs(comparisons)
```

get_ref_scores *Get Reference Scores*

Description

Create reference scores of same writer and different writer scores from a dataframe of cluster fill rates.

Usage

```
get_ref_scores(rforest, df, seed = NULL, downsample_diff_pairs = FALSE)
```

Arguments

rforest	A **ranger** random forest created with train_rf.
df	A dataframe of cluster fill rates created with get_cluster_fill_rates with an added writer ID column.
seed	Optional. An integer to set the seed for the random number generator to make the results reproducible.
downsample_diff_pairs	If TRUE, the different writer pairs are down-sampled to equal the number of same writer pairs. If FALSE, all different writer pairs are used.

Value

A list of scores

Examples

```
get_ref_scores(rforest = random_forest, df = validation)
```

interpret_slr *Interpret an SLR Value*

Description

Verbally interprent an SLR value.

Usage

```
interpret_slr(df)
```

Arguments

df	A dataframe 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_scores *Plot Scores*

Description

Plot same writer and different writers reference similarity scores from a validation set. The similarity scores are greater than or equal to zero and less than or equal to one. The interval from 0 to 1 is split into n_bins. The proportion of scores in each bin is calculated and plotted. Optionally, a vertical dotted line may be plotted at an observed similarity score.

Usage

```
plot_scores(scores, obs_score = NULL, n_bins = 50)
```

Arguments

scores	A dataframe of scores calculated with get_ref_scores()
obs_score	Optional. A similarity score calculated with calculate_slr()
n_bins	The number of bins

Details

The methods used in this package typically produce many times more different writer scores than same writer scores. For example, ref_scores contains 79,600 different writer scores but only 200 same writer scores. Histograms, which show the frequency of scores, don't handle this class imbalance well. Instead, the rate of scores is plotted.

Value

A ggplot2 plot of histograms

Examples

```
plot_scores(scores = ref_scores)

plot_scores(scores = ref_scores, n_bins = 70)

# Add a vertical line 0.1 on the horizontal axis.
plot_scores(scores = ref_scores, obs_score = 0.1)
```

---

random_forest *A* **ranger** *Random Forest and Data Frame of Distances*

Description

A list that contains a trained random forest created with **ranger** and the dataframe of distances used to train the random forest.

Usage

```
random_forest
```

Format

A list with the following components:

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

distance_measures A vector of the distance measures used to train the random forest: c('abs', 'euc')

Examples

```
# view the random forest
random_forest$rf

# view the distance measures used to train the random forest
random_forest$distance_measures
```

`ref_scores` *Reference Similarity Scores*

Description

A list containing two dataframes. The same_writer dataframe contains similarity scores from same writer pairs. The diff_writer dataframe contains similarity scores from different writer pairs. The similarity scores are calculated from the validation dataframe with the following steps:

1. The absolute and Euclidean distances are calculated between pairs of writer profiles.
2. `random_forest` uses the distances between the pair to predict the class of the pair as same writer or different writer.
3. The proportion of decision trees that predict same writer is used as the similarity score.

Usage

```
ref_scores
```

Format

A list with the following components:

same_writer A dataframe of 1,800 same writer similarity scores. The columns docname1 and writer1 record the file name and the writer ID of the first handwriting sample. The columns docname2 and writer2 record the file name and writer ID of the second handwriting sample. The match column records the class, which is same, of the pairs of handwriting samples. The similarity scores between the pairs of handwriting samples are in the score column.

diff_writer A dataframe of 717,600 different writer similarity scores. The columns docname1 and writer1 record the file name and the writer ID of the first handwriting sample. The columns docname2 and writer2 record the file name and writer ID of the second handwriting sample. The match column records the class, which is different, of the pairs of handwriting samples. The similarity scores between the pairs of handwriting samples are in the score column.

Examples

```
summary(ref_scores$same_writer)

summary(ref_scores$diff_writer)

plot_scores(ref_scores)
```

templateK40 *Cluster Template with 40 Clusters*

Description

A cluster template created by **handwriter** with 40 clusters. This template was created from 100 handwriting samples from the CSAFE Handwriting Database, the CVL Handwriting Database, and the IAM Handwriting Database.

Usage

```
templateK40
```

Format

A list containing the contents of the cluster template.

cluster A vector of cluster assignments for each graph used to create the cluster template. The clusters are numbered sequentially 1, 2,...,40.

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

K The number of clusters in the template (40).

n The number of training graphs to used to create the template (32,708).

wcd The within cluster distances, the distance between each graph and the nearest cluster center, on the final 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

```
handwriter::plot_cluster_centers(templateK40)
```

test *A Test Set of Cluster Fill Rates*

Description

Writers from the CSAFE Handwriting Database and the CVL Handwriting Database were randomly assigned to train, validation, and test sets.

Usage

```
test
```

Format

A dataframe with 332 rows and 43 variables:

docname The file name of the handwriting sample.

writer Writer ID. There are 83 distinct writer ID's. Each writer has four documents in the dataframe.

doc The name of the handwriting prompt.

total_graphs The total number of graphs in the document.

cluster1 The proportion of graphs in cluster 1

cluster2 The proportion of graphs in cluster 2

cluster3 The proportion of graphs in cluster 3

cluster4 The proportion of graphs in cluster 4

cluster5 The proportion of graphs in cluster 5

cluster6 The proportion of graphs in cluster 6

cluster7 The proportion of graphs in cluster 7

cluster8 The proportion of graphs in cluster 8

cluster9 The proportion of graphs in cluster 9

cluster10 The proportion of graphs in cluster 10

cluster11 The proportion of graphs in cluster 11

cluster12 The proportion of graphs in cluster 12

cluster13 The proportion of graphs in cluster 13

cluster14 The proportion of graphs in cluster 14

cluster15 The proportion of graphs in cluster 15

cluster16 The proportion of graphs in cluster 16

cluster17 The proportion of graphs in cluster 17

cluster18 The proportion of graphs in cluster 18

cluster19 The proportion of graphs in cluster 19

cluster20 The proportion of graphs in cluster 20

cluster21 The proportion of graphs in cluster 21

cluster22 The proportion of graphs in cluster 22

cluster23 The proportion of graphs in cluster 23

cluster24 The proportion of graphs in cluster 24

cluster25 The proportion of graphs in cluster 25

cluster26 The proportion of graphs in cluster 26

cluster27 The proportion of graphs in cluster 27

cluster28 The proportion of graphs in cluster 28

cluster29 The proportion of graphs in cluster 29

cluster30 The proportion of graphs in cluster 30

cluster31 The proportion of graphs in cluster 31

cluster32 The proportion of graphs in cluster 32

cluster33 The proportion of graphs in cluster 33

cluster34 The proportion of graphs in cluster 34

cluster35 The proportion of graphs in cluster 35

cluster36 The proportion of graphs in cluster 36

cluster37 The proportion of graphs in cluster 37

cluster38 The proportion of graphs in cluster 38

cluster39 The proportion of graphs in cluster 39

cluster40 The proportion of graphs in cluster 40

Details

The test dataframe contains cluster fill rates for 332 handwritten documents from the CSAFE Handwriting Database and the CVL Handwriting Database. The documents are from 83 writers. The CSAFE Handwriting Database has nine repetitions of each prompt. Two London Letter prompts and two Wizard of Oz prompts were randomly selected from each writer. The CVL Handwriting Database does not contain multiple repetitions of prompts and four Engligh language prompts were randomly selected from each writer.

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`. Finally, `get_cluster_fill_rates` calculated the cluster fill rates.

Source

https://forensicstats.org/handwritingdatabase/, https://cvl.tuwien.ac.at/research/cvl-databases/an-off-line-database-for-writer-retrieval-writer-identification-and-word-spotting/

---

train *A Training Set of Cluster Fill Rates*

Description

Writers from the CSAFE Handwriting Database and the CVL Handwriting Database were randomly assigned to train, validation, and test sets.

Usage

```
train
```

Format

A dataframe with 800 rows and 43 variables:

docname The file name of the handwriting sample.

writer Writer ID. There are 200 distinct writer ID's. Each writer has 4 documents in the dataframe.

doc The name of the handwriting prompt.

total_graphs The total number of graphs in the document.

cluster1 The proportion of graphs in cluster 1

cluster2 The proportion of graphs in cluster 2

cluster3 The proportion of graphs in cluster 3

cluster4 The proportion of graphs in cluster 4

cluster5 The proportion of graphs in cluster 5

cluster6 The proportion of graphs in cluster 6

cluster7 The proportion of graphs in cluster 7

cluster8 The proportion of graphs in cluster 8

cluster9 The proportion of graphs in cluster 9

cluster10 The proportion of graphs in cluster 10

cluster11 The proportion of graphs in cluster 11

cluster12 The proportion of graphs in cluster 12

cluster13 The proportion of graphs in cluster 13

cluster14 The proportion of graphs in cluster 14

cluster15 The proportion of graphs in cluster 15

cluster16 The proportion of graphs in cluster 16

cluster17 The proportion of graphs in cluster 17

cluster18 The proportion of graphs in cluster 18

cluster19 The proportion of graphs in cluster 19

cluster20 The proportion of graphs in cluster 20

cluster21 The proportion of graphs in cluster 21

cluster22 The proportion of graphs in cluster 22

cluster23 The proportion of graphs in cluster 23

cluster24 The proportion of graphs in cluster 24

cluster25 The proportion of graphs in cluster 25

cluster26 The proportion of graphs in cluster 26

cluster27 The proportion of graphs in cluster 27

cluster28 The proportion of graphs in cluster 28

cluster29 The proportion of graphs in cluster 29

cluster30 The proportion of graphs in cluster 30

cluster31 The proportion of graphs in cluster 31

cluster32 The proportion of graphs in cluster 32

cluster33 The proportion of graphs in cluster 33

cluster34 The proportion of graphs in cluster 34

cluster35 The proportion of graphs in cluster 35

cluster36 The proportion of graphs in cluster 36

cluster37 The proportion of graphs in cluster 37

cluster38 The proportion of graphs in cluster 38

cluster39 The proportion of graphs in cluster 39

cluster40 The proportion of graphs in cluster 40

Details

The train dataframe contains cluster fill rates for 800 handwritten documents from the CSAFE Handwriting Database and the CVL Handwriting Database. The documents are from 200 writers. The CSAFE Handwriting Database has nine repetitions of each prompt. Two London Letter prompts and two Wizard of Oz prompts were randomly selected from each writer. The CVL Handwriting Database does not contain multiple repetitions of prompts and four English language prompts were randomly selected from each writer.

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`. Finally, `get_cluster_fill_rates` calculated the cluster fill rates.

Source

https://forensicstats.org/handwritingdatabase/, https://cvl.tuwien.ac.at/research/cvl-databases/an-off-line-database-for-writer-retrieval-writer-identification-and-word-spotting/

---

train_rf *Train a Random Forest*

Description

Train a random forest with **ranger** from a dataframe of writer profiles estimated with `get_cluster_fill_rates`. `train_rf` calculates the distance between all pairs of writer profiles using one or more distance measures. Currently, the available distance measures are absolute, Manhattan, Euclidean, maximum, and cosine.

Usage

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

Arguments

df	A dataframe of writer profiles 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 dataframe.
downsample_diff_pairs	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.

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 random forest

Examples

```
rforest <- train_rf(
  df = train,
  ntrees = 200,
  distance_measures = c("euc"),
  run_number = 1,
  downsample = TRUE
)
```

`validation` *A Validation Set of Cluster Fill Rates*

Description

Writers from the CSAFE Handwriting Database and the CVL Handwriting Database were randomly assigned to train, validation, and test sets.

Usage

```
validation
```

Format

A dataframe with 1,200 rows and 43 variables:

docname The file name of the handwriting sample.

writer Writer ID. There are 300 distinct writer ID's. Each writer has 4 documents in the dataframe.

doc The name of the handwriting prompt.

total_graphs The total number of graphs in the document.

cluster1 The proportion of graphs in cluster 1

cluster2 The proportion of graphs in cluster 2

cluster3 The proportion of graphs in cluster 3

cluster4 The proportion of graphs in cluster 4

cluster5 The proportion of graphs in cluster 5

cluster6 The proportion of graphs in cluster 6

cluster7 The proportion of graphs in cluster 7

cluster8 The proportion of graphs in cluster 8

cluster9 The proportion of graphs in cluster 9

cluster10 The proportion of graphs in cluster 10

cluster11 The proportion of graphs in cluster 11

cluster12 The proportion of graphs in cluster 12

cluster13 The proportion of graphs in cluster 13

cluster14 The proportion of graphs in cluster 14

cluster15 The proportion of graphs in cluster 15

cluster16 The proportion of graphs in cluster 16

cluster17 The proportion of graphs in cluster 17

cluster18 The proportion of graphs in cluster 18

cluster19 The proportion of graphs in cluster 19

cluster20 The proportion of graphs in cluster 20

cluster21 The proportion of graphs in cluster 21

cluster22 The proportion of graphs in cluster 22

cluster23 The proportion of graphs in cluster 23

cluster24 The proportion of graphs in cluster 24

cluster25 The proportion of graphs in cluster 25

cluster26 The proportion of graphs in cluster 26

cluster27 The proportion of graphs in cluster 27

cluster28 The proportion of graphs in cluster 28

cluster29 The proportion of graphs in cluster 29

cluster30 The proportion of graphs in cluster 30

cluster31 The proportion of graphs in cluster 31

cluster32 The proportion of graphs in cluster 32

cluster33 The proportion of graphs in cluster 33

cluster34 The proportion of graphs in cluster 34

cluster35 The proportion of graphs in cluster 35

cluster36 The proportion of graphs in cluster 36

cluster37 The proportion of graphs in cluster 37

cluster38 The proportion of graphs in cluster 38

cluster39 The proportion of graphs in cluster 39

cluster40 The proportion of graphs in cluster 40

Details

The validation dataframe contains cluster fill rates for 1,200 handwritten documents from the CSAFE Handwriting Database and the CVL Handwriting Database. The documents are from 300 writers. The CSAFE Handwriting Database has nine repetitions of each prompt. Two London Letter prompts and two Wizard of Oz prompts were randomly selected from each writer. The CVL Handwriting Database does not contain multiple repetitions of prompts and four English language prompts were randomly selected from each writer.

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`. Finally, `get_cluster_fill_rates` calculated the cluster fill rates.

Source

https://forensicstats.org/handwritingdatabase/, https://cvl.tuwien.ac.at/research/cvl-databases/an-off-line-database-for-writer-retrieval-writer-identification-and-word-spotting/

Index

∗ **cluster**
templateK40, 14
∗ **datasets**
cfc, 4
random_forest, 12
ref_scores, 13
test, 14
train, 16
validation, 20

calculate_slr, 2, *11*
calculate_slr(), *11*
cfc, 4
compare_documents, 5
compare_writer_profiles, 6, *9*

get_cluster_fill_counts, *3, 4*, *7*, *16*, *18*, *21*
get_cluster_fill_rates, *3*, *6*, 7, *7, 8*, *10*, *16*, *18, 19*, *21*
get_clusters_batch, *3, 4*, *16*, *18*, *21*
get_distances, 8
get_rates_of_misleading_slrs, 9
get_ref_scores, 10
get_ref_scores(), *11*

interpret_slr, 10

plot_scores, 11
process_batch_dir, *4*, *16*, *18*, *21*
processDocument, *3*

random_forest, 12
ranger::ranger(), *5*, *7*
ref_scores, 13

templateK40, *4*, 14
test, 14
train, 16
train_rf, *10*, 18

validation, 20