Package: sorenR (via r-universe)

November 3, 2026

Title Experimental Self-Organizing Representations with Bounded Updates

Version 0.0.1

Description A pure R experimental prototype graph representation with a deterministic out-of-sample map, frozen preprocessing, bounded replay, and transactional updates checked against a finite-anchor displacement budget. Support scores are uncalibrated and methodological superiority is not established.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics, utils

Suggests testthat (>= 3.1.7), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-11-03 12:33:30 UTC

RemoteUrl https://github.com/CTTIR/sorenR

RemoteRef HEAD

RemoteSha 1e0904119c1bac14f1c588ce230adac76f2469cb

Contents

print.soren_model 2
soren_control 3
soren_diagnostics 4
soren_embedding 5
soren_fit 5
soren_neighborhood_audit 6
soren_save 8

soren_transform 8
soren_update 9

Index **10**

`print.soren_model` *Display an experimental representation*

Description

Display an experimental representation

Usage

```
## S3 method for class 'soren_model'
print(x, ...)

## S3 method for class 'soren_model'
summary(object, ...)

## S3 method for class 'soren_model'
plot(
  x,
  ...,
  xlab = "SOREN 1",
  ylab = "SOREN 2",
  main = "Exploratory retained representation"
)
```

Arguments

`x`, `object`	A fitted model.
`...`	Additional plotting arguments (unused for print and summary).
`xlab`, `ylab`	Axis labels for the plot.
`main`	Plot title.

Value

Print returns the model invisibly; summary returns diagnostics; plot returns `NULL` invisibly.

soren_control *Configure an experimental SOREN model*

Description

All budgets refer to standardized input space or raw output coordinates. Adaptive support scores are heuristics, not calibrated tests of drift.

Usage

```
soren_control(
  max_prototypes = 32L,
  initial_prototypes = 8L,
  memory_size = 128L,
  neighbors = 3L,
  iterations = 30L,
  learning_rate = 0.2,
  stability_mode = "fixed",
  stability_budget = 0.01,
  tolerance = 1e-10,
  repulsion = 0.1,
  clip = 3,
  growth_threshold = 2,
  growth_min_support = 3L,
  history_size = 32L,
  chunk_size = 1024L,
  seed = 42L
)
```

Arguments

max_prototypes	Maximum codebook size (at least two).
initial_prototypes	Initial codebook size, at most the maximum.
memory_size	Maximum retained observations (at least two).
neighbors	Maximum mutual nearest neighbors per prototype.
iterations	Maximum layout descent iterations per fit or update.
learning_rate	Prototype step in (0,1].
stability_mode	One of fixed, adaptive, frozen, unrestricted.
stability_budget	Mean squared displacement budget on old replay anchors.
tolerance	Absolute numerical feasibility tolerance.
repulsion	Nonedge loss multiplier, nonnegative.
clip	Maximum individual standardized residual norm.

`growth_threshold`
: Growth residual threshold in bandwidth units.

`growth_min_support`
: Minimum observations supporting a growth proposal.

`history_size`
: Maximum retained update diagnostic records.

`chunk_size`
: Maximum rows per transform allocation.

`seed`
: Local random seed; caller random state is restored.

Value

A validated list of class `soren_control`.

`soren_diagnostics` *Inspect a fitted representation*

Description

Inspect a fitted representation

Usage

```
soren_diagnostics(object, newdata = NULL)
```

Arguments

`object`
: A fitted model.

`newdata`
: Optional observations for support diagnostics.

Details

Prototype stress compares all pairwise standardized input distances with raw output distances; it is not a biological ground truth or an uncertainty probability. Input sensitivity and refit stability require separate experiments.

Value

A list of graph counts, retained memory, raw finite-anchor displacement, support information and prototype distance distortion. Scores are uncalibrated.

soren_embedding *Retrieve the coordinates of retained items*

Description

Retrieve the coordinates of retained items

Usage

```
soren_embedding(object, type = c("replay", "prototypes"))
```

Arguments

object	A fitted model.
type	Either replay slots or prototypes.

Value

A coordinate matrix with identifiers in row names. Replay IDs may occur twice when reservoir and coverage slots retain the same observation.

soren_fit *Fit an experimental prototype graph representation*

Description

Fit an experimental prototype graph representation

Usage

```
soren_fit(x, n_components = 2L, control = soren_control(), row_ids = NULL)
```

Arguments

x	Dense finite numeric matrix; columns are features.
n_components	Output dimension, between one and the number of features.
control	Validated settings from soren_control().
row_ids	Optional unique character observation identifiers.

Details

Preprocessing is learned only here and frozen for future observations. The method is experimental; improved retention or adaptation is not established.

Value

A soren_model with prototypes and bounded standardized replay.

Examples

```
x <- cbind(seq_len(30), sin(seq_len(30)))
model <- soren_fit(x)
soren_transform(model, x[1:3, , drop = FALSE])
```

---

soren_neighborhood_audit

Audit neighborhoods on supplied query and reference observations

---

Description

Audit neighborhoods on supplied query and reference observations

Usage

```
soren_neighborhood_audit(
  object,
  query,
  reference,
  k = 15L,
  exclude_reference = NULL,
  chunk_size = object$control$chunk_size
)
```

Arguments

object	A fitted soren_model.
query	Dense finite numeric query matrix with the exact training schema. Zero rows are allowed.
reference	Nonempty dense finite numeric reference matrix with the exact training schema. The caller supplies and checks the provenance of both banks.
k	Positive integer number of neighbors, no greater than the eligible reference count for any query.
exclude_reference	NULL for no exclusions, or one integer-valued index per query: zero excludes nothing; a positive index excludes that reference row in both spaces. No self-match is inferred from names or values.
chunk_size	Positive integer maximum rows per query/reference block.

Details

The reference metric is squared Euclidean distance after the fitted, frozen per-feature centering and scaling. It is compared with squared Euclidean distance in the current embedding. Neither bank updates the model. Independent evaluation requires caller-supplied observations not used for fitting or tuning; this function cannot certify that independence. This is a geometric diagnostic, not biological validation, a calibrated probability, or evidence of superiority over another method.

Ties use exact equality of computed distances. If c observations are closer than the kth distance and e are tied there, each tied observation has weight (k-c)/e. Recall is the sum of products of input and embedding membership weights, divided by k. This is expected overlap under independent uniform choices within each boundary tie block, calculated without random draws. Identical fully tied spaces score k/N, where N is the eligible reference count, rather than one. Boundary sizes and weights help interpret collapse. Duplicate coordinates remain distinct candidates unless explicitly excluded. For a self-audit, supply `exclude_reference = seq_len(nrow(query))` when reference and query contain the same rows in the same order.

Mapping uses the existing map one row at a time in a fixed prototype order, so audit block size does not change its reduction path. Bulk transformation can differ at floating-point roundoff. Direct coordinate differences avoid squared-norm cancellation; intermediate nonfinite values cause an error. Caller inputs, model, and random state are preserved.

Two reference passes per query block bound distance and top-k workspace by order B squared plus B times k, where B is `chunk_size`. Preprocessed and mapped blocks require order B times (p+d), for p features and d components; one-row mapping additionally uses order m plus p plus d, for m prototypes. The existing model, caller inputs, validation/coercion copies, R runtime, and order n-query output are additional allocations. In particular, integer inputs may require full double copies. No complete transformed reference bank is retained. This trades repeated mapping for bounded distance workspace and is intended for small CPU studies.

Value

A list with `per_query`, `summary`, `reference`, `tie_rule`, and `calibrated = FALSE`. The per-query data frame has positional `query_index`, `recall_at_k`, `eligible_reference_count`, `excluded_reference`, and `input_boundary_size`, `embedding_boundary_size`, `input_boundary_weight`, `embedding_boundary_weight`. Boundary size counts observations at the kth distance; weight is their individual inclusion probability. The summary contains the unweighted `mean_recall_at_k`, `n_query`, `n_reference`, `k`, `n_excluded`, and `input_tied_query_count`, `embedding_tied_query_count`, `input_tied_query_fraction`, `embedding_tied_query_fraction` (boundary size greater than one). Empty queries return typed zero-row output, zero counts, and `NA_real_` for the mean and fractions. No inputs, model, or neighbor matrices are retained.

Examples

```
x <- cbind(a = sin(1:30), b = cos(1:30))
model <- soren_fit(x, control = soren_control(iterations = 2L))
query <- cbind(a = sin(31:33), b = cos(31:33))
reference <- cbind(a = sin(34:45), b = cos(34:45))
soren_neighborhood_audit(model, query, reference, k = 3L)$summary
```

soren_save *Save or load a trusted model*

Description

Save or load a trusted model

Usage

```
soren_save(object, file)

soren_load(file)
```

Arguments

object	A fitted model.
file	Local file path. Only load files from trusted sources.

Details

Serialized models include bounded standardized replay and prototypes. These can contain sensitive information and are not anonymized. Validation is not a security sandbox for untrusted R serialization.

Value

soren_save invisibly returns the file path; soren_load returns a validated model.

soren_transform *Map observations with a frozen representation*

Description

Map observations with a frozen representation

Usage

```
soren_transform(object, newdata, diagnostics = FALSE)
```

Arguments

object	A fitted soren_model.
newdata	Dense numeric matrix with the exact training column schema.
diagnostics	Return support diagnostics alongside coordinates?

Details

The normalized radial basis map is deterministic and consumes no random numbers. It lies in the convex hull of prototype coordinates. Far-field behavior is not a reliable extrapolation. No preprocessing is refitted.

Value

A coordinate matrix, or a list with coordinates and support diagnostics.

---

soren_update *Transactionally update a representation*

Description

Transactionally update a representation

Usage

```
soren_update(object, newdata, row_ids = NULL)
```

Arguments

object	A fitted model, which is not modified in place.
newdata	Dense numeric matrix with the original feature schema.
row_ids	Optional identifiers, unique within this batch and distinct from currently retained identifiers. Unretained historical IDs cannot be checked.

Details

Feasibility is checked on old retained anchors after every proposal. Growth proposals that violate the budget are rolled back in full. Same-size proposals are backtracked. A finite-anchor mean budget does not constrain every point, held-out data, or cumulative displacement over many updates.

Value

A new model. Inspect the last history record for accepted, rollback or frozen status, actual displacement and budget.

Index

plot.soren_model *(*print.soren_model*)*, 2
print.soren_model, 2

soren_control, 3
soren_diagnostics, 4
soren_embedding, 5
soren_fit, 5
soren_load *(*soren_save*)*, 8
soren_neighborhood_audit, 6
soren_save, 8
soren_transform, 8
soren_update, 9
summary.soren_model
*(*print.soren_model*)*, 2