

Package: gpfield (via r-universe)

October 10, 2026

Title Change-of-Support Gaussian-Process Regression for Field and Multi-Environment-Trial Data

Version 0.4.1

Description A self-contained, contract-emitting member of the agricultural analytics orchestra for spatial and spatio-temporal Gaussian-process regression over field and multi-environment-trial layouts. Fits an exact Gaussian process to spatial (and optionally temporal) coordinates with a choice of squared-exponential or 'Matern' covariance, estimating the hyperparameters by marginal-likelihood optimisation. Its defining feature is change of support: a prediction can be requested at a support different from the one observed -- a point process predicted as block (paddock-scale) averages, or block averages disaggregated toward points -- with the predictive variance propagated through the support map. Predictions abstain with a typed, machine-readable reason when the estimated spatial range or the data support cannot bear the requested resolution, so the member refuses cleanly rather than returning a confident wrong surface. Results are emitted as a versioned, provenance-complete object compatible with the orchestra ensemble-manifest contract, so a smoothed field surface composes directly with downstream calibration, uncertainty-quantification, causal-inference and decision tools. The package is released under the MIT licence and takes no heavy external Gaussian-process dependency: the exact solver is base R.

License MIT + file LICENSE

URL <https://github.com/max578/gpfield>

BugReports <https://github.com/max578/gpfield/issues>

Depends R (>= 4.1.0)

Imports digest, S7, stats, utils

Suggests agridat, callr, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
Language en-AU
Roxygen list(markdown = TRUE)
Config/roxygen2/version 8.1.0
Repository https://max578.r-universe.dev
Date/Publication 2026-10-09 22:32:20 UTC
RemoteUrl https://github.com/max578/gpfield
RemoteRef main
RemoteSha b6f355aec2b88f3be4b6c12a5102353a3a37fbe4

Contents

as_orchestra_manifest	2
block_support	3
gp_field_plot	4
gp_field_smooth	5
gp_fit	6
gp_predict	7
gpfield_abstention	9
gpfield_fit_class	10
gpfield_prediction_class	11
gpfield_smooth_class	11
gpfield_spec	12
gpfield_spec_class	14
is_gpfield_abstention	14
orchestra_manifest	15
verify_manifest	15
Index	17

as_orchestra_manifest *Emit a gpfield result as an orchestra manifest*

Description

The federation’s emit / adapt generic. The gpfield_prediction method turns a smoothed surface into the shared orchestra_manifest contract: inferential_target = "predictions", the predictive mean and standard deviation (and, for block support, the block bounds) carried in outputs, and the model provenance – kernel, estimated range, support, log-likelihood – in metadata. The payload is integrity-hashed so a downstream consumer can verify it with [verify_manifest\(\)](#).

Usage

```
as_orchestra_manifest(x, ...)
```

Arguments

x A gpfield_prediction object.

... Method-specific arguments. The gpfield_prediction method accepts run_id, an optional character run identifier; when omitted a content hash is derived, so two identical predictions get the same identifier.

Value

An orchestra_manifest S7 object.

See Also

[verify_manifest\(\)](#), [gp_predict\(\)](#)

Examples

```
set.seed(1L)
grid <- expand.grid(x = seq_len(8L), y = seq_len(8L))
grid$yield <- 2 + 0.3 * grid$x + stats::rnorm(nrow(grid), 0, 0.2)
fit <- gp_fit(gpfield_spec(c("x", "y"), "yield"), grid, seed = 1L)
pred <- gp_predict(fit, grid)
as_orchestra_manifest(pred)
```

block_support

Describe a block (areal) support for change-of-support prediction

Description

Defines one or more blocks over which the latent field is to be integrated to a block average – the paddock-scale change-of-support target. Blocks are axis-aligned rectangles (a row of lower / upper bounds per coordinate axis); irregular or polygonal footprints are a planned extension, not supported here. The prediction integrates the GP posterior over each block by deterministic quadrature and propagates the full within-block covariance into the block average's predictive variance.

Usage

```
block_support(lower, upper, ids = NULL, n_quad = 200L)
```

Arguments

lower	Numeric matrix or data frame, one row per block, columns aligned to the fit's spatial coordinate axes, holding each block's lower bounds.
upper	As lower, holding the upper bounds.
ids	Optional character block identifiers (default "block1", ...).
n_quad	Integer target number of quadrature points per block (default 200), realised as the nearest tensor grid of axis midpoints.

Value

A gpfield_block_support classed list consumed by `gp_predict()`.

See Also

`gp_predict()`

Examples

```
blocks <- block_support(
  lower = rbind(c(0, 0), c(4, 4)),
  upper = rbind(c(4, 4), c(8, 8))
)
blocks
```

gp_field_plot

Draw a smoothed field surface (requires ggplot2)

Description

Renders the point-support prediction from `gp_field_smooth()` as a raster of the smoothed trait over the row-column plane. The colour scale is the colourblind-safe viridis option. Requires the optional ggplot2 package and the two coordinate columns to be present in the prediction locations.

Usage

```
gp_field_plot(prediction, row = "row", col = "col")
```

Arguments

prediction	A point-support gpfield_prediction, or the gpfield_smooth object returned by <code>gp_field_smooth()</code> .
row, col	Character scalars naming the coordinate columns in the prediction locations (default "row" and "col").

Value

A ggplot object.

See Also

[gp_field_smooth\(\)](#)

Examples

```
set.seed(1L)
field <- expand.grid(row = seq_len(10L), col = seq_len(10L))
field$yield <- 3 + 0.2 * field$row - 0.1 * field$col +
  stats::rnorm(nrow(field), 0, 0.3)
sm <- gp_field_smooth(field, response = "yield", seed = 1L)
gp_field_plot(sm)
```

gp_field_smooth

Smooth a row-column field or trial layout

Description

One call from a multi-environment-trial (or any row-column field) data frame to a smoothed surface. Fits a gpfield Gaussian process over the row and column coordinates and predicts the latent trait surface back onto a plot grid – the observed grid by default, or a finer grid through refine. The abstention of [gp_predict\(\)](#) is inherited: refining only inserts points strictly between the *observed* row-column locations, so if the observed grid's own spacing is already coarser than the estimated correlation range can resolve, [gp_field_smooth\(\)](#) returns a [gpfield_abstention\(\)](#) at every refine level rather than an over-confident surface – refining a grid the range cannot support does not manufacture support.

Usage

```
gp_field_smooth(
  data,
  response,
  row = "row",
  col = "col",
  kernel = "matern",
  nu = 1.5,
  refine = 1L,
  seed = NULL
)
```

Arguments

data	A data.frame with the row, column and response columns.
response	Character scalar naming the trait column to smooth.
row, col	Character scalars naming the row and column coordinate columns (default "row" and "col").
kernel	Character covariance kernel passed to <code>gpfield_spec()</code> : "matern" (default) or "se".
nu	Matern smoothness (default 1.5); ignored for "se".
refine	Positive integer grid-refinement factor. 1 (default) predicts on the observed plot grid; 2 predicts on a grid twice as fine on each axis, and so on. Refining never itself escapes an abstention: the observed grid's own spacing is checked against the estimated correlation range before any refinement is applied, so an unsupported observed grid stays unsupported at every refine level.
seed	Optional integer seed recorded with the fit.

Value

A `gpfield_smooth` S7 object carrying the `@fit` (a `gpfield_fit`) and the `@prediction` (a `gpfield_prediction`), or a `gpfield_abstention()` when the fit or the prediction cannot be supported.

See Also

`gp_fit()`, `gp_predict()`, `gp_field_plot()`

Examples

```
set.seed(1L)
field <- expand.grid(row = seq_len(10L), col = seq_len(10L))
field$yield <- 3 + 0.2 * field$row - 0.1 * field$col +
  stats::rnorm(nrow(field), 0, 0.3)
sm <- gp_field_smooth(field, response = "yield", seed = 1L)
sm@prediction
```

gp_fit

Fit a gpfield Gaussian process to field or trial data

Description

Solves an exact Gaussian-process regression of the response on the spatial (and optionally temporal) coordinates named by the `gpfield_spec`. The coordinates and response are standardised, the chosen kernel's hyperparameters are estimated by maximising the marginal likelihood, and the solved Cholesky system is stored so that `gp_predict()` can predict cheaply – at the observed support or another.

Usage

```
gp_fit(spec, data, seed = NULL)
```

Arguments

spec	A gpfield_spec from gpfield_spec() .
data	A data.frame carrying the coordinate columns, the optional time column, and the response column named by spec.
seed	Optional integer RNG seed recorded with the fit for provenance. The exact fit is deterministic, so the seed is metadata rather than a source of randomness.

Details

The estimated length-scale is reported in raw coordinate units as the effective (practical) correlation range – the distance at which the covariance has decayed to 0.05, so points beyond it are effectively independent. That range is the quantity the change-of-support and abstention machinery reasons about: a prediction finer than the range is unsupported, and the fit makes that range explicit rather than implicit.

Value

A gpfield_fit S7 object, or a [gpfield_abstention\(\)](#) with reason "degenerate_fit" when the marginal-likelihood optimisation does not yield a usable covariance.

See Also

[gpfield_spec\(\)](#), [gp_predict\(\)](#), [gp_field_smooth\(\)](#)

Examples

```
set.seed(1L)
grid <- expand.grid(x = seq_len(8L), y = seq_len(8L))
grid$yield <- 2 + 0.3 * grid$x + sin(grid$y / 2) + stats::rnorm(nrow(grid), 0, 0.2)
spec <- gpfield_spec(coords = c("x", "y"), response = "yield")
fit <- gp_fit(spec, grid, seed = 1L)
fit
```

gp_predict

Predict a gpfield surface, optionally at a different support

Description

The package's central verb. Given a fitted GP, predicts the latent field at the requested locations and, critically, at the requested support. With support = "point" (default) the result is the GP posterior mean and standard deviation at each query point. With support = "block" the result is the change-of-support prediction: each block's value is the integral of the latent field over the block, and its predictive standard deviation is computed from the full within-block posterior covariance so spatial correlation is propagated, not averaged away.

Usage

```
gp_predict(
  fit,
  newdata = NULL,
  support = c("point", "block"),
  blocks = NULL,
  min_support_ranges = 1,
  spacing_tol = 1,
  seed = NULL
)
```

Arguments

<code>fit</code>	A <code>gpfield_fit</code> from <code>gp_fit()</code> .
<code>newdata</code>	For point support, a <code>data.frame</code> of query coordinates with the fit's coordinate (and time) columns. Ignored when <code>support = "block"</code> .
<code>support</code>	Character: "point" (default) or "block". Block support requires blocks.
<code>blocks</code>	A <code>gpfield_block_support</code> from <code>block_support()</code> , required when <code>support = "block"</code> .
<code>min_support_ranges</code>	Numeric, block support only: a block must have at least one training observation within this many correlation ranges of its footprint, else the verb abstains. Default 1. Ignored (with a warning) at point support.
<code>spacing_tol</code>	Numeric, point support only: the point grid's spacing may be at most this multiple of the estimated range before the verb abstains with "range_too_short". Default 1 (spacing may not exceed the range). Ignored (with a warning) at block support.
<code>seed</code>	Retained for backward compatibility. Block quadrature is now deterministic (a tensor grid, not Monte-Carlo draws), so a block prediction is reproducible without a seed and never perturbs the caller's random stream; this argument no longer affects the result.

Details

Honest abstention runs before any value is returned. If the estimated correlation range is shorter than the requested point spacing – so the query grid asks the GP to interpolate between effectively independent observations – the verb returns a `gpfield_abstention()` with reason "range_too_short". If a requested block has no training observation within one correlation range of its footprint, the verb abstains with reason "support_gap". An empty query abstains with "empty_query". The rule errs toward refusing a confident wrong surface over emitting one.

Value

A `gpfield_prediction` S7 object on success, or a `gpfield_abstention()` when the support cannot bear the request.

See Also

`gp_fit()`, `block_support()`, `gp_field_smooth()`

Examples

```
set.seed(1L)
grid <- expand.grid(x = seq_len(10L), y = seq_len(10L))
grid$yield <- 2 + 0.25 * grid$x + 0.1 * grid$y +
  stats::rnorm(nrow(grid), 0, 0.2)
fit <- gp_fit(gpfield_spec(c("x", "y"), "yield"), grid, seed = 1L)

# Point prediction on a fine grid.
q <- expand.grid(x = seq(1, 10, by = 1), y = seq(1, 10, by = 1))
pt <- gp_predict(fit, q)

# Change of support: two paddock-scale block averages.
bl <- block_support(lower = rbind(c(1, 1), c(6, 6)),
  upper = rbind(c(5, 5), c(10, 10)))
blk <- gp_predict(fit, support = "block", blocks = bl, seed = 1L)
blk
```

gpfield_abstention	<i>Construct a typed gpfield abstention</i>
--------------------	---

Description

Returns a small classed object signalling that the data cannot support a prediction, with a machine-readable reason code and human-readable detail. Used in place of a confident-but-wrong prediction so that callers can branch on a refusal rather than trust an unsupported surface.

Usage

```
gpfield_abstention(
  reason,
  detail = NA_character_,
  scope = "gp_predict",
  diagnostics = list()
)
```

Arguments

reason	Character scalar reason code. One of "range_too_short" (the estimated correlation range is shorter than the requested prediction spacing, so interpolation would be unsupported extrapolation between effectively independent points), "support_gap" (a requested block has too few – or no – observations within its own footprint or one correlation range of it), "degenerate_fit" (the marginal-likelihood optimisation did not yield a usable hyperparameter set), or "empty_query" (the prediction grid contained no rows).
--------	---

detail	Character scalar human-readable detail, or NA.
scope	Character scalar naming what was refused (default "gp_predict").
diagnostics	Optional named list of the numbers behind the refusal (for example the estimated range and the requested spacing) so the caller can act on the margin, not just the verdict.

Value

An object of class "gpfield_abstention": a list with reason, detail, scope, diagnostics and abstained = TRUE.

Examples

```
a <- gpfield_abstention(
  "range_too_short",
  "estimated range 2.1 m < requested spacing 10 m"
)
is_gpfield_abstention(a)
```

gpfield_fit_class	<i>A fitted gpfield Gaussian process</i>
-------------------	--

Description

The result of `gp_fit()`: the originating gpfield_spec, the standardised training coordinates and response, the estimated hyperparameters (marginal variance, length-scale and noise variance), the Cholesky factor and weight vector of the solved system, and the marginal log-likelihood at the optimum. Predict from it – at the observed support or another – with `gp_predict()`.

Value

An S7 object of class gpfield_fit.

See Also

`gp_fit()`, `gp_predict()`

`gpfield_prediction_class`*A gpfield prediction at a requested support*

Description

The result of a successful `gp_predict()`: the queried locations or block definitions, the predictive mean and standard deviation, the inferential support ("point" or "block"), and the number of observations that informed each query (carried so a consumer can see how well-supported each value is). Unsuccessful predictions return a `gpfield_abstention()` instead.

Value

An S7 object of class `gpfield_prediction`.

See Also

`gp_predict()`

`gpfield_smooth_class` *A fitted-and-smoothed field*

Description

The successful result of `gp_field_smooth()`: the fitted `gpfield_fit` and the point-support `gpfield_prediction` over the (possibly refined) plot grid, carried together as one typed object so the smoothing path returns the same kind of inspectable value as the rest of the package rather than a bare list. Access the parts with `@fit` and `@prediction`. An unsuccessful smoothing returns a `gpfield_abstention()` instead.

Value

An S7 object of class `gpfield_smooth`.

See Also

`gp_field_smooth()`, `gp_field_plot()`

gpfield_spec

*Specify a gpfield Gaussian-process model***Description**

Builds the configuration consumed by `gp_fit()`. A specification fixes the covariance kernel and (for "matern") its smoothness, names the spatial coordinate columns in the data frame the fit will read, optionally names a time column for a spatio-temporal model, and sets the diagonal jitter used to condition the kernel. Nothing is fitted here; the specification is a small, inspectable object so the model is declared before any data are touched.

Usage

```
gpfield_spec(
  coords,
  response,
  kernel = "matern",
  nu = 1.5,
  time = NA_character_,
  nugget = 1e-06,
  anisotropic = FALSE,
  solver = c("exact", "lowrank"),
  n_inducing = 200L
)
```

Arguments

coords	Character vector naming the spatial coordinate columns (one for a transect, two for a field, three or more if the caller has projected additional spatial axes). When a time column is also given, it is appended to the coordinate set as an extra, separately-scaled axis.
response	Character scalar naming the response column to smooth.
kernel	Character covariance kernel: "matern" (default) or "se" (squared-exponential).
nu	Matern smoothness; one of 0.5, 1.5 (default) or 2.5. Ignored for the squared-exponential kernel.
time	Optional character scalar naming a time column for a spatio-temporal fit, or NA (default) for a purely spatial model. When a time axis is present the fit is anisotropic by necessity – a unit of space and a unit of time are not exchangeable, so time is given its own length-scale even if anisotropic is left at its default FALSE.
nugget	Positive numeric jitter added to the kernel diagonal for numerical conditioning. Distinct from the estimated observation-noise variance, which is a fitted hyper-parameter.

anisotropic	Logical. If TRUE and there are at least two coordinate axes, fit one length-scale per axis (automatic relevance determination) rather than a single isotropic length-scale. The kernel stays isotropic in a per-axis-rescaled coordinate space, so each axis gets its own correlation range – the right model when a field varies on different scales along different directions (an east-west tillage gradient versus a north-south slope, or space versus a time axis). Default FALSE (a single length-scale, the original behaviour).
solver	Character: "exact" (default) fits the exact GP by a full Cholesky factorisation, cubic in the number of observations. "lowrank" fits a Deterministic Training Conditional (DTC) sparse GP on n_{inducing} space-filling inducing points – cost $O(n * n_{\text{inducing}}^2)$ – for fields too large for the exact solver. The low-rank solver is isotropic (a single length-scale) and falls back to the exact solver when the data have no more rows than n_{inducing} .
n_{inducing}	Integer number of inducing points for solver = "lowrank" (default 200, minimum 3); ignored for the exact solver.

Details

The default kernel is the "matern" family at $\nu = 1.5$, the usual choice for field data: it is once-differentiable rather than the squared-exponential's infinite smoothness, which avoids the over-smoothing that the squared exponential imposes across real agronomic gradients.

Value

A gpfield_spec S7 object.

See Also

[gp_fit\(\)](#), [gp_predict\(\)](#)

Examples

```
spec <- gpfield_spec(coords = c("x", "y"), response = "yield")
spec

# An anisotropic field: a separate length-scale per axis.
aniso <- gpfield_spec(c("x", "y"), "yield", anisotropic = TRUE)
aniso

# A large field: the low-rank sparse solver on 150 inducing points.
big <- gpfield_spec(c("x", "y"), "yield", solver = "lowrank",
                   n_inducing = 150L)
big

# A spatio-temporal specification with the squared-exponential kernel.
st <- gpfield_spec(
  coords = c("x", "y"), response = "ndvi", kernel = "se", time = "doy"
)
st
```

gpfield_spec_class	<i>A gpfield model specification</i>
--------------------	--------------------------------------

Description

The configuration of a spatial (or spatio-temporal) Gaussian process before it is fitted: which covariance kernel to use, the smoothness for the Matern family, the names of the spatial coordinate columns, an optional time-axis column, and the jitter added to the kernel diagonal for numerical conditioning. Construct one with `gpfield_spec()` and fit it with `gp_fit()`.

Value

An S7 object of class `gpfield_spec`.

See Also

`gpfield_spec()`, `gp_fit()`

is_gpfield_abstention	<i>Is an object a gpfield abstention?</i>
-----------------------	---

Description

Is an object a gpfield abstention?

Usage

```
is_gpfield_abstention(x)
```

Arguments

x	Any object.
---	-------------

Value

A single logical: TRUE when x is a gpfield_abstention.

Examples

```
is_gpfield_abstention(gpfield_abstention("support_gap"))
is_gpfield_abstention(42)
```

orchestra_manifest	<i>The orchestra ensemble-manifest contract (gpfield-side implementation)</i>
--------------------	---

Description

A versioned, hashed, provenance-complete S7 result object, property-compatible with the federation's reference orchestra_manifest. A gpfield_prediction is adapted into one through [as_orchestra_manifest\(\)](#): the predictive mean and standard deviation ride in outputs, the queried locations and the model provenance ride in metadata, and the payload is integrity-hashed so a tampered surface is detected downstream.

Value

An S7 object of class orchestra_manifest.

See Also

[as_orchestra_manifest\(\)](#), [verify_manifest\(\)](#)

verify_manifest	<i>Verify an orchestra manifest's payload integrity</i>
-----------------	---

Description

Recomputes the payload hash from the object's own load-bearing slots and compares it to the stored data_hash. A mismatch means the payload was modified after emission. Matches the reference contract's verify_manifest().

Usage

```
verify_manifest(m)
```

Arguments

m	An orchestra_manifest object.
---	-------------------------------

Value

A list with logical ok and a human-readable message.

See Also

[as_orchestra_manifest\(\)](#)

Examples

```
set.seed(1L)
grid <- expand.grid(x = seq_len(8L), y = seq_len(8L))
grid$yield <- 2 + 0.3 * grid$x + stats::rnorm(nrow(grid), 0, 0.2)
fit <- gp_fit(gpfield_spec(c("x", "y"), "yield"), grid, seed = 1L)
pred <- gp_predict(fit, grid)
m <- as_orchestra_manifest(pred)
verify_manifest(m)
```

Index

`as_orchestra_manifest`, [2](#)
`as_orchestra_manifest()`, [15](#)

`block_support`, [3](#)
`block_support()`, [8](#), [9](#)

`gp_field_plot`, [4](#)
`gp_field_plot()`, [6](#), [11](#)
`gp_field_smooth`, [5](#)
`gp_field_smooth()`, [4](#), [5](#), [7](#), [9](#), [11](#)
`gp_fit`, [6](#)
`gp_fit()`, [6](#), [8–10](#), [12–14](#)
`gp_predict`, [7](#)
`gp_predict()`, [3–7](#), [10](#), [11](#), [13](#)
`gpfield_abstention`, [9](#)
`gpfield_abstention()`, [5–8](#), [11](#)
`gpfield_fit_class`, [10](#)
`gpfield_prediction_class`, [11](#)
`gpfield_smooth_class`, [11](#)
`gpfield_spec`, [12](#)
`gpfield_spec()`, [6](#), [7](#), [14](#)
`gpfield_spec_class`, [14](#)

`is_gpfield_abstention`, [14](#)

`orchestra_manifest`, [15](#)

`verify_manifest`, [15](#)
`verify_manifest()`, [2](#), [3](#), [15](#)