New Metrics for Assessing Projection Pursuit Indexes, and Guiding Optimisation Choices

H. Sherry Zhang[1], Dianne Cook[2], Nicolas Langrené[3], Jessica Wai Yin Leung[2]

ARTICLE HISTORY
Compiled October 14, 2024

[1] Department of Statistics and Data Sciences, University of Texas at Austin, Austin, United States
[2] Department of Econometrics and Business Statistics, Monash University, Melbourne, Australia
[3] Department of Mathematical Sciences, Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College, Zhuhai, China

ABSTRACT
The projection pursuit (PP) guided tour interactively optimises a criterion function known as the PP index, to explore high-dimensional data by revealing interesting projections. Optimisation of some PP indexes can be non-trivial, if they are non-smooth functions, or the optimum has a small "squint angle", detectable only from close proximity. To address these challenges, this study investigates the performance of a recently introduced swarm-based algorithm, Jellyfish Search Optimiser (JSO), for optimising PP indexes. The performance of JSO for visualising data is evaluated across various hyper-parameter settings and compared with existing optimisers. Additionally, methods for calculating the smoothness and squintability properties of the PP index are proposed. They are used to assess the optimiser performance in the presence of PP index complexities. A simulation study illustrates the use of these performance metrics to compare the JSO with existing optimisation methods available for the guided tour. The JSO algorithm has been implemented in the R package, `tourr`, and functions to calculate smoothness and squintability are available in the `ferrn` package.

KEYWORDS
projection pursuit; jellyfish search optimiser (JSO); optimisation; grand tour; high-dimensional data; exploratory data analysis

CONTACT: H. Sherry Zhang. Email: huize.zhang@austin.utexas.edu. Dianne Cook. Email: dicook@monash.edu. Nicolas Langrené. Email: nicolaslangrene@uic.edu.cn. Jessica Wai Yin Leung. Email: Jessica.Leung@monash.edu.

1. Introduction

Projection pursuit (PP) (Kruskal (1969), Friedman and Tukey (1974), Huber (1985)) is a dimension reduction technique aimed at identifying informative linear projections of data. This is useful for exploring high-dimensional data, and creating plots of the data that reveal the main features to use for publication. The method involves optimising an objective function known as the PP index (e.g. Hall (1989), Cook, Buja, and Cabrera (1993), Lee and Cook (2010), Loperfido (2018), Loperfido (2020)), which defines the criterion for what constitutes interesting or informative projections. Let $X \in \mathbb{R}^{n\times p}$ be the data matrix, $A \in \mathbb{R}^{p\times d}$ be an orthonormal matrix, where A belongs to the Stiefel manifold $\mathcal{A} = V_d(\mathbb{R}^p)$. The projection $Y = XA$ is a linear transformation that maps data from a p-dimensional space into a d-dimensional space. The index function $f(XA) : \mathbb{R}^{n\times d} \to \mathbb{R}$ is a scalar function that measures an interesting aspect of the projected data, such as deviation from normality, presence of clusters, or non-linear structure. For a fixed sample of data, PP finds the orthonormal basis A that maximises the index value of the projection, $Y = XA$:

$$\max_{A\in\mathcal{A}} \quad f(XA) \quad \text{subject to} \quad A'A = I_d \tag{1}$$

It is interesting to note that when using PP visually, one cares less about A than the plane described by A, because the orientation in the plane is irrelevant. The space of planes belongs to a Grassmann manifold. This is usually how the projection pursuit guided tour (PPGT) (Cook et al. 1995) operates, when using geodesic interpolation between starting and target planes. It interpolates plane to plane, removing irrelevant

within plane spin, and is agnostic to the basis (A) used to define the plane. Thus, indexes which are used for the PPGT should be rotationally invariant.

Index functions are quite varied in form, partially depending on the data that is being projected. Figure 1 shows two examples. Huber plots (Huber 1990) of 2D data sets are in (a) and (c), showing the PP index values for all 1D projections of the 2D data in polar coordinates, which reveals the form of these functions. The dashed circle is a baseline set at the average value, and the straight line marks the optimal projection. Plots (b) and (d) show the respective best projections of the data as histograms. Indexes like the holes, central mass and skewness (Cook, Buja, and Cabrera 1993) are generally smooth for most data sets, but capture only large patterns. Many indexes are noisy and non-convex, requiring an effective and efficient optimisation procedure to explore the data landscape and achieve a globally optimal viewpoint of the data. The skewness index computed for trimodal data, in (a), is smooth with a large squint angle but has three modes, and thus is not convex. The binned normality index (a simple version of a non-normality index as described in Huber (1985)) computed on the famous RANDU data, in (c), is noisier and has a very small squint angle. The discreteness cannot be seen unless the optimiser is very close to the optimal projection.

Optimisation of PP is often discussed when new indexes are proposed (Posse 1995; Marie-Sainte, Berro, and Ruiz-Gazen 2010; Grochowski and Duch 2011). Cook et al. (1995) tied the optimisation more closely to the index, when they introduced the PPGT, which monitors the optimisation visually so that the user can see the projected data leading in and out of the optimum. An implementation is available in the `tourr` package (Wickham et al. 2011) in R (R Core Team 2023). Zhang et al. (2021) illustrated how to diagnose optimisation processes, particularly focusing on the guided tour, and revealed a need for improved optimisation. While improving the quality of the optimisation solutions

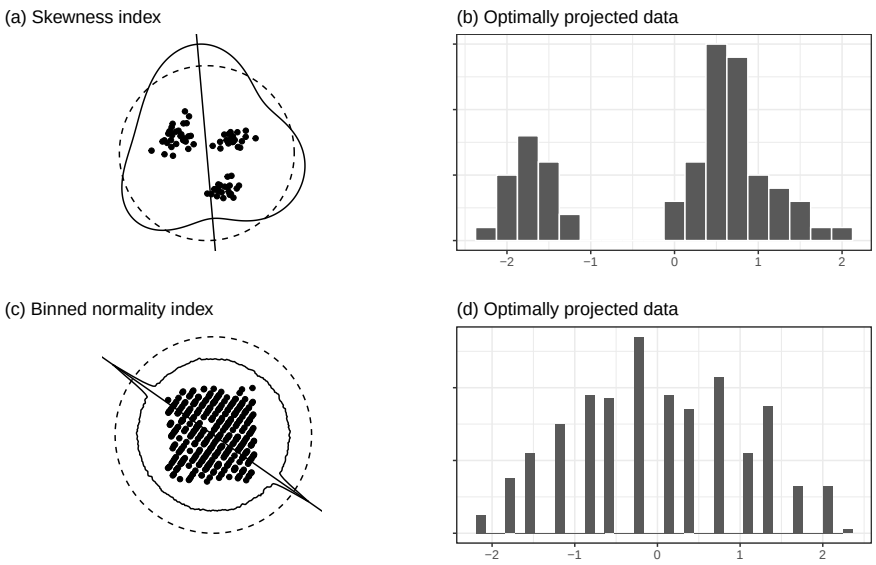

Figure 1. Examples of PP indexes with large (top row) and small (bottom row) squint angles, shown with a Huber plot, and histogram of the projected data corresponding to the optimal projection. A Huber plot shows the PP index values for all 1D data projections in polar coordinates.

in the tour is essential, it is also important to be able to view the data projections as the optimisation progresses. Integrating the guided tour with a global optimisation algorithm that is efficient in finding the global optimal and enables viewing of the projected data during the exploration process is a goal.

Here, the potential for a Jellyfish Search Optimiser (JSO) (see Chou and Truong (2021), and Rajwar, Deep, and Das (2023)) for the PPGT is explored. JSO, inspired by the search behaviour of jellyfish in the ocean, is a swarm-based metaheuristic designed to solve global optimisation problems. Compared to traditional methods, JSO has demonstrated stronger search ability and faster convergence, and requires fewer tuning parameters. These practicalities make JSO a promising candidate for enhancing PP optimisation.

The primary goal of the study reported here is to investigate the performance of JSO in PP optimisation for the guided tour. It is of interest to assess how quickly and closely the optimiser reaches a global optimum, for various PP indexes that may have

differing complexities. To observe the performance of JSO with different types of PP indexes, metrics are introduced to capture specific properties of the index including squintability (based on Tukey and Tukey (1981)'s squint angle) and smoothness. Here, we mathematically define metrics for squintability and smoothness, which is a new contribution for PP research. A series of simulation experiments using various datasets and PP indexes are conducted to assess JSO's behaviour and its sensitivity to hyper-parameter choices (number of jellyfish and maximum number of tries). The relationship between the JSO performance, hyper-parameter choices and properties of PP indexes (smoothness and squintability) is analysed to provide guidance on selecting optimisers for practitioners using projection pursuit. Additionally, this work should guide the design of new PP indexes and facilitate better optimization for PP.

The paper is structured as follows. Section 2 introduces the background of the PPGT, reviews existing optimisers and index functions in the literature. Section 3 introduces the metrics that measure two properties of PP indexes, smoothness and squintability, and Section 4 describes the new JSO to be used for the PPGT. Section 5 outlines two simulation experiments to assess JSO's performance: one comparing JSO's performance improvements relative to an existing optimiser, Creeping Random Search (CRS), and the other studying the impact of PP index properties on optimisation performance, and Section 6 presents the results. Section 7 discusses the implementation of JSO in the `tourr` package and the PP property calculation in the `ferrn` package. Section 8 summarises the work and provides suggestions for future directions.

2. Projection pursuit, tours, index functions and optimisation

A tour on high-dimensional data is constructed by geodesically interpolating between pairs of planes. Any plane is described by an orthonormal basis, A_t, where t represents time in the sequence. The term "geodesic" refers to maintaining the orthonormality constraint so that each view shown is correctly a projection of the data. The PP guided tour operates by geodesically interpolating to target planes (projections) which have high PP index values, as provided by the optimiser. The geodesic interpolation means that the viewer sees a continuous sequence of projections of the data, so they can watch patterns of interest forming as the function is optimised. There are five optimisation methods implemented in the `tourr` package:

- a pseudo-derivative, that searches locally for the best direction, based on differencing the index values for very close projections.
- a brute-force optimisation (CRS).
- a modified brute force algorithm described in Posse (1995).
- an essentially simulated annealing (Bertsimas and Tsitsiklis 1993) where the search space is reduced during the optimisation.
- a very localised search, to take tiny steps to get closer to the local maximum.

There are numerous PP index functions available: introduced in Huber (1985), Cook, Buja, and Cabrera (1993), Lee et al. (2005), Lee and Cook (2010), Grimm (2016), Ursula Laa and Valencia (2022). Most are relatively simply defined, for any projection dimension, and implemented because they are relatively easy to optimise. A goal is to develop PP indexes based on scagnostics (L. Wilkinson, Anand, and Grossman (2005), Leland Wilkinson and Wills (2008)), but the blockage is their optimisation as these

tend to be noisy, with potentially small squint angles.

An initial investigation of PP indexes, and the potential for scagnostics is described in Laa and Cook (2020). To be useful here an optimiser needs to be able to handle index functions that are possibly not very smooth. In addition, because data structures might be relatively fine, the optimiser needs to be able to find maxima that occur with a small squint angle, that can only be seen from very close by. One last aspect that is useful is for an optimiser to return local maxima in addition to the global one because data can contain many different and interesting features.

3. Properties of PP indexes

Laa and Cook (2020) has proposed five criteria for assessing projection pursuit indexes (smoothness, squintability, flexibility, rotation invariance, and speed). Since not all index properties affect the optimisation process, the focus here is on the first two properties, *smoothness* (Section 3.1) and *squintability* (Section 3.2), for which metrics are proposed to quantify them.

3.1. *Smoothness*

This subsection proposes a metric for the smoothness of a projection pursuit index.

A classical way to describe the smoothness of a function is to identify how many continuous derivatives of the function exist. This can be characterized by Sobolev spaces (Adams and Fournier 2003).

Definition 1 (sobolev space)**.** *The Sobolev space* $W^{k,p}(\mathbb{R})$ *for* $1 \leq p \leq \infty$ *is the set of all functions* f *in* $L^p(\mathbb{R})$ *for which all weak derivatives* $f^{(\ell)}$ *of order* $\ell \leq k$ *exist and*

have a finite L^p norm.

The Sobolev index k in Definition 1 can be used to characterize the smoothness of a function: if $f \in W^{k,p}$, then the higher k, the smoother f. While this Sobolev index k is a useful measure of smoothness, it can be difficult to compute or even estimate in practice.

To obtain a computable estimator of the smoothness of the index function f, we propose an approach based on random fields. If a PP index function f is evaluated at some random bases, as is done at the initialization stage of JSO, then these random index values can be interpreted as a random field, indexed by a space parameter, namely the random projection basis. This analogy suggests to use this random training sample to fit a spatial model. We propose to use a Gaussian process equipped with a Matérn covariance function, due to the connections between this model and Sobolev spaces, see for example Porcu et al. (2024).

The distribution of a Gaussian process is fully determined by its mean and covariance function. The smoothness property comes into play in the definition of the covariance function: if a PP index is very smooth, then two close projection bases should produce close index values (strong correlation); by contrast, if a PP index is not very smooth, then two close projection bases might give very different index values (fast decay of correlations with respect to distance between bases). Popular covariance functions are parametric positive semi-definite functions. In particular, the Matérn class of covariance functions has a dedicated parameter to capture the smoothness of the Gaussian field.

Definition 2 (Matérn covariance function)**.** *The Matérn covariance function K is*

defined by

$$K(u) = K_{\nu,\eta,\ell}(u) := \eta^2 \frac{\left(\sqrt{2\nu}\frac{\|u\|}{\ell}\right)^{\nu}}{\Gamma(\nu)2^{\nu-1}} \mathcal{K}_{\nu}\left(\sqrt{2\nu}\frac{\|u\|}{\ell}\right) , \tag{2}$$

where $\|u\|$ *is the Euclidean norm of* $u \in \mathbb{R}^{p\times d}$, $\nu > 0$ *is the smoothness parameter,* η *is the outputscale,* ℓ *is the lengthscale, and* $\mathcal{K}_{\nu}$ *is the modified Bessel function [DLMF 10.25].*

The Matérn covariance function can be expressed analytically when ν is a half-integer, the most popular values in the literature being $\frac{1}{2}$, $\frac{3}{2}$ and $\frac{5}{2}$ (Rasmussen and Williams 2006). The parameter ν, called *smoothness parameter*, controls the decay of the covariance function. As such, it is an appropriate measure of smoothness of a random field, as shown by the simulations on Figure 2 and Figure 3. For example, Karvonen (2023) showed that if a function f has a Sobolev index of k, then the smoothness parameter estimate ν in (2) cannot be asymptotically less than k. See the survey Porcu et al. (2024) for additional results on the connection between the Matérn model and Sobolev spaces. An interesting result is that the asymptotic case $\nu \to \infty$ coincides with the Gaussian kernel: $K_{\infty}(u) = \exp(-\|u\|^2/2)$.

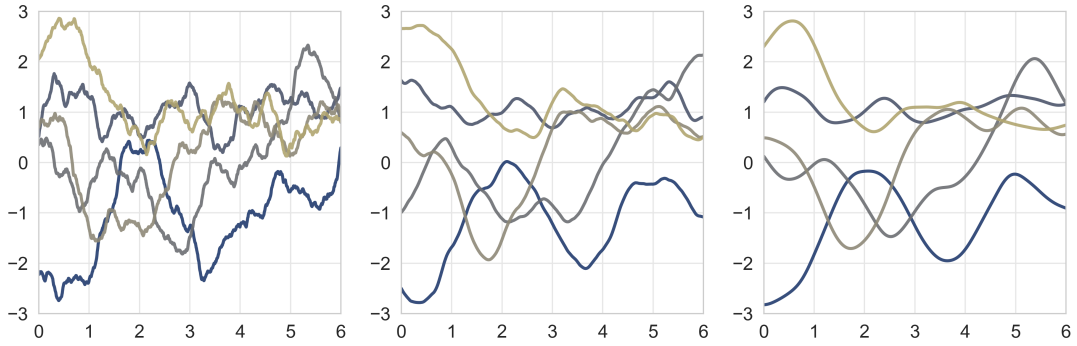

Figure 2. Five random simulations from a Gaussian Process defined on $\mathbb{R}$ with zero mean and Matérn-ν covariance function, with $\nu = 1$ (left), $\nu = 2$ (middle), and $\nu = 4$ (right), showing that higher values of ν produce smoother curves.

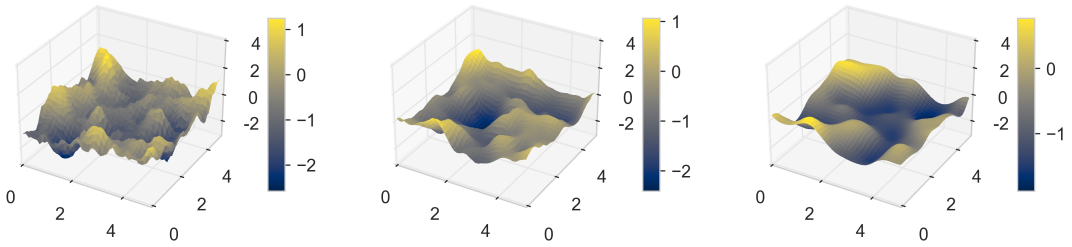

Figure 3. One random simulation from a Gaussian Process defined on $\mathbb{R}^2$ with zero mean and Matérn-ν covariance function, with $\nu = 1$ (left), $\nu = 2$ (middle), and $\nu = 4$ (right), showing that higher values of ν produce smoother surfaces.

In view of these results, the parameter ν is suggested as a measure of the smoothness of the PP index function by fitting a Gaussian process prior with Matérn covariance on a dataset generated by random evaluations of the index function, as done at the initialization stage of the jellyfish search optimization. There exist several R packages, such as `GpGp` (Guinness, Katzfuss, and Fahmy 2021) or `ExaGeoStatR` (Abdulah et al. 2023), to fit the hyper-parameters of a GP covariance function on data, which is usually done by maximum likelihood estimation. In this project, the `GpGp` package is used.

Definition 3. *Let* $\mathbf{A} = [A_1, \dots, A_N] \in (\mathbb{R}^{p\times d})^N$ *be d-dimensional projection bases, let* $\mathbf{y} = [f(XA_1), \dots, f(XA_N)]$ *be the corresponding PP index values, and let* $\mathbf{K} = [K_\theta(A_i - A_j)]_{1\le i,j\le N} \in \mathbb{R}^{N\times N}$ *be the Matérn covariance matrix evaluated at the input bases, where the vector* θ *contains all the parameters of the multivariate Matérn covariance function* K *(smoothness, outputscale, lengthscales). The log-likelihood of the parameters* θ *is defined by*

$$\mathcal{L}(\theta) = \log p(\mathbf{y}\,|\mathbf{A}, \theta) = -\frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log(\det(\mathbf{K} + \sigma^2\mathbf{I})) - \frac{N}{2}\log(2\pi) \quad (3)$$

where the nugget parameter σ *is the standard deviation of the intrinsic noise of the Gaussian process. The optimal parameters (including smoothness) are obtained by maximum*

log-likelihood

$$\theta^* = \max_{\theta} \mathcal{L}(\theta) \tag{4}$$

The resulting optimal smoothness parameter ν is chosen as our smoothness metric.

The value of the optimal smoothness parameter $\nu > 0$ can be naturally interpreted as follows: the higher ν, the smoother the index function.

3.2. *Squintability*

This section defines the projection distance and introduces the squintability metric, followed by a description of two numerical approaches for computing squintability.

Definition 4 (projection distance)**.** *Let $A \in \mathbb{R}^{p\times d}$ is a d-dimensional orthonormal matrix, and let A^* be the optimal matrix that achieves the maximum index value for a given data. The projection distance between A and A^*, $r(A, A^*)$, is defined by $r(A, A^*) = \|AA' - A^*A^{*\prime}\|_F$ where $\|.\|_F$ denotes the Frobenius norm, given by $\|M\|_F = \sqrt{\sum_{ij} M_{ij}^2}$.*

Definition 5 (squint angle)**.** *Let A and B be two d-dimensional orthonormal matrices in $\mathbb{R}^p$. The squint angle θ between the subspace spanned by A and B is defined as the smallest principal angle between these subspaces: $\theta = \min_{i\in\{1,\cdots,d\}} \arccos(\tau_i)$, where τ_i are the singular values of the matrix $M = A^T B$ obtained from its singular value decomposition.*

Squintability can be defined as how the index value $f(XA)$ changes with respect to the projection distance $r(A, A^*)$, over the course of the JSO:

Definition 6 (squintability)**.** *Let $g : \mathbb{R} \mapsto \mathbb{R}$ be a decreasing function that maps the projection distance $r(A, A^*)$ to the index value $f(XA)$, such that $g(r) = g(r(A, A^*)) = f(XA)$. The squintability of an index function f is defined by*

$$\varsigma(f) = -4 \times \min_r g'(r) \times \arg\min_r g'(r) \tag{5}$$

where 4 is a constant scaling factor, $-\min_r g'(r)$ represents the largest gradient of $-g$ and $\arg\min_r g'(r)$ represents the projection distance at which this largest gradient is attained. If the set $\mathcal{D} := \arg\min_r g'(r)$ contains more than one value r, the largest value $\max(\mathcal{D})$ is used in equation (5).

It is expected that these two values should be both high in the case of high squintability (fast increase in g early on), and both low in the case of low squintability (any substantial increase in g happens very late, close to the optimal angle). This suggests that their product (5) should provide a sensible measure of squintability. The multiplicative constant 4, which can be deemed arbitrary, does not change the interpretation of the squintability metric ς; it is here to adjust the range of values of ς and simplify the explicit formula for ς obtained later on.

From the definition, the following proposition can be derived:

Proposition 1. *Let g_1 be a convex, strictly decreasing function and g_2 be a concave, strictly decreasing function, both defined on $[0, R]$ where $R > 0$, and taking values in $[0, 1]$. Then*

$$\varsigma(g_1) < \varsigma(g_2)$$

Proof. Let $r_1 := \arg\min_r g_1'(r)$ and $r_2 := \arg\min_r g_2'(r)$. Since g_1 is convex, $g_1''(r) \geq 0 \ \forall r \in [0, R]$, meaning that g_1' is an increasing function, taking its minimum for $r_1 = 0$. As a result, $\varsigma(g_1) = -4 \times g'(0) \times 0 = 0$. Since g_2 is concave, $g_2''(r) \leq 0 \ \forall r \in [0, R]$, meaning that g_2' is a decreasing function, taking its minimum for $r_2 = D > 0$. Since g_2 is strictly decreasing, $g_2'(r) < 0 \ \forall r \in [0, R]$. In particular $g_2'(R) < 0$. As a result $\varsigma(g_2) = -4 \times g'(R) \times R > 0$. This proves that $\varsigma(g_1) < \varsigma(g_2)$. □

From Tukey and Tukey (1981) and Laa and Cook (2020), a large squint angle implies that the objective function value is close to optimal even when the perfect view to see the structure is far away. A small squint angle means that the PP index value improves substantially only when the perfect view is close by. As such, low squintability implies rapid improvement in the index value when near the perfect view. For PP, a small squint angle is considered to be undesirable because it means that the optimiser needs to be very close to be able to "see" the optimum. Thus, it could be difficult for the optimiser to find the optimum.

It is expected that for a PP index with high squint angle, the optimization (1) should make substantial progress early on. Conversely, for a PP index with low squint angle, it might take a long while for the optimization to make substantial progress, as the candidate projections would need to be very close to the optimal one for the structure of the index function to be visible enough to be amenable to efficient optimization. This observation suggests that the extreme values of g' and the projection distances for which these values are attained, are crucial in the mathematical definition of squintability.

To compute the squintability metric (5) in practice, several approaches are possible. The first one is to propose a parametric model for g, and use it to obtain an explicit formula

for ς. Numerical experiments suggest a scaled sigmoid shape as described below. Define

$$\ell(x) := \frac{1}{1 + \exp(\theta_3(x - \theta_2))} \ , \tag{6}$$

which is a decreasing logistic function depending on two parameters θ_2 and θ_3, such that $\ell(\theta_2) = \frac{1}{2}$. Then, define

$$g(x) = (\theta_1 - \theta_4)\frac{\ell(x) - \ell(x_{\max})}{\ell(0) - \ell(x_{\max})} + \theta_4 \ , \tag{7}$$

which depends on three additional parameters, θ_1, θ_2, and $x_{\max}$, such that $g(0) = \theta_1$ and $g(x_{\max}) = \theta_4$. Under the parametric model (7), the squintability metric (5) can be shown to be equal to

$$\varsigma = \frac{(\theta_1 - \theta_4)\theta_2\theta_3}{\ell(0) - \ell(x_{\max})} \ . \tag{8}$$

In practice, the parameters of this model (7) can be estimated numerically, for example by non-linear least squares, and then used to evaluate ς as in equation (8).

Alternatively, one can estimate (5) in a nonparametric way, for example by fitting g using kernel regression, then numerically estimate the angle at which $-g'$ attains its highest value.

4. The jellyfish optimiser

The Jellyfish Search Optimiser (JSO) mimics the natural movements of jellyfish, which include passive and active motions driven by ocean currents and their swimming patterns, respectively. In the context of optimization, these movements are abstracted to explore the search space, aiming to balance exploration (searching new areas) and exploitation (focusing on promising areas). The algorithm aims to find the optimal solution by adapting the behaviour of jellyfish to navigate towards the best solution over iterations (Chou and Truong 2021).

To solve the optimisation problem embedded in the PP guided tour, a starting projection, an index function, the number of jellyfish, and the maximum number of trials (tries) are provided as input. Then, the current projection is evaluated by the index function. The projection is then moved in a direction determined by a random factor, influenced by how far along we are in the optimisation process. Occasionally, completely new directions may be taken like a jellyfish might with ocean currents. A new projection is accepted if it is an improvement compared to the current one, rejected otherwise. This process continues and iteratively improves the projection, until the pre-specified maximum number of trials is reached.

Algorithm: Jellyfish Optimizer Pseudo Code

Input: `current_projections`, `index_function`, `trial_id`, `max_trial`

Output: `optimised_projection`

Initialize `current_best` as the projection with the best index value from `current_projections`, and `current_idx` as the array of index values for each projection in `current_projections`

for each `trial_id` in 1 to `max_tries` **do**

Calculate the time control value, c_t, based on `current_idx` and `max_trial`

if c_t is greater than or equal to 0.5 **then**

Define trend based on the `current_best` and `current_projections`

Update each projection towards the trend using a random factor and orthonormalisation

else

if a random number is greater than $1 - c_t$ **then**

Slightly adjust each projection with a small random factor (passive)

else

For each projection, compare with a random jellyfish and adjust towards or away from it (active)

Update the orientation of each projection to maintain consistency

Evaluate the new projections using the index function

if any new projection is worse than the current, revert to the `current_projections` for that case

Determine the projection with the best index value as the new `current_best`

if `trial_id` $\geq$ `max_trial`, print the last best projection **exit**

return the set of projections with the updated `current_best` as the `optimised_projection`

The JSO implementation involves several key parameters that control its search process in optimization problems. These parameters are designed to guide the exploration and

exploitation phases of the algorithm. While the specific implementation details can vary depending on the version of the algorithm or its application, the focus is on two main parameters that are most relevant to our application: the number of jellyfish and the maximum number of tries.

5. Assessing optimisers: two simulation studies

This section describes the details of two simulation studies: one compares the JSO performance with an existing optimiser, Creeping Random Search (CRS) (Zhang et al. 2021; Laa and Cook 2020) to explore JSO's behaviour under different data dimensions. The second simulation first examines the JSO performance under different hyper-parameters combinations (number of jellyfish and the maximum number of tries), and then studies the effect of index properties (smoothness and squintability), along with JSO hyper-parameters, and data dimension, on the JSO performance through a logistic regression.

The performance of JSO is measured by the success rate, defined as the proportion of simulations that achieves a final index value within 0.05 of the best index value found among all 50 simulations (see Figure 4 for an illustration). This comparison is based on projection pursuit to find the pipe shape investigated by Laa and Cook (2020) using the `holes` index.

5.1. *Performance of JSO relative to CRS*

The JSO performance is investigated both in comparison to the existing optimizer, CRS, in a pipe-finding problem using the holes index in the 6, 8, 10, and 12-dimensional space. Fifty simulations are conducted in four data dimensions ($d = 6, 8, 10, 12$) with each optimiser. JSO uses 100 jellyfish with a maximum of 100 tries, while the CRS allows

a maximum of 1000 samples at each iteration before the algorithm terminates. The different numbers account for the multiple paths of JSO to enable fairer comparison with CRS. The results of the simulation are collected using the data structure proposed in Zhang et al. (2021) for assessing JSO, where the design parameters are stored along with index value, projection basis, random seed, and computation time.

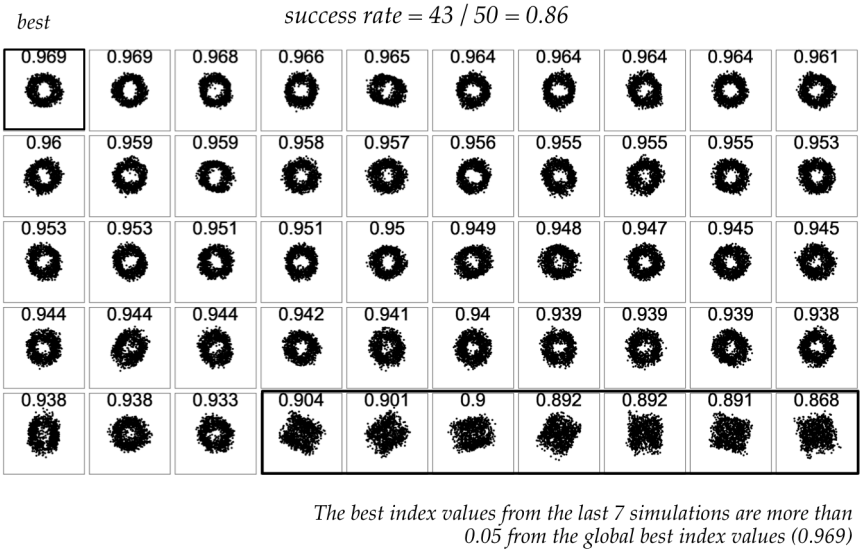

Figure 4. Illustration of success rate calculation: Final projections based on projection pursuit to find the pipe shape in 8D data using the holes index, optimised by CRS, in 50 simulations. The 50 final projections are sorted by their index values. The highest index value found across all simulations is 0.969. Out of the 50 simulations, 43 achieved an index value within 0.05 of the best, resulting in a success rate of 0.86 (43/50).

5.2. *Factors affecting JSO success rate: JSO hyper-parameters and index properties*

To examine JSO's performance across various hyper-parameter combinations, the pipe-finding problem using the holes index is repeated for fifty times in each hyper-parameter combination: 20, 50, and 100 jellyfish and a maximum of 50 and 100 attempts. The success rate is calculated for each combination.

To assess JSO's performance across various PP problems, two different data shapes,

pipe and a sine wave, are investigated in 6D and 8D spaces using six different PP indexes: `dcor2d_2`, `loess2d`, `MIC`, `TIC`, `spline`, and `stringy`, under varied JSO hyper-parameters. This results in a total of 52 scenarios, comprising of 24 computed on the pipe data and 28 on the sine-wave data. Again, JSO is run 50 times for each scenario to calculate the success rate for each PP problem.

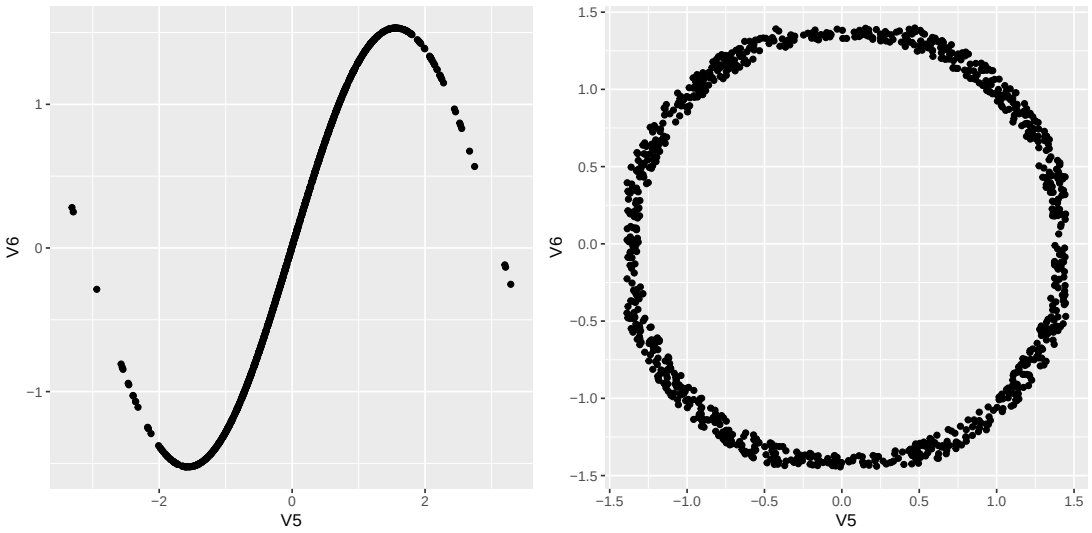

Figure 5. The sine and pipe structures in the example data, along with index values for these projections.

Smoothness and squintability are computed following the procedures outlined in Section 3.1 and Section 3.2 and as illustrated in Figure 6 and Figure 7. To compute smoothness, 500 random bases are simulated. Index values are calculated for each random basis, followed by fitting a Gaussian process model to obtain the smoothness measure for the index.

To compute squintability, 50 random bases are sampled and interpolated to the optimal basis with a step size of 0.005. Index values and projection distances are calculated for these interpolated bases. A binning procedure is applied to average the index values within each 0.005 projection distance bin. The four-parameter scaled logistic function (7) is fitted to the index values against projection distances, estimated by non-linear

least squares to obtain the squintability measure as Equation 8.

To construct a relationship among success rate, smoothness, squintability, and JSO hyper-parameters, a generalised linear model is fitted using a binomial family and a logit link function. The data is pre-processed by 1) scaling the JSO hyper-parameters by a factor of 10 for interpretation, 2) creating a new binary variable `long_time` to indicate cases with an average run time over 20 seconds, and 3) re-coding the success rate for the `stringy` index as 0, because none of the 50 simulations correctly identified the sine-wave shape.

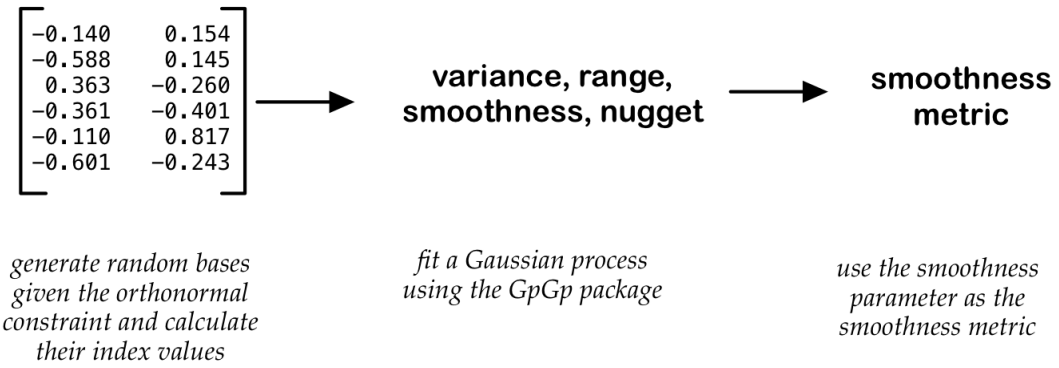

Figure 6. Steps for calculating smoothness in a projection pursuit problem. Given the target shape, data dimension and the index function: 1) sample random bases given the orthonormality contraint and calculate their corresponding index values, and 2) fit a Gaussian process model of index values against the sampled bases to obtain the smoothness measure.

6. Results

6.1. *Performance of JSO relative to CRS*

Performance of JSO is assessed based on the final projections across the two optimisers: JSO and CRS. The final projections found by the two optimisers are presented in Figure 8, broken down by 10th quantile, faceted by the data dimensions. In the 6-dimensional data scenario, JSO consistently identifies a clear pipe shape. The CRS also

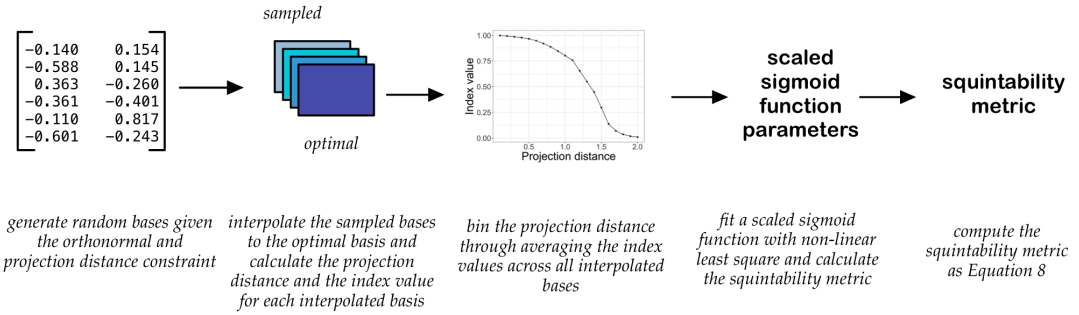

Figure 7. Steps for calculating squintability in a projection pursuit problem. Given the target shape, data dimension and the index function: 1) sample random bases given the orthonormality and projection distance contraint and calculates their corresponding index values, 2) interpolate the sampled bases to the optimal basis and calculate the projection distance and the index value. 3) bin the index values by projection distances through averaging index value, 4) fit the scaled sigmoid function in equation (5) to the binned index values against projection distances using non-linear least square to obtian the squintability measure using equation (8).

finds the pipe shape but with a wide rim, suggesting a further polish search may be required. With increasing dimensions, JSO may not always identify the pipe shape due to random sampling, but it still finds the pipe shape in over 50% of cases. Compared to CRS, JSO achieves higher index values and clearer pipe shapes across all quantiles in data of 8, 10, 12 dimensions, suggesting its advantage in exploring high-dimensional spaces.

6.2. *Factors affecting JSO success rate: JSO hyper-parameters and index properties*

The effect of JSO hyper–parameters (number of jellyfish and the maximum number of tries) on the success rate is presented in Figure 9. The uncertainty is quantified through 500 bootstrap replicates for each case. As the number of jellyfish and maximum tries increase, the success rate also increases. For simpler problems (6 dimensions), small parameter values (20 jellyfish and a maximum of 50 tries) can already achieve a high success rate. However, larger parameter values (i.e. 100 jellyfish and a maximum of

Table 1. Parameters estimated from the Gaussian process (outputscale η, lengthscale ℓ, smoothness ν, and nugget σ) and scaled logistic function (θ_1 to θ_4 and ς) for the pipe-finding and sine-wave finding problems. The columns ν and ς represent the smoothness and squintability measures respectively.

shape	index	d	η	ℓ	ν	σ	θ_1	θ_2	θ_3	θ_4	ς
pipe	holes	6	0.03	0.39	3.08	0.0	1.02	0.86	3.37	0.02	3.05
pipe	holes	8	0.02	0.32	2.86	0.0	1.01	0.87	3.26	0.03	2.96
pipe	holes	10	0.02	0.36	2.80	0.0	1.02	0.88	3.15	0.02	2.95
pipe	holes	12	0.02	0.36	2.82	0.0	1.01	0.88	3.34	0.00	3.12
sine	MIC	6	0.06	0.29	3.15	0.0	0.89	0.57	1.62	-0.02	1.26
sine	MIC	8	0.06	0.33	3.09	0.0	0.93	0.33	1.31	-0.03	0.77
sine	TIC	6	0.17	0.34	3.06	0.0	0.95	0.54	1.72	-0.03	1.32
sine	TIC	8	0.16	0.31	3.17	0.0	0.95	0.56	1.72	-0.03	1.37
sine	dcor	6	0.09	0.31	3.06	0.0	0.95	1.04	2.74	-0.02	2.95
sine	loess	6	0.24	0.45	3.53	0.0	1.02	1.04	2.65	0.08	2.76
sine	splines	6	0.10	0.32	3.11	0.0	1.01	1.05	2.73	-0.01	3.12
sine	stringy	6	0.00	1128.28	1.03	134.8	1.05	0.01	254.73	0.05	2.96

100 tries) are necessary for higher-dimensional problems (8, 10, and 12 dimensions). Increasing both parameters enhances the performance of JSO, but it also extends the computational time required for the optimisation, which can be computationally intensive when evaluating the index function (such as scagnostic indexes) multiple times across numerous iterations.

Smoothness and squintability are calculated across a collection of pipe-finding and sine-wave finding problems to construct the relationship between success rate, JSO hyper-parameters, and index properties. Table 1 presents the parameters estimated from the Gaussian process (outputscale η, lengthscale ℓ, smoothness ν, and nugget σ) and from the scaled logistic function (θ_1 to θ_4) for calculating smoothness and squintability in each PP problem considered. The column ν is used as the smoothness metric and the column ς is calculated as equation (8) as the squintability metric.

Table 2 presents the results from fitting a logistic regression model using the proportion of success as the response and smoothness, squintability, dimension, running time, number of jellyfish and maximum number of tries, as predictors. Long runtime is a binary

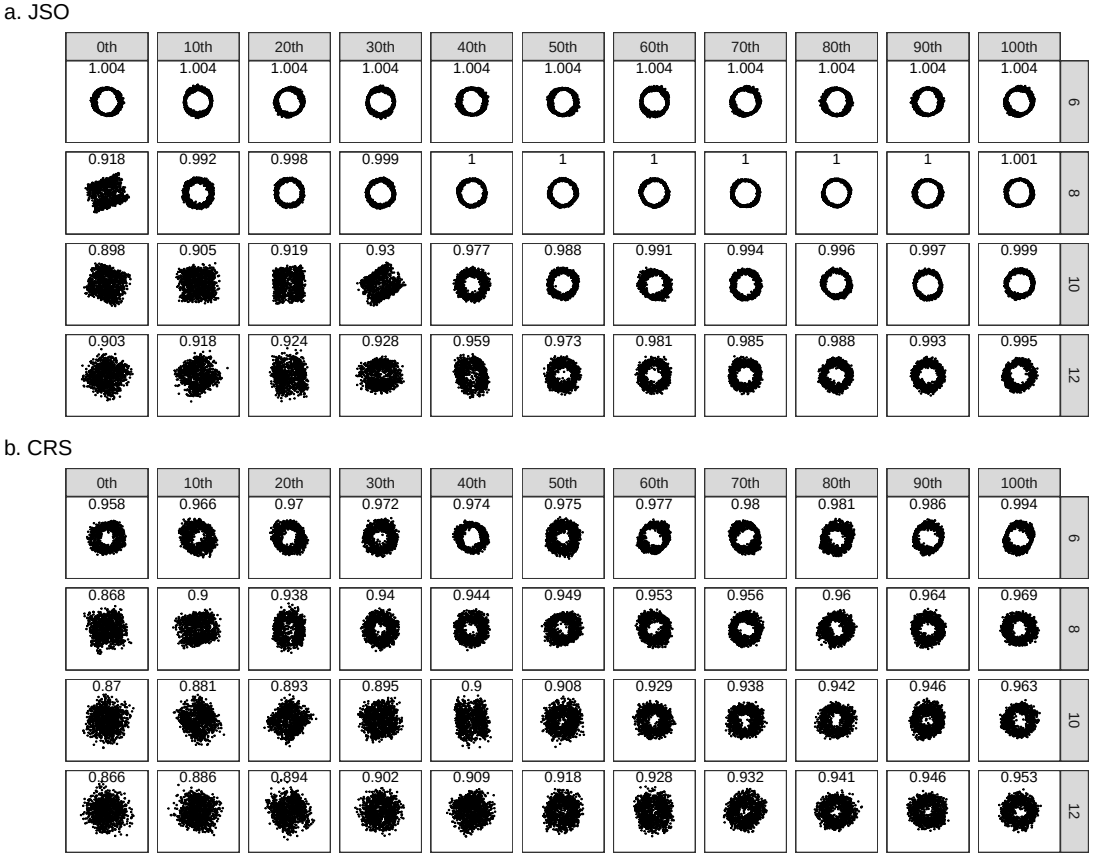

Figure 8. Projections found by the JSO and CRS at each 10th quantile across 50 simulations. The projection pursuit problem is to find the pipe shape using the holes index in the 6, 8, 10, and 12-dimensional spaces. The JSO uses 100 jellyfish and a maximum number of tries of 100. The CRS uses a maximum of 1000 tries in each step of random sampling step before the algorithm terminates. In the 6-D data space, JSO always finds a clear pipe shape while the CRS also finds the pipe shape but with a wide rim. At higher data dimensions, JSO finds a higher index value and a clearer pipe shape across all the quantiles than the CRS

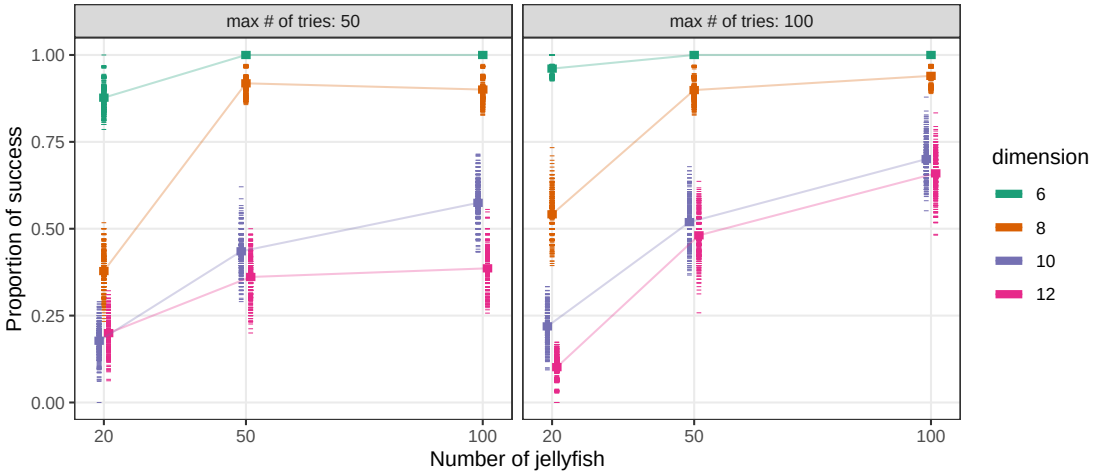

Figure 9. The proportion of simulations that reach near-optimal index values in the pipe-finding problem using the holes index. The proportion is calculated based on the number of simulations, out of 50, that achieve an index value within 0.05 of the best-performing simulation. To quantify uncertainty, 500 bootstrap samples are generated. The thin segments represent the proportion for each of the 500 bootstrap replicates, while the thicker segments represent the mean of these bootstrap samples, connnected by lines. As the dimensionality increases, the proportion of simulations reaching the optimal index value decreases.

variable, where greater than 20 seconds is labelled 1. The fit suggests that JSO success rate is only affected by squintability, dimension and number of jellies. Interestingly, the smoothness of the index is not a problem for the JSO - this is promising! Running the optimisation longer and allowing more tries also does not affect success rate. An increase in squintability of one unit increases the success rate by 1235%. As dimension increases by one the success rate halves. Increasing the number of jellies by 10, increases the succes rate by 27%.

Table 2. Jellyfish success rate relative to index properties and jellyfish hyper-parameters. This is the summary from a logistic regression fit to smoothness, squintability, dimension, running time, number of jellyfish and maximum number of tries. Interestingly, squintability and dimension strongly affect jellyfish optimisation success. The number of jellies marginally affects success, but index smoothness, running longer and increasing the number of tries do not.

Characteristic	**OR (95% CI)**[1]	**p-value**
Smoothness	89.1 (1.23 to 4,865,159)	0.289

Squintability	12.4 (3.08 to 125)	**0.006**
Dimension	0.57 (0.30 to 0.96)	**0.048**
Long runtime	0.17 (0.00 to 4.96)	0.357
Number of jellyfish	1.27 (0.99 to 1.68)	**0.069**
Maximum number of tries	1.10 (0.82 to 1.52)	0.524

[1]OR = Odds Ratio, CI = Confidence Interval

7. Practicalities

The simulation studies have compared the JSO with CRS and shown how smoothness and squintability affect the JSO success rate. Here we explain how they can be used for new index and optimiser development. Using the JSO optimiser for PPGT requires a change in user behaviour. For all the existing optimisers a single optimisation path is followed. The JSO has multiple optimisation paths, and needs additional tools to incorporate into the PPGT, as explained here.

7.1. *Using the JSO in a PPGT*

To use JSO for PPGT in the `tourr` package, specify `search_f = search_jellyfish` in the guided tour. Unlike existing optimisers, the animation function `animate_*()` won't directly render the tour path for JSO due to the generation of multiple tour paths. To visualise the path visited by each individual jellyfish, assign the animation to an object (`res <- animate_xy(...)`). This will save the bases visited by JSO, along with relevant metadata, as a tibble data object (see Zhang et al. (2021) for more details on the data object). The bases visited for individual jellyfish can then be extracted and

viewed using the `planned_tour()` function.

7.2. *Computing the index properties for your new index*

The `ferrn` package (Zhang et al. 2021) provides functionality for computing the smoothness and squintability metrics. Both metrics require sampling random bases using the `sample_bases()` function, followed by the metric calculation with `calc_smoothness()` or `calc_squintability()`.

7.2.1. Smoothness

To sample bases for calculating smoothness, the following arguments are required: the index function, the dataset, and the number of random bases to sample. The output of `sample_bases()` is a data frame with a list-column of sampled basis matrix and index value. Parallellasation is available to speed up the index value computation through the `parallel = TRUE` argument.

The `calc_smoothness()` function takes the output from `sample_bases()` and fits a Gaussian process model to the index values against the sampled basis, as the location, to obtain the smoothness metric. Starting parameters and additional Gaussian process arguments can be specified and details of the fit can be accessed through the `fit_res` attribute of the output.

7.2.2. Squintability

Bases sampling for squintability includes an additional step of interpolating between the sampled bases and the best basis. This step is performed when the arguments `step_size` and `min_proj_dist` in the `sample_bases()` function are set to non-NA

numerical values. Given the projection distance typically ranging from 0 to 2, it is recommended to set `step_size` to 0.1 or lower, and `min_proj_dist` to be at least 0.5 to ensure a meaningful interpolation length.

The `calc_squintability()` function computes squintability using two methods: 1) parametrically, by fitting a scaled sigmoid function through non-linear least square (`method = "nls"`), and 2) non-parametrically, using kernel smoothing (`method = "ks"`). A `bin_width` argument is required to average the index values over the projection distance before the fit. For the parametric case, the output provides the estimated parameters for the scaled sigmoid function (θ_1 to θ_4) and the calculated squintability metric as Equation 8. For the non-parametric case, it shows the maximum gradient attained (`max_d`), the corresponding projection distance (`max_dist`), and the squintability metric as their products.

8. Conclusion

This paper has presented new metrics to mathematically define desirable features of PP indexes, squintability and smoothness, and used these to assess the performance of the new jellyfish search optimiser. The metrics will be generally useful for characterising PP indexes, and help with developing new indexes.

In the comparison of the JSO against the currently used CRS, as expected the JSO vastly outperforms CRS, and provides a high probability of finding the global optimum. The JSO obtains the maximum more cleanly, with a slightly higher index value, and plot of the projected data showing the structure more clearly.

The JSO performance is affected by the hyper-parameters, with a higher chance of reaching the global optimum when more jellyfish are used and the maximum number of tries

is increased. However, it comes at a computational cost, as expected. The performance declines if the projection dimension increases and if the PP index has low squintability. The higher the squintability the better chance the JSO can find the optimum. However, interestingly smoothness does not affect the JSO performance.

One future direction of this work is to test the JSO, along with its variations and other swarm-based optimisers, on a broader range of indexes, for example, scagnostic indexes.

9. Acknowledgement

The article is created using Quarto (Allaire et al. 2022) in R (R Core Team 2023). The source code for reproducing the work reported in this paper can be found at: https://github.com/huizezhang-sherry/paper-jso. The simulation data produced in Section 5 can be found at https://figshare.com/articles/dataset/Simulated_raw_data/26039506. Nicolas Langrené acknowledges the partial support of the Guangdong Provincial Key Laboratory IRADS (2022B1212010006, R0400001-22) and the UIC Start-up Research Fund UICR0700041-22.

Supplementary materials

The supplementary materials available at https://github.com/huizezhang-sherry/paper-jso include: 1) details of the indexes used in the simulation study, 2) the script to get started with using JSO in a PPGT and calculating smoothness and squintability, as explained in Section 7, and (3) the full code to reproduce the plots and summaries in this paper.

References

Abdulah, Sameh, Yuxiao Li, Jian Cao, Hatem Ltaief, David Keyes, Marc Genton, and Ying Sun. 2023. "Large-Scale Environmental Data Science with ExaGeoStatR." *Environmetrics* 34 (1): e2770. https://doi.org/10.1002/env.2770.

Adams, Robert, and John Fournier. 2003. *Sobolev Spaces*. Vol. 140. Pure and Applied Mathematics. Elsevier.

Allaire, J. J., Charles Teague, Carlos Scheidegger, Yihui Xie, and Christophe Dervieux. 2022. *Quarto* (version 1.2). https://doi.org/10.5281/zenodo.5960048.

Bertsimas, Dimitris, and John Tsitsiklis. 1993. "Simulated Annealing." *Statistical Science* 8 (1): 10–15. https://doi.org/10.1214/ss/1177011077.

Chou, Jui-Sheng, and Dinh-Nhat Truong. 2021. "A Novel Metaheuristic Optimizer Inspired by Behavior of Jellyfish in Ocean." *Applied Mathematics and Computation* 389 (January): 125535. https://doi.org/10.1016/j.amc.2020.125535.

Cook, D., A. Buja, and J. Cabrera. 1993. "Projection Pursuit Indexes Based on Orthonormal Function Expansions." *Journal of Computational and Graphical Statistics* 2 (3): 225–50. https://doi.org/10.2307/1390644.

Cook, D., A. Buja, J. Cabrera, and C. Hurley. 1995. "Grand Tour and Projection Pursuit." *Journal of Computational and Graphical Statistics* 4 (3): 155–72. https://doi.org/10.1080/10618600.1995.10474674.

DLMF. 2024. "NIST Digital Library of Mathematical Functions." https://dlmf.nist.gov/10.25.

Friedman, J. H., and J. W. Tukey. 1974. "A Projection Pursuit Algorithm for Exploratory Data Analysis." *IEEE Transactions on Computers* C-23 (9): 881–90. https://doi.org/10.1109/T-C.1974.224051.

Grimm, Katrin. 2016. "Kennzahlenbasierte Grafikauswahl." Doctoral thesis, Universität Augsburg.

Grochowski, Marek, and Włodzisław Duch. 2011. "Fast Projection Pursuit Based on Quality of Projected Clusters." In *Adaptive and Natural Computing Algorithms*, edited by Andrej Dobnikar, Uroš Lotrič, and Branko Šter, 89–97. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20267-4_10.

Guinness, Joseph, Matthias Katzfuss, and Youssef Fahmy. 2021. "GpGp: Fast Gaussian Process Computation Using Vecchia's Approximation." R package.

Hall, Peter. 1989. "On Polynomial-Based Projection Indices for Exploratory Projection Pursuit." *The Annals of Statistics* 17 (2): 589–605. https://doi.org/10.1214/aos/1176347127.

Huber, Peter J. 1985. "Projection Pursuit." *The Annals of Statistics* 13 (2): 435–75. https://doi.org/10.1214/aos/1176349519.

———. 1990. "Data Analysis and Projection Pursuit." Technical Report PJH-90-1. Dept. of Mathematics, Massachusetts Institute of Technology.

Karvonen, Toni. 2023. "Asymptotic Bounds for Smoothness Parameter Estimates in Gaussian Process Interpolation." *SIAM/ASA Journal on Uncertainty Quantification* 11 (4): 1225–57. https://doi.org/10.1137/22M149288X.

Kruskal, J. B. 1969. "Toward a Practical Method Which Helps Uncover the Structure of a Set of Observations by Finding the Line Transformation Which Optimizes a New 'Index of Condensation'." In *Statistical Computation*, edited by R. C. Milton and J. A. Nelder, 427–40. New York: Academic Press. https://doi.org/10.1016/B978-0-12-498150-8.50024-0.

Laa, Ursula, and Dianne Cook. 2020. "Using Tours to Visually Investigate Properties of New Projection Pursuit Indexes with Application to Problems in Physics." *Compu-*

tational Statistics 35 (3): 1171–1205. https://doi.org/10.1007/s00180-020-00954-8.

Lee, Eun-Kyung, and Dianne Cook. 2010. "A Projection Pursuit Index for Large p Small n Data." *Statistics and Computing* 20 (3): 381–92. https://doi.org/10.1007/s11222-009-9131-1.

Lee, Eun-Kyung, Dianne Cook, Sigbert Klinke, and Thomas Lumley. 2005. "Projection Pursuit for Exploratory Supervised Classification." *Journal of Computational and Graphical Statistics* 14 (4): 831–46. https://doi.org/10.1198/106186005X77702.

Loperfido, Nicola. 2018. "Skewness-Based Projection Pursuit: A Computational Approach." *Computational Statistics and Data Analysis* 120 (C): 42–57. https://doi.org/10.1016/j.csda.2017.11.001.

———. 2020. "Kurtosis-Based Projection Pursuit for Outlier Detection in Financial Time Series." *The European Journal of Finance* 26 (2-3): 142–64. https://doi.org/10.1080/1351847X.2019.1647864.

Marie-Sainte, Souad Larabi, Alain Berro, and Anne Ruiz-Gazen. 2010. "An Efficient Optimization Method for Revealing Local Optima of Projection Pursuit Indices." In *Swarm Intelligence*, edited by Marco Dorigo, Mauro Birattari, Gianni A. Di Caro, René Doursat, Andries P Engelbrecht, Dario Floreano, Luca Maria Gambardella, et al., 60–71. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15461-4_6.

Porcu, Emilio, Moreno Bevilacqua, Robert Schaback, and Chris Oates. 2024. "The Matérn Model: A Journey Through Statistics, Numerical Analysis and Machine Learning." *Statistical Science* 39 (3): 469–92. https://doi.org/10.1214/24-STS923.

Posse, Christian. 1995. "Projection Pursuit Exploratory Data Analysis." *Computational Statistics and Data Analysis* 20 (6): 669–87. https://doi.org/10.1016/0167-9473(95)00002-8.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rajwar, Kanchan, Kusum Deep, and Swagatam Das. 2023. "An Exhaustive Review of the Metaheuristic Algorithms for Search and Optimization: Taxonomy, Applications, and Open Challenges." *Artificial Intelligence Review*, 1–71. https://doi.org/10.1007/s10462-023-10470-y.

Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

Tukey, P. A., and J. W. Tukey. 1981. *Graphical Display of Data in Three and Higher Dimensions*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley. https://books.google.com.au/books?id=WBzvAAAAMAAJ.

Ursula Laa, Andreas Buja, Dianne Cook, and German Valencia. 2022. "Hole or Grain? A Section Pursuit Index for Finding Hidden Structure in Multiple Dimensions." *Journal of Computational and Graphical Statistics* 31 (3): 739–52. https://doi.org/10.1080/10618600.2022.2035230.

Wickham, H., D. Cook, H. Hofmann, and A. Buja. 2011. "tourr: An R Package for Exploring Multivariate Data with Projections." *Journal of Statistical Software* 40 (2): 1–18. https://doi.org/10.18637/jss.v040.i02.

Wilkinson, L., A. Anand, and R. Grossman. 2005. "Graph-Theoretic Scagnostics." In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 157–64. https://doi.org/10.1109/INFVIS.2005.1532142.

Wilkinson, Leland, and Graham Wills. 2008. "Scagnostics Distributions." *Journal of Computational and Graphical Statistics* 17 (2): 473–91. https://doi.org/10.1198/106186008X320465.

Zhang, H. Sherry, Dianne Cook, Ursula Laa, Nicolas Langrené, and Patricia Menéndez. 2021. "Visual Diagnostics for Constrained Optimisation with Application to Guided Tours." *The R Journal* 13: 624–41. https://doi.org/10.32614/RJ-2021-105.