Studying the Performance of the Jellyfish Optimiser for the Application of Projection Pursuit

Alice Anonymous[a,*], Bob Security[b], Cat Memes[b], Derek Zoolander

[a] *Some Institute of Technology, Department Name, Street Address, City, Postal Code*

[b] *Another University, Department Name, Street Address, City, Postal Code*

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum augue turpis, dictum non malesuada a, volutpat eget velit. Nam placerat turpis purus, eu tristique ex tincidunt et. Mauris sed augue eget turpis ultrices tincidunt. Sed et mi in leo porta egestas. Aliquam non laoreet velit. Nunc quis ex vitae eros aliquet auctor nec ac libero. Duis laoreet sapien eu mi luctus, in bibendum leo molestie. Sed hendrerit diam diam, ac dapibus nisl volutpat vitae. Aliquam bibendum varius libero, eu efficitur justo rutrum at. Sed at tempus elit.

Keywords: projection pursuit, optimization, jellyfish optimiser, data visualisation, high-dimensional data

Let's use British English ("American or British usage is accepted, but not a mixture of these")

1. Introduction [Nicolas and Jessica]

The artificial jellyfish search (JS) algorithm [1] is a swarm-based metaheuristic optimisation algorithm inspired by the search behaviour of jellyfish in the ocean. It is one of the newest swarm intelligence algorithms [2], which was shown to have stronger search ability and faster convergence with few algorithmic parameters compared to classic optimization methods [1]-[3].

Effective optimisation is an important aspect of many methods employed for visualising high-dimensional data (X). Here we are concerned about computing informative linear projections of high-dimensional (p) data using projection pursuit (PP) (Kruskal [4], Friedman and Tukey [5]). This involves optimising a function (e.g. Hall [6], Cook et al. [7], Lee and Cook [8], Loperfido [9], Loperfido [10]), called the projection pursuit index (PPI), that defines what is interesting or informative in a projection.

These PPI are defined on projections (XA), which means that there is a constraint that needs to be considered when optimising. A projection of data is defined by a $p \times d$ orthonormal matrix A, and this imposes the constraint on the elements of A, that columns need have norm equal to 1 and the product of columns need to sum to zero.

Cook et al. [11] introduced the PP guided tour, which enabled interactive visualisation of the optimisation in order to visually explore high-dimensional data. It is implemented in the R [XXX REF] package `tourr` [12]. The optimisation that is implemented is fairly basic, and potential problems were highlighted by Zhang

[*] Corresponding author

Email addresses: `alice@example.com` (Alice Anonymous), `bob@example.com` (Bob Security), `cat@example.com` (Cat Memes), `derek@example.com` (Derek Zoolander) et al. [13]. Implementing better optimisation functionality is a goal, but it needs to be kept in mind that the guided tour also has places importance on watching the projected data as the optimisation progresses.

Here we explore the potential for a jellyfish optimisation to be integrated with the guided tour. Section 2 explains the optimisation that is used in the current the projection pursuit guided tour. Section 3 provides more details on the jellyfish optimiser and formalises several characteristics of projection pursuit indexes that are help to measure optimisaer performance. Section 4 describes a simulation study on performance of the jellyfish for several types of data and index functions. Section 5 summarises the work and provides suggestions for future directions.

2. Projection pursuit, index functions and optimisation [Di and Sherry]

3. The jellyfish optimiser and properties of PP indexes [Nicolas and Jessica]

The jellyfish 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 in a way that balances exploration (searching new areas) and exploitation (focusing on promising areas). The algorithm aims to find the optimal solution by adapting the jellyfish's behavior to navigate towards the best solution over iterations [1].

Below is the pseudo-code for this visualisation application.

[Put the pseudo-code in, add specifics to this visualisation application]

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, we focus on two main parameters that are most relevant to our application: the number of jellyfish and drift.

Laa and Cook [14] has proposed five criteria for assessing projection pursuit indexes (smoothness, squintability, flexibility, rotation invariance, and speed). Since not all the properties affects the execution of the optimisation, here we consider the three relevant properties (smoothness, squintability, and speed), and propose three metrics to evaluate these three properties.

3.1. Smoothness

An intuitive way to measure smoothness of a function would be to count how many continuous derivatives exist. To help define smoothness in our context, we can make use of the Sobolev spaces: functions f in the Sobolev space $W^{p,\infty}$ have all derivatives of order less than p continuous. Smoothness would then be the highest p such that the index function belongs to $W^{p,\infty}$. We can relax this and consider $W^{p,q}$ Sobolev spaces.

Consider the following definition of partial derivative. Let U be an open subset of $\mathbf{R}^n$ and $f : U \to \mathbf{R}$. The partial derivative of function f at the point $\mathbf{a} = (a_1, \dots, a_n) \in U$ with respective to the i-th variable x_i is defined as

$$\frac{\delta}{\delta x_i} f(\mathbf{a}) = \lim_{h \to 0} \frac{f(a_1, \dots, a_{i-1}, a_i + h, a_{i+1}, \dots, a_n) - f(a_1, \dots, a_i, \dots, a_n)}{h} = \lim_{h \to 0} \frac{f(\mathbf{a} + h\mathbf{e_i}) - f(\mathbf{a})}{h}.$$

where $\mathbf{e_i}$ is the unit vector of the i-th variable x_i.

If the derivative is not well-defined, we propose to approximate the derivative using the following expression:

$$\frac{1}{n_i}\sum_i \frac{\|f(\mathbf{a}+h_i\mathbf{e_i})-f(\mathbf{a})\|^p}{h_i}$$

where $p \in [0,1]$. The choice of p reflects the penalty behaviour. For the application in this paper, we choose $p = 1$.

- h_i is an neighbourhood area around a. We can insert different values of h to approximate the above quantity and observe how this quantity changes.

3.2. Squintability

From the literature, it is commonly understood that a large squint angle implies that the function is easy to optimize, because we do not need to be very close to the perfect view to see the structure. A small squint angle means that the derivative of the index function can still be very large values near the optimal point and will rapidly change as we get even closer to the optimum. As such we can observe the second order gradient, which is the rate of change of gradient, over the space we are searching. For some index functions, the second order gradient is not well-defined, we can approximate the second order gradient vector in similar fashion as the above section.

To the best of our knowledge, this is the first attempt to measure the notion of squintability.

3.3. Speed

The speed of optimizing an index function can be calculated/measured using the computational complexity (in big O notation, with respect to the sample size) of computing the index function.

4. Application [Di and Sherry]

The jellyfish optimiser has been implemented in the tourr package [15] and we will use the diagnostic plots proposed in the ferrn package [13] to visualise the optimisation process.

4.1. Going beyond 10D

The pipe-finding problem is initially used to investigate indexes and optimisers in Laa and Cook [14], and we extend it from a 6D problem to a 12D problem.

Jellyfish optimiser, as a multi-start algorithm, is efficient in [...] for high-dimensional problems

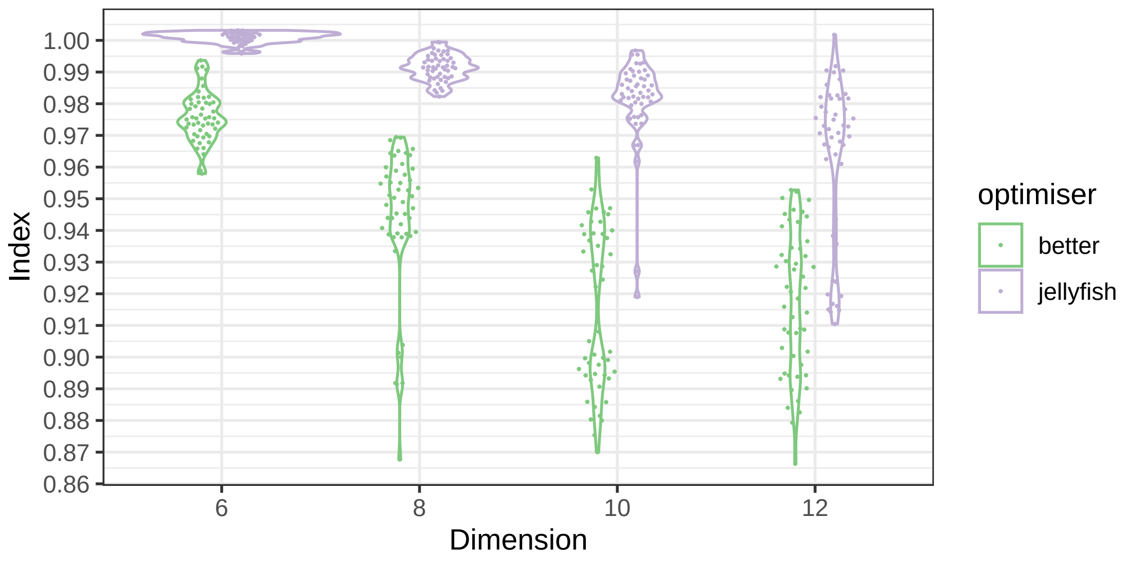

Figure 1: sthis sdfaksdlf

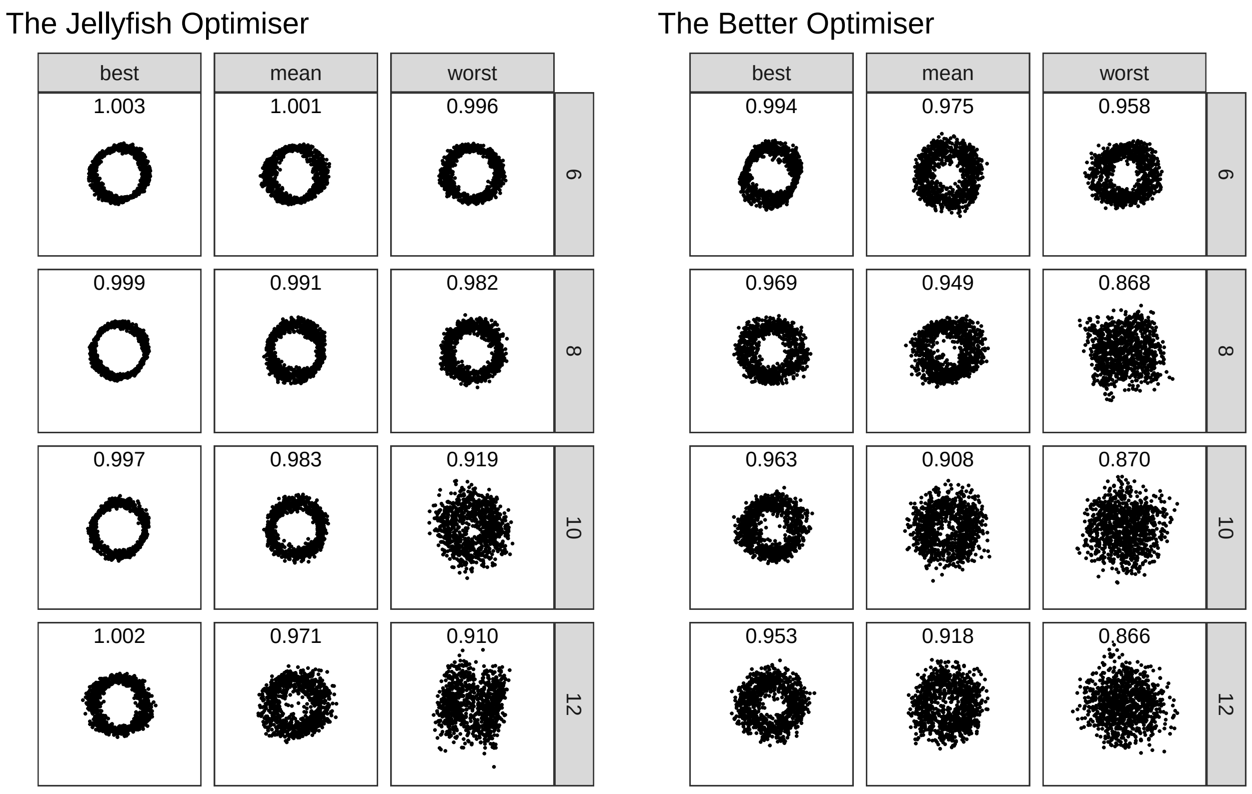

Figure 2: sthis sdfaksdlf

4.2. On skewness and kurtosis index

4.3. Another data example

5. Conclusion [Di and Sherry]

References

[1] J.-S. Chou, D.-N. Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, Applied Mathematics and Computation 389 (2021) 125535. doi:10.1016/j.amc.2020.125535.

[2] K. Rajwar, K. Deep, S. Das, An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges, Artificial Intelligence Review (2023) 1–71. doi:10.1007/s10462-023-10470-y.

[3] J.-S. Chou, A. Molla, Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems, Scientific Reports 12 (2022) 19157. doi:10.1038/s41598-022-23121-z.

[4] J. B. Kruskal, 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: R. C. Milton, J. A. Nelder (Eds.), Statistical Computation, Academic Press, New York, 1969, pp. 427–440.

[5] J. H. Friedman, J. W. Tukey, A Projection Pursuit Algorithm for Exploratory Data Analysis, IEEE Transactions on Computing C 23 (1974) 881–889.

[6] P. Hall, On polynomial-based projection indices for exploratory projection pursuit, The Annals of Statistics 17 (1989) 589–605. URL: https://doi.org/10.1214/aos/1176347127.

[7] D. Cook, A. Buja, J. Cabrera, Projection pursuit indexes based on orthonormal function expansions, Journal of Computational and Graphical Statistics 2 (1993) 225–250. URL: https://doi.org/10.2307/1390644.

[8] E.-K. Lee, D. Cook, A projection pursuit index for large p small n data, Statistics and Computing 20 (2010) 381–392. URL: https://doi.org/10.1007/s11222-009-9131-1.

[9] N. Loperfido, Skewness-based projection pursuit: A computational approach, Computational Statistics and Data Analysis 120 (2018) 42–57. doi:https://doi.org/10.1016/j.csda.2017.11.001.

[10] N. Loperfido, Kurtosis-based projection pursuit for outlier detection in financial time series, The European Journal of Finance 26 (2020) 142–164. doi:https://doi.org/10.1080/1351847X.2019.1647864.

[11] D. Cook, A. Buja, J. Cabrera, C. Hurley, Grand tour and projection pursuit, Journal of Computational and Graphical Statistics 4 (1995) 155–172. URL: https://doi.org/10.1080/10618600.1995.10474674.

[12] H. Wickham, D. Cook, H. Hofmann, A. Buja, tourr: An R package for exploring multivariate data with projections, Journal of Statistical Software 40 (2011) 1–18. URL: http://doi.org/10.18637/jss.v040.i02.

[13] H. S. Zhang, D. Cook, U. Laa, N. Langrené, P. Menéndez, Visual diagnostics for constrained optimisation with application to guided tours, The R Journal 13 (2021) 624–641. doi:10.32614/RJ-2021-105.

[14] U. Laa, D. Cook, Using tours to visually investigate properties of new projection pursuit indexes with application to problems in physics, Computational Statistics 35 (2020) 1171–1205. doi:10.1007/s00180-020-00954-8.

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