Mode hunting through active information

Daniel Andrés Díaz–Pachón[a,*], Juan Pablo Sáenz[b], J. Sunil Rao[a,**],
Jean-Eudes Dazard[c]

[a] *Division of Biostatistics - University of Miami, Don Soffer Clinical Research Center, 1120 NW 14th St, Miami FL, 33136*
[b] *Department of Industrial Engineering - University of Miami, McArthur Engineering Building, 1251 Memorial Drive, Coral Gables FL, 33146*
[c] *Center for Proteomics and Bioinformatics, Case Western Reserve University, 10900 Euclid Av. Cleveland, OH 44106*

Abstract

We propose a new method to find modes based on active information. We develop an algorithm that, when applied to the whole space, will say whether there are any modes present *and* where they are; this algorithm will reduce the dimensionality without resorting to Principal Components; and more importantly, population-wise, will not detect modes when they are not present.

1. Introduction

When Newton and Leibniz developed Calculus independently during the seventeenth century AD [11, 13], one of their main goals was to find a way to determine the position of maxima (or minima) in an Euclidean space. Of course, their interest in constructing a mathematical theory to discover maxima shows the already existent need to actually recognize these critical points —a fact well illustrated by the usage Newton made of his own "fluxions and infinite series" when a few decades later he proposed physical mechanics to explain gravitation in his *Principia* [12], arguably the most important scientific book of all times (see, e.g., [1]).

Since then, the vast majority of the scientific endeavor has been related in one way or another to finding these maxima. In statistics, modes (and generalizations to regions with high accumulation of points) have been always central to the theory. In turn, modes have paved the way to introduce related concepts in data structures like bumps, components, clusters, or classes, among others. But chasing modes has proven to be extremely difficult, and large amounts of statistical research are devoted to improve already existing methods or developing new ones that allow a more efficient way to tackle the problem. Due to

[*] Corresponding author
[**] Principal corresponding author

the advent of big data, bump hunting has thus acquired more and more importance. In this paper, we are primarily concerned with the particular case of *mode* hunting (so from here on, every time that we are talking about *bumps*, we refer in particular to *modes*, a language convenience to avoid redundancies.)

Parametrical methods work only relatively well in low dimensions. The way they work is rather straightforward: we find the distribution of a sample, say through a goodness of fit test, so then we know the position of its modes. For instance, if we find that our sample is distributed like a $\mathcal{N}(\mu, \sigma^2)$, we will know that there is a bump at μ. Thus, it is not difficult to see how big data and large dimensionality will render useless most of these approaches. For this reason, non-parametrical approaches have become the default, even though these are also affected by the curse of dimensionality, among other problems in large dimensions. Our proposed strategy takes elements from both: like a goodness of fit test, it will compare the empirical distribution to a uniform one, if there are any differences between the two distributions, there is a mode. Like non-parametrical methods it will go for the particular modes to chase them. Therefore, the algorithm we present here will say whether there are any bumps present *and* where they are.

When $p \gg 0$, it is usually the case that parsimony is important in the sense that most of the interest is reduced to a few p' variables such that $0 < p' \ll p$. The usual strategy used to reduce the dimensionality of the original information is to apply principal components (see. e.g. [5]), but this kind of reduction presents several problems in that the response might be associated to the zeroed dimensions [8, 10]. The algorithm we propose here reduces the dimensionality with no resource to principal components. This is the goal of Section 3.

Maybe the best known bump hunting algorithm is the Patient Rule Induction Method (PRIM) [7]. PRIM is a greedy algorithm made of three stages: peeling, pasting, and covering. The most important to understand here is peeling: The data lives in a p-dimensional box B and, among a set $\mathcal{B}$ of elegible sub-boxes, we are going to remove the sub-box b^* that maximizes the probability of $B - b$, for $b \in \mathcal{B}$. The boxes in $\mathcal{B}$ are the ones at the "extreme" of each dimension j; i.e., in each dimension we consider $b_{j^-} = \{\mathbf{x} : x_j < x_{j(\alpha)}\}$ and $b_{j^+} = \{\mathbf{x} : x_j > x_{j(1-\alpha)}\}$, where $x_{j(\alpha)}$ and $x_{j(1-\alpha)}$ are, respectively, the α quantiles and $(1-\alpha)$ quantiles. Then $\mathcal{B}$ is made of $2p$ boxes. Once we have $B - b^*$ we iterate the same procedure but now replacing B by $B - b^*$ and choosing the subboxes among the ones defined by the α and $(1-\alpha)$ quantiles in $B - b^*$. We iterate the process until the peeled box B^* reaches some probability threshold β. At the end, according to PRIM, there is a bump in B^*.

However, PRIM has at least two limitations: it becomes computationally expensive in high dimensions and, as shown by Polonik and Wang, it has problems in two or more dimensions because "PRIM might not be able to resolve two distinct modes" [14]. The second problem persists even when the tuning meta-parameters of the algorithm are chosen carefully. This means that in the end PRIM identifies as modes some things that are not modes. By contrast, the algorithm we propose here finds modes whenever they are present, up to the tuning of a parameter we call b, and we can be sure that it won't identify as modes anything that is not a mode.

In Section 3 we introduce our algorithm formally, in Section 4 we prove it is consistent. Section 5 presents some examples that illustrate the reach of the algorithm proposed. Section 6 uses the fact that low probability makes for large information, so that the curse of dimensionality in probabilistic terms can be seen also as a bless in terms of information. As for this Introduction, we now proceed to explain the basic facts of what we call *active information* and its central role in mode hunting.

2. Active information

Conservation of Information (CoI) is a concept that has been around since the nineteenth century when Lady Lovelace for the first time alluded to it. Bringsjord et al. paraphrase her like this [2]:

> *Computers can't create anything. For creation requires, minimally, originating something. But computers originate nothing; they merely do that which we order them, via programs, to do.*

Almost two centuries after Lady Lovelace, many statements on CoI have been posed but they have only strengthen the case on information being conserved. The more definite affirmation came right at the end of the twentieth century, when Wolpert and Macready proved their so-called No Free Lunch Theorems (NFLT), in which they showed that, in the absence of additional information, no computer search is better on average than a blind search [15, 16]. But in real life some searchs do better than others. Why? Because the programmer is adding the relevant information given what he knows about the search. In Wolpert and MacReady's words, a search can be improved only by "incorporating problem-specific knowledge into the behavior of the [optimization or search] algorithm."

Active information (AI), following on the reasoning subjacent to the NFLT, has been introduced in order to measure how much information is being added by the programmer in a computer search [4]. In what remains of this section, we will introduce it in terms of mode hunting.

Assume a sample space $\Omega = \{1, \ldots, N\}$, with $N \in \mathbb{N}$. Since, by its very definition, the uniform distribution over Ω is the only one with total absence of bumps, we know that any nonuniform distribution over Ω *do* forms bumps. This simple idea can be taken as a proof for the existence of bumps —use any of the goodness-of-fit tests in the market and compare against uniformity. If the difference is significative, reject the hypothesis of uniformity, which is equivalent to accept that the underlying distribution has bumps. This is useful by itself, but would be incomplete if it does not say *where* the bumps are. A way to detect their location follows:

Denoting $\mathbf{U}$ as the uniform distribution, let's define

$$I_\Omega = -\log \mathbf{U}(\omega)$$

as the *endogenous information* of the event ω. Denoting now as $\mathbf{S}$ the empirical distribution coming from the sample, define

$$I_S = -\log \mathbf{S}(\omega)$$

as the *exogenous information* of the event ω. Finally, we can define the *active information* of the event ω as [4]:

$$I_+ := I_\Omega - I_S = \log \frac{\mathbf{S}(\omega)}{\mathbf{U}(\omega)}.$$

With this apparatus, we propose to calculate the active information of each singleton event in Ω. Modes, whenever they are present, would be located at events such that their active information is positive. In fact, the bigger the active information, the bigger the mode is going to be.

3. Active Information Mode Hunting (AIMH) Algorithm

We mention at the beginning of the Introduction that standard parametrical approaches do not use to do well when big data and/or high dimensions are involved, so non-parametrical methods are preferred when we are chasing bumps. This is one of the main reasons to introduce algorithmic-type methods like PRIM [7] or Local Sparse Bump Hunting [3]. In the same spirit, we present here an algorithm we have dubbed Active Information Mode Hunting (AIMH) in which we mix the parametrical idea of a goodness-of-fit test to compare the data against a normal, with the non-parametrical advantage that is offered by an algorithmic approach.

Let's assume a p-dimensional rectangular space of finite Lebesgue measure $\mathbf{I}_p = I_1 \times \cdots \times I_p$. Then the r.v. with maximum entropy is $\mathbf{U}(\mathbf{I}_p)$; i.e., a p-dimensional uniform distribution whose parameter is the Lebesgue measure of $\mathbf{I}_p$. Consider $\mathbf{D} := \{1, \ldots, p\}$, and take a subset $\mathbf{D'} := \{i_1, \ldots, i_{p'}\}$, with $p' < p$, then $U_{i_1,\ldots,i_{p'}}$ is a projection of the uniform in p dimensions in p' dimensions. With this, we can extend the idea of the previous Section to find the p-dimensional modes as well as the modes in the projections through Algorithm 1 in page 5.

Then $\mathcal{C}^{(i)}$ is the collection of bumps in i dimensions found with this method, where $i = 1 \ldots, p$.

The algorithm is better understood informally in plain English:

1. Partitionate every single dimension into subintervals.
2. Calculate the AI of each subinterval in each dimension.
3. In each dimension, keep those subintervals whose AI is bigger than some threshold b_1 (In fact, this can be generalized a little further considering different thresholds in every dimension b_{1i}, for i from 1 to p.)

Algorithm 1 Active Information Mode Hunting (AIMH)

```
• For $i$ from 1 to $p$:
    – Partitionate the interval $I_i$ into $r_i$ subintervals $I_{ij}$, with $j = 1, \dots, r_i$.
• For $i$ from 1 to $p$:
    – For $j$ from 1 to $r_i$:
        * Calculate the AI:
              $I_+(I_{ij}) = \log \frac{P_s[X_s \in I_{ij}]}{P[U_i \in I_{ij}]}$,
          where $X_s$ is the r.v. of the empirical distribution.
    – Define $\mathcal{C}_i = \{I_{ij} : I_+(I_{ij}) > b_1; j = 1, \dots, r\}$, for a pre-specified
      $b_1 \in \mathbb{R}^+$.
• Define $\mathcal{C}^{(1)} := \{\mathcal{C}_i : i = 1, \dots p\}$.
• For $k$ from 2 to $p$:
    – Take every $k$-dimensional hyper-rectangle $I_{i_1 j_1} \times \cdots \times I_{i_{k-1} j_{k-1}} \times$
      $I_{i_k j_k} \in \mathcal{C}^{(k-1)} \times \mathcal{C}^{(1)}$ and calculate its AI:
          $I_+ \left(I_{i_1 j_1} \times \cdots \times I_{i_k j_k}\right) = \log \frac{P_s[X_s \in I_{i_1 j_1} \times \cdots \times I_{i_k j_k}]}{P[U_{i_1,\dots,i_k} \in I_{i_1 j_1} \times \cdots \times I_{i_k j_k}]}$.
    – Define
          $\mathcal{C}_{i_1 \dots i_k} = \big\{I_{i_1 j_1} \times \cdots \times I_{i_{k-1} j_{k-1}} \times I_{i_k j_k} \in \mathcal{C}^{(k-1)} \times \mathcal{C}^{(1)} :$
              $I_+ \left(I_{i_1 j_1} \times \cdots \times I_{i_k j_k}\right) > b_k; \; i_k \notin \{i_1, \dots, i_{k-1}\}$ and
              $j_\ell = 1, \dots, r_\ell, \text{ for } \ell = 1, \dots, k\big\}$,
      for a pre-specified $b_k \in \mathbb{R}^+$.
    – Define $\mathcal{C}^{(k)} = \bigcup_{\mathcal{I}} \mathcal{C}_{i_1 \dots i_k}$, where $\mathcal{I}$ comprises the $\binom{p}{k}$ cases of different
      $\mathcal{C}_{i_1 \dots, i_k}$.
    – If $\mathcal{C}^{(k)} = \emptyset$, halt.
```

4. For $i \neq j$, take all the subintervals in dimensions i and j whose AI was above the threshold b_1 (or b_{1i} and b_{1j} in the generalization) and construct bi-dimensional boxes.

5. Define a threshold b_2 (or b_{2ij}) and collect the bi-dimensional boxes in dimensions i and j whose AI is above the threshold.

6. Construct three-dimensional cubes with the boxes whose AI was above the threshold b_2 and the subintervals whose AI was above the threshold b_1.

7. Collect the cubes whose AI is above a threshold b_3.

8. Repeat recurrently the procedure adding the significative unidimensional subintervals to the significative hyper-cubes, to form hyper- cubes with one additional dimension, either until exhaustion of dimensions or until there is a particular number of dimensions in which no hyper-cube has AI higher than the specified threshold.

Several remarks are in order.

First, in some cases, it is not necessary to measure the AI against the uniform distribution, since the background distribution is already given. In these cases, all that is needed is simply to replace **U** by the relevant probability, say **Q**. In this case, the algorithm presented also works replacing in each dimension the marginal uniform by the corresponding marginal distribution.

Second, we can also consider more general σ-finite spaces, replacing the endogenous distribution by a distribution of interest. If the due knowledge about the moments is given, this endogenous distribution can take the form of a maximum entropy distribution (see, e.g., [6]). In reality, however, this is only of theoretical interest, since in applications all the spaces will be effectively finite, therefore uniformity will do as background in more computational implementations.

Third, the algorithm depends on the choice of the metaparameters r_i (the number of intervals in the partition per dimension) and b_i, for $i = 1, \ldots, p$. Under reasonable assumptions, it is expected of Algorithm 1 to diminish the size of the space considered at a fast speed. If it is reasonable to set every dimension to have the same amount of subintervals (i.e., $r = r_1 = \cdots = r_p$), then it also makes sense to define $b_{k+1} := b_k r^{k+1}$, so that actually we only need to define b_1. From now on, we use this assumption.

4. Consistency

In this section we prove that Algorithm 1 in Section 3 estimates consistently the modes of the population distribution function. Without loss of generality, we assume that the underlying space is $[0, 1]^p$, and, as a simplifying assumption, we partition every dimension in the same amount of subintervals, i.e., $r_1 = \cdots = r_p = r$.

Let's define first a more general version of AI with respect to a different underlying distribution:

Definition 1 (**AI under X with respect to Y**)**.** Let X and Y be continuous r.v.'s over the same sample space Ω, and let A in Ω be a Borel set such that

$P_Y(A) > 0$, then

$$I_+^{X|Y}(A) = \log \frac{P_X(A)}{P_Y(A)}.$$

With this definition we prove the following result, from which consistence is derived:

Theorem 1. *Let F_n be an empirical distribution function coming from the continuous distribution F over the sample space $[0,1]$. The AI under F_n with respect to U consistently estimates the AI under F with respect to U:*

$$I_+^{F_n|U}(A) \xrightarrow{n\to\infty} I_+^{F|U}(A), \quad a.s. \tag{1}$$

In order to prove Theorem 1, for a vector $\mathbf{x} = (x_1, \ldots, x_p) \subset [0,1]^p$, we define first $[0,\mathbf{x}]^p$ as the p-dimensional box $[0,x_1] \times \cdots \times [0,x_p]$. We have the following two lemmas:

Lemma 1. *Let F_n be an empirical distribution function taken from a continuous distribution F over the sample space $[0,1]$, then*

$$\left| I_+^{F_n|F}([0,\boldsymbol{x}]^p) \right| \xrightarrow{n\to\infty} 0, \quad a.s.$$

Proof. By the Glivenko-Cantelli Theorem, $F_n(x)$ converges to $F(x)$, with probability one, for every $\mathbf{x} \in [0,1]^p$. Since $F_U(\mathbf{x}) = x_1 \cdots x_p$, the quotient $F_n(\mathbf{x})/F_U(\mathbf{x})$ is always defined for $\mathbf{x} \in (0,1]^p$. An application of the continuous mapping theorem completes the proof. □

AI satisfies this very convenient property:

Lemma 2. *Let X, Y, and Z be r.v. in Ω. Let A be a measurable set such that $F_Y(A)$ and $F_Z(A)$ are not zero. Then*

$$I_+^{X|Z}(A) = I_+^{X|Y}(A) + I_+^{Y|Z}(A).$$

Proof.

$$I_+^{F|H}(A) = \log \frac{P_X(A)}{P_Z(A)} = \log \frac{P_X(A)}{P_Y(A)} + \log \frac{P_Y(A)}{P_Z(A)} = I_+^{F|G}(A) + I_+^{G|H}(A).$$

□

Theorem 1 follows now easily from the previous two lemmas:

Proof of Theorem 1. Let X_n be the r.v. associated to F_n, X the r.v. associated to the limiting distribution F, and U a uniform in $[0,1]$. By Lemma 2, we obtain for $A = [0,\mathbf{x}]^p$ that

$$I_+^{F_n|U}(A) = I_+^{F_n|F}(A) + I_+^{F|U}(A).$$

By Lemma 1, as $n \to \infty$, we obtain then that $I_+^{F_n|U}(A) \to I_+^{F|U}(A)$. Since the class $\{[0,\mathbf{x}]^p : \mathbf{x} \in [0,1]^p\}$ is a π-system that generates the Borel σ-algebra, the monotone class Theorem (Dynkin's Theorem) generalizes the result to every Borel set A such that $F(A) \neq 0 \neq |A|$. □

5. Examples

In this section we examine three examples. The first illustrates in one dimension how our algorithm works, the last two explain in two dimension the workings of the algorithm. The third in particular is very interesting, since it shows its superiority to PRIM in that it finds bumps that PRIM is not able to detect.

A nice feature of Algorithm AIMH is that the AI of each box can be calculated exactly due to the fact that we are using the probability induced by the empirical distribution and the uniform one. Or in the following examples, we can limit ourselves to compare the AI in the relevant boxes based on the probabilities of the regions under the parametrical endogenous and exogenous distributions. Therefore, no simulations are needed.

5.1. Normal of low variance $N(0, 10)$

In this example we examine the active information in an interval centered around the mean.

Let $X \sim N(0, 10)$ restricted to $\Omega = [-8.5, 8.5]$. Call $A = [-.5, .5]$. Thus, $P[X \in A] = 0.0659$. From here we can calculate the active information of the event A as follows:

$$\begin{aligned} I_+(A) &\approx \log_2 1.1203 \\ &\approx 0.1638. \end{aligned}$$

Then, for $r = 1/17$ and any $b > .163$ (a very small bound!), our algorithm does conclude that there is no bump in A. When we realize that $P[X \in A]$, and $P[U \in A]$ are pretty close (0.0659 and 0.0526, respectively), it does not look that surprising. However, if we are sure that there is a mode (like in this case), we can always lower our bound. In this case $b = 0.16$ will find a bump for the interval A.

This example shows how the process works in one dimension: of all the possible subintervals we consider only the ones in which the active information has more bits than b_1.

5.2. Bivariate normal $N((0, 0), \boldsymbol{I})$

Consider $\mathbf{Z} \sim N((0, 0), \mathbf{I})$, with marginals X and Y. Call $A = [-.5, .5]$. Then

$$P(X \in A) = P(Y \in A) \approx 0.3829.$$

Considering again the interval $\Omega = [-8.5, 8.5]$ in each dimension, we have that

$$\begin{aligned} I_+(A) &= \log_2(.3829 \times 17) \\ &= \log_2(6.5093) \\ &= 2.7. \end{aligned}$$

Therefore, if our $b_1 > 2.5$, then the interval A in each dimension is collected. Then we make the Cartesian product $A \times A$ and calculate its AI:

$$\begin{aligned} I_+(A \times A) &\approx \log_2(0.1444 \times 289) \\ &\approx \log_2(41.73) \\ &\approx 5.38. \end{aligned}$$

This example illustrates how to apply the algorithm when $p = 2$. It reveals the nice feature of Algorithm 1 that it does not even need to consider a whole dimension, but only the portion in each dimension that if finds relevant. That is why we only consider the subintervals centered around the mean to construct the boxes in dimension 2, because only those had information bigger than 2.5.

The example also illustrates the importance of considering an algorithmic method, since choosing first all subintervals in every dimension (which in this example amounts to one subinterval in each dimension) allows us to go in a second step to make boxes to look at the AI in them.

5.3. Bivariate mixture of normals

Consider two bivariate random vectors $\mathbf{X} \sim N\left((-3,0), \frac{1}{4}\mathbf{I}\right)$ and $\mathbf{Y} \sim N\left((3,0), \frac{1}{4}\mathbf{I}\right)$ with distributions $f_{\mathbf{X}}$ and $f_{\mathbf{Y}}$, respectively. Consider the new random vector $\mathbf{Z}$ with density

$$f_{\mathbf{Z}} = \frac{1}{2} f_{\mathbf{X}} + \frac{1}{2} f_{\mathbf{Y}}$$

Thus each the marginal density of Z_1 is given by $f_{Z_1} = \frac{1}{2} f_{X_1} + \frac{1}{2} f_{Y_1}$, and the marginal $Z_2 \sim N(0, 1/4)$. We also consider the space $\Omega = [-8.5, 8.5]$ in the first axis, and $[-4.5, 4.5]$ in the second one.

Then partitioning each dimension in intervals of length one (centered around every integer), and taking $b_1 = 2.6$, we obtain:

$$\begin{aligned} P(Z_1 \in [-3.5, -2.5]) &= 0.5(0.6827) + 0.5P(Y_1 \in [-3.5, -2.5]) \\ &\approx 0.5(0..6827) + \varepsilon \\ &\approx 0.3413 + \varepsilon \\ P(Z_1 \in [2.5, 3.5]) &\approx 0.3413 + \varepsilon. \end{aligned}$$

For $P(Z_2 \in [-.5, .5]) \approx 0.6827$.

The active information of these intervals is as follows:

$$\begin{aligned} I_+^{Z_1}([2.5, 3.5]) &\approx \log_2(0.3413 \times 19) \\ &\approx \log_2 6.48 \\ &\approx 2.69. \\ I_+^{Z_2}([-0.5, 0.5]) &\approx \log_2(0.6827 \times 9) \\ &\approx \log_2 6.14 \\ &\approx 2.61. \end{aligned}$$

No other intervals have active information bigger than 2.6. Therefore in Z_1 we take the intervals $[-3.5, 2.5]$ and $[2.5, 3.5]$; in Z_2 we take the interval $[-0.5, 0.5]$.

Having done this, we now consider the active information inside the Cartesian products $A = [-3.5, -2.5] \times [-0.5, 0.5]$ and $B = [2.5, 3.5] \times [-0.5, 0.5]$.

$$\begin{aligned} I_+(A) = I_+(B) &\approx \log_2(0.4577 \times 171) \\ &\approx 6.29. \end{aligned}$$

This example is interesting and important. Polonik and Wang illustrated through some simulations the incapacity of PRIM in two or more dimensions to differentiate the two picks of a bimodal symmetric mixture of two normals; PRIM finds a single region containing the two bumps and the valley between them (see Figure 3 in [14]). Our example has the same characteristics as the distribution considered in [14], and shows that, given an adequate tuning of the parameters, active information is able to differentiate the two picks through two different boxes. In fact, in general, active information is able to find as many modes as there are present, provided the right choice of the parameters.

As in the previous example, even with only two dimensions, this one again illustrates the importance of going algorithmically increasing the dimension of the boxes that are relevant in terms of AI.

6. The bless of dimensionalty

As it is well known, as long as the number of dimensions is increasing, and the number of observations is fixed, the curse of dimensionality will make it difficult to find uniform distributions (see, e.g., [9], pp. 22–27). The easy way to see this is to consider a discrete uniform r.v. with parameter N in a sample space Ω. Then, the random vector whose components are iid uniform r.v. living each in Ω lives itself in the Cartesian product $\Omega \times \cdots \times \Omega$ (p times), so that each point has probability N^{-p}. Thus most, if not all, learning methods —and this is not restricted to bump hunting— will fail to detect uniform distributions in high dimensions, unless the sample size is growing exponentially with the number of dimensions.

On the other hand, this exact problem serves well in terms of information —lower probability always makes for logarithmically higher information. We can state formally the previous assertion in terms of the following Proposition:

Proposition 1. *When we are considering the active information of a random vector whose components are iid uniform random variables distributed uniformly in a sample space Ω, the active information of the random vector becomes a linear function of p with slope $-I_\Omega$ and intercept $-I_S$.*

Proof.

$$\begin{aligned} I_+ &= I_{\Omega\times\cdots\times\Omega} - I_S \\ &= -\log N^{-p} - I_S \\ &= -pI_\Omega - I_S. \end{aligned}$$

□

7. Discussion

We have proposed the AIMH algorithm, which consistently finds bumps, whenever they are present, and which informs of the lack of bumps when these are not there. Additionally, our proposed algorithm works well in high dimensional spaces as it reduces the dimensionality without resorting to Principal Components. The AIMH algorithm takes elements from both parametric and non-parametric methods to determine the presence and locations of modes. Furthermore, the AIMH algorithm has distinct advantages over the PRIM algorithm as it is not as computationally expensive in high dimensions, and will always find the modes given that the parameters r_i, and b_i are tuned accurately.

The tuning of r_i, and b_i when data is involved, exceeds the scope of this research and will be subject to further study. In applied cases, these parameters need to be determined with the existing data via cross-validation or by the researcher's knowhow using Bayesian inference.

References

[1] David Berlinski. *Newton's Gift: How Sir Isaac Newton Unlocked The System of the World.* Free Press, 2002.

[2] S. Bringsjord, P. Bello, and D. Ferrucci. Creativity, the turing test, and the (better) lovelace test. *Minds and Machines*, 11(1):3–27, 2001.

[3] Jean-Eudes Dazard and J. Sunil Rao. Local Sparse Bump Hunting. *J. Comput. Graph. Stat.*, 19(4):900–929, 2010.

[4] William A. Dembski and Robert J. Marks II. Conservation of Information in Search: Measuring the Cost of Success. *IEEE Transactions on Systems, Man and Cybernetics A, Systems & Humans*, 5(5):1051–1061, September 2009.

[5] Daniel Andrés Díaz-Pachón, Jean-Eudes Dazard, and J. Sunil Rao. Unsupervised Bump Hunting Using Principal Components. In S. Ejaz Ahmed, editor, *Big and Complex Data Analysis: Methodologies and Applications*, pages 325–345. Springer International Publishing, 2017.

[6] Daniel Andrés Díaz-Pachón and Robert J. Marks II. Generalized active information: Extensions to unbounded domains. *BIO-Complexity*, 2020(3):1–6, 2020.

[7] Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Stat. Comput.*, 9:123–143, 1999.

[8] Ali S. Hadi and Robert F. Ling. Some Cautionary Notes on the Use of Principal Components Regression. *Am. Stat.*, 52(1):15–19, 1998.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Science, New York, 2009.

[10] Ian T. Jolliffe. A Note on the Use of Principal Components in Regression. *J. Roy. Stat. Soc. C*, 31(3):300–303, 1982.

[11] Gottfried Leibniz. Nova methodus pro maximis et minimis. *Acta Eroditorum*, X:467–473, October 1684.

[12] Isaac Newton. *PhilosophiæNaturalis Principia Mathematica.* London, 1687.

[13] Isaac Newton. *The Method of Fluxions and Infinite Series with its application to the Geometry of Curve-Lines.* Henry Woodfall, 1736.

[14] Wolfgang Polonik and Zailong Wang. PRIM analysis. *J. Multivar. Anal.*, 101(3):525–540, 2010.

[15] David H. Wolpert and William G. MacReady. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[16] David H. Wolpert and William G. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.