Received: 31 January 2018 | Revised: 22 October 2018 | Accepted: 22 October 2018

DOI: 10.1002/asmb.2430

SPECIAL ISSUE PAPER

WILEY

Mode hunting through active information

Daniel Andrés Díaz-Pachón[1] | Juan Pablo Sáenz[2] | J. Sunil Rao[1] | Jean-Eudes Dazard[3]

[1]Division of Biostatistics, Don Soffer Clinical Research Center, University of Miami, Miami, Florida

[2]Department of Industrial Engineering, University of Miami, Coral Gables, Florida

[3]Center for Proteomics and Bioinformatics, Case Western Reserve University, Cleveland, Ohio

Correspondence
J. Sunil Rao, Division of Biostatistics, Don Soffer Clinical Research Center, University of Miami, 1120 NW 14th St, Miami, FL 33136.
Email: jrao@miami.edu

Daniel Andrés Díaz-Pachón, Division of Biostatistics, Don Soffer Clinical Research Center, University of Miami, 1120 NW 14th St, Miami, FL 33136.
Email: Ddiaz3@umiami.edu

Abstract

We propose a new method to find modes based on active information. We develop an algorithm called active information mode hunting (AIMH) that, when applied to the whole space, will say whether there are any modes present *and* where they are. We show AIMH is consistent and, given that information increases where probability decreases, it helps to overcome issues with the curse of dimensionality. The AIMH also reduces the dimensionality with no resource to principal components. We illustrate the method in three ways: with a theoretical example (showing how it performs better than other mode hunting strategies), a real dataset business application, and a simulation.

KEYWORDS

active information, high dimensional, mode hunting

1 | INTRODUCTION

Finding maxima has been closely related to the development of science. In statistics, modes have always been central to the theory. Modes are defined as maximum accumulation points (regions) in discrete (continuous) spaces. In turn, modes have paved the way to introduce related concepts in data structures such as bumps, components, clusters, or classes, among others. However, chasing modes has proven to be extremely difficult in large and even not-so-large dimensions, so huge amounts of statistical research are devoted to improve already existing methods or developing new ones to allow a more efficient way to tackle the problem.[1-4] In this paper, we are primarily concerned with the particular case of *mode* hunting. Thus, we propose here Active Information Mode Hunting (AIMH), a multivariate algorithm to chase modes based on information theory.

Mode hunting based on parametric methods does not work well in general. The few cases where it works are only in low dimensions. Most of these methods infer the distribution of a sample, say through a goodness of fit test, so that through the distribution itself, we determine the position of its modes. For instance, if we find that our sample is distributed as 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 challenge most of these approaches. It is not uncommon for data in large dimensions to be multimodal; this explains the growing interest on estimating, in large dimensions, conditional densities $f(y|\mathbf{x})$, where y is the outcome and $\mathbf{x}$ is the input, instead of the mere regression $\mathbf{E}(Y|\mathbf{x})$ —multimodality makes conditional expectation almost uninformative.[5] For this reason, nonparametric approaches have become popular, even though these are also affected by the curse of dimensionality, among other problems in large dimensions. Our proposed strategy in this paper takes elements from both. The intuition is as follows: like a goodness of fit test, we will compare the empirical distribution to a uniform one; if the empirical distribution does not come from the uniform, there is at least a mode present. Like nonparametric methods, an algorithm then will identify locations of the particular modes. In fact, as we will see in Section 3, the algorithm we present here, called AIMH, will determine whether there are any bumps present *and* where they are.

When the number of dimensions is large ($p \gg 0$), 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,[6] but this kind of reduction presents several problems in that the response might be associated to the zeroed dimensions.[7,8] The AIMH reduces the dimensionality avoiding the use of principal components.

Maybe the bes-known bump hunting algorithm is the Patient Rule Induction Method (PRIM).[9] The 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 subboxes, we are going to remove the subbox 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, ie, 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".[10] 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, as illustrated in Section 5, with AIMH, we can find modes whenever they are present, up to the tuning of a particular parameter, and we can be sure that it would not identify as modes anything that is not a mode.

Another algorithm for bump hunting, based on PRIM, is Local Sparse Bump Hunting (LSBH).[11] In LSBH goes through a partition in several subspaces using classification and regression trees (CART). Second, a Principal Components Analysis (PCA) or a Sparse Principal Components Analysis (SPCA) is run inside each partition. Third, (multi)modality is tested inside each partition. Fourth, PRIM is performed inside each of the partitions where modes were detected.

It is a good idea to make the partition of the space using CART. Nonetheless, the use of SPCA generates other complications. For instance, it is well known that in regression, a SPCA on the covariates space can be misleading since it is possible that some of the lower-variance components are more correlated to the response than the larger-variance ones.[7,8] Second, in applications, more often than not, interpretation of sparse principal components SPC is not clear—beyond the abstraction, it is not obvious at all how to interpret the response in terms of the rotated and reduced space induced by SPC because we do not know what those variables represent in reality. Third, when bump hunting is reduced to chasing modes by definition, there are more modes in spaces with lower variance than higher ones. To illustrate this last point, consider the bidimensional space $X = (X_1, X_2)$, where $X_1 \sim N(\mu, 1)$ and $X_2 \sim N(\mu, 10)$ and X_1 is independent of X_2. In this case, the second variable has higher variance (so it would be selected by PCA), but the bump is more clearly found in X_1 precisely because of its lower variance.

The PRIM and the LSBH are bump hunting algorithms in regression settings, ie, they are both trying to maximize the expectation of the response conditioned on a set of covariates. By contrast, our algorithm here differs from them in at least two important ways. First, we are dealing with modes, not with general bumps; this means that our algorithm here does not find maxima of any measure but only of frequencies. Second, our algorithm is a multivariate analysis, not a regression one; this means that we are not fitting a model, we are not maximizing a response in terms of explanatory variables; our goal here is more modest—to find regions with maximum frequencies in a multivariate space.

In Section 3, we introduce the algorithm AIMH formally. In Section 4, we prove that it is consistent for finding modes. Section 5 presents some theoretical examples that illustrate its reach. Section 6 applies AIMH to a real business situation comparing goals and pledges from Kickstarter, a reward-based crowdfunding platform that allows people to look for funding for creative projects. Section 7 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 blessing in terms of information. Since AIMH is based on information—actually it derives its name from a concept called *active information* (AI)—we proceed to explain it here in Section 2.

2 | ACTIVE INFORMATION

Following on the reasoning subjacent to the famous No-Free-Lunch theorems,[12,13] active information (AI) has been introduced in computer science to measure how much information is being added by the programmer in a computer search.[14-16] In what remains of this section, we will introduce it in terms of mode hunting.

Assume a sample space $\Omega = \{1, \dots, N\}$, with $N \in \mathbb{N}$. The uniform distribution over Ω, by its very definition, is the only one with total absence of modes. This simple idea can be taken to detect the existence of modes—use any of the goodness-of-fit tests in the market and any deviation from uniformity will imply that the underlying distribution has modes. Such a simple concept, however, would be useless if it does not say *where* these modes are. A way to detect their location follows:

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

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

as the *endogenous information* of the event ω. Denoting now $\mathbf{S}$ as 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 AI of the event ω as

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

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

Here, AI is defined in dimension 1. Through a parametric approach, we could compare any p-dimensional exogenous distribution against a p-dimensional uniform, but this presents a complication. In applications, even if the data is p-dimensional, there might be an interest on finding modes in dimensions lower than p too. Thus, it would be unfeasible to consider all the $2^p - 1$ possible subsets of variables (see the fourth remark at the end of Section 3). Therefore, a nonparametric method is needed. The goal of the next section is to construct an algorithm that, through AI, would be able to detect modes in dimensions higher than 1.

3 | AIMH ALGORITHM

We mention at the beginning of the Introduction that standard parametric approaches do not use to do well when big data and/or high dimensions are involved, so nonparametric methods are preferred when we are chasing bumps. This is one of the main reasons to introduce algorithmic-type methods like PRIM[9] or LSBH.[11] In the same spirit, we present here an algorithm we have dubbed Active Information Mode Hunting (AIMH) in which we mix the parametric idea of a goodness-of-fit test to compare the data against a normal, with the nonparametric advantage that is offered by an algorithmic approach.

Let us assume a p-dimensional rectangular space of finite Lebesgue measure $\mathbf{I}_p = I_1 \times \dots \times I_p$. Then, the r.v. with maximum entropy is $\mathbf{U}(\mathbf{I}_p)$, ie, a p-dimensional uniform distribution whose parameter is the Lebesgue measure of $\mathbf{I}_p$. Consider $\mathbf{D} := \{1, \dots, p\}$, and take a subset $\mathbf{D}' := \{i_1, \dots, i_{p'}\}$, with $p' < p$, then $U_{i_1,\dots,i_{p'}}$ 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 7.

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

The algorithm is better understood informally in plain English:

1. Make a partition of 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).
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 bidimensional boxes.
5. Define a threshold b_2 (or b_{2ij}) and collect the bidimensional 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.

Algorithm 1 Active Information Mode Hunting (AIMH)

- For i, from 1 to p:
  - Make a partition of 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 prespecified $b_1 \in \mathbb{R}^+$.
- 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
  $$\begin{aligned}\mathcal{C}_{i_1 \dots i_k} = \big\{ & I_{i_1 j_1} \times \cdots \times I_{i_{k-1} j_{k-1}} \times I_{i_k j_k} \in \mathcal{C}^{(k-1)} \times \mathcal{C}^{(1)} : \\ & I_+\left(I_{i_1 j_1} \times \cdots \times I_{i_k j_k}\right) > b_k;\ i_k \notin \{i_1, \dots, i_{k-1}\} \text{ and} \\ & j_\ell = 1, \dots, r_\ell, \text{ for } \ell = 1, \dots, k \big\},\end{aligned}$$
  for a prespecified $b_k \in \mathbb{R}^+$.
  - 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.

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.[17] 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 meta-parameters r_i (the number of intervals in the partition per dimension) and b_i, for $i = 1, \dots, p$. Since in large dimensions, most of the variables are usually noisy, it is expected of Algorithm 1 to diminish the size of the space considered at a fast speed (see Section 8). In fact, the number of dimensions considered in the end is bounded by how many dimensions were selected in the first stage, in the unidimensional analysis. If AIMH chose p' dimensions in the first stage, the algorithm is not going beyond a p'-dimensional analysis. Thus, b_1 will also affect how many steps the algorithm is going to take.

Fourth, comparing AIMH to subsets regression might be tempting, but it is misleading in several ways. (i) We are not dealing here with regression but with a more mundane multivariate analysis. (ii) The final analysis does not involve $2^p - 1$ different "models". In fact, consider that for $i = 1, \dots, p$, there are $\binom{p}{i}$ possible analyses involving exactly i dimensions.

Nonetheless, as stated in the previous remark, it uses to be the case that most of the variables are noisy. Therefore, calling $p^{(i)}$ the number of variables in i dimensions with a mode above the threshold, we have that $p^{(i)} \ll \binom{p}{i}$, so that the number of analyses considered in the end is much less than $2^p - 1$.

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 make a partition of every dimension in the same amount of subintervals, ie, $r_1 = \cdots = r_p = r$.

Let us 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 consistency 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*

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

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 $F_U(\mathbf{x}) = x_1 \ldots x_p$, 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. □

The 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 | THE BLESSING OF DIMENSIONALITY

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, eg, pages 22 to 27 in the book of Hastie et al[18]). 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 $\Omega \times \ldots \times \Omega$ (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 AI of a random vector whose components are iid uniform random variables distributed uniformly in a sample space Ω, the AI of the random vector becomes a linear function of p with slope $-I_\Omega$ and intercept $-I_S$.*

Proof.

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

□

6 | EXAMPLES

In this section, we examine three examples. The first illustrates in one dimension how our algorithm works, the last two explain in two dimensions 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.

6.1 | Normal of low variance $N(0, 10)$

In this example, we examine the AI 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 AI 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 AI has more bits than b_1.

6.2 | Bivariate normal $N((0,0),\mathbf{I})$

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.

6.3 | Bivariate mixture of normals

Consider two bivariate random vectors $\mathbf{X} \sim N((-3,0), \frac{1}{4}\mathbf{I})$ and $\mathbf{Y} \sim N((3,0), \frac{1}{4}\mathbf{I})$ 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 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 into 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 AI 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 AI 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 AI 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[10] 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 the work of Polonik and Wang[10]). Our example has the same characteristics as the distribution considered in the work of Polonik and Wang,[10] and shows that, given an adequate tuning of the parameters, AI is able to differentiate the two picks through two different boxes. In fact, in general, AI is able to find as many modes as there are present, provided the right choice of the parameters.

As in the previous example, this one illustrates again the importance of increasing through an algorithm the dimension of the boxes that are relevant in terms of AI.

7 | REAL DATA ILLUSTRATION: KICKSTARTER PLEDGES AND GOALS

To further illustrate our mode hunting method, we have used *Pledge* and *Goal* data from Kickstarter. Kickstarter is a reward-based crowdfunding platform that allows people to look for funding for creative projects. For these projects, creators choose one of the 15 different categories, a short description, a title, a duration of up to 60 days, a location, a funding goal, and a set of rewards for different contribution levels.

The funding goal is the amount of money that the project seeks to raise to be able to complete the project and fulfill the rewards, whereas individual pledges to any given project may range between $1 and $10 000. The funding for projects depends on its ability to raise enough pledges to meet the funding goal, if the goal is met, the project receives the complete pledged amount after platform fees that range between 8% and 10%; if the project does not meet the funding goal, it receives nothing and no fees are charged.

Since the platform's launch in April 2009, and as of February 2018, 14 million people have backed a project, $3.5 billion has been pledged, and 139 205 projects have been successfully funded. The most successful projects, as of February 2018, include *Exploding Kittens*, a project for a card game, which had 219 382 backers, and *Pebble Time*, a project for a smartwatch, which raised $20 338 986.

Multiple studies for the success of Kickstarter campaigns can be found in the literature[19] where Lamidi analyzed 41 965 projects from 2009 to 2011 using 17 features. Lamidi's analyses conclude that a Random Forest was a superior approach than a k-Nearest Neighbor algorithm or a Logistic Regression to predict a projects success, with an accuracy of 68% compared to 63.3% and 66.42%, respectively. Barbi and Bigelli[20] evaluate the success of Kickstarter projects in and outside the United States, they conclude that drivers of success include a smaller funding goal, a shorter project duration, a higher number of reward levels, and the presence of a video. Kaur and Gera[21] evaluate the effects of social media connectivity on the success of Kickstarter projects and propose a Logistic Regression model that predicts project success with a 76.7% accuracy, and state that the success of a campaign largely depends on the networking skills and efforts of the creator.

Zvilichovsky et al[22] evaluate the *Making-the-product-happen* effect, in which consumers increase their support for a project when they believe that it is crucial for the product's creation, and provide analysis from over 200 000 Kickstarter projects, from which 33% of all successful campaigns hinge on the support of less than 4 backers. As such, it is to be

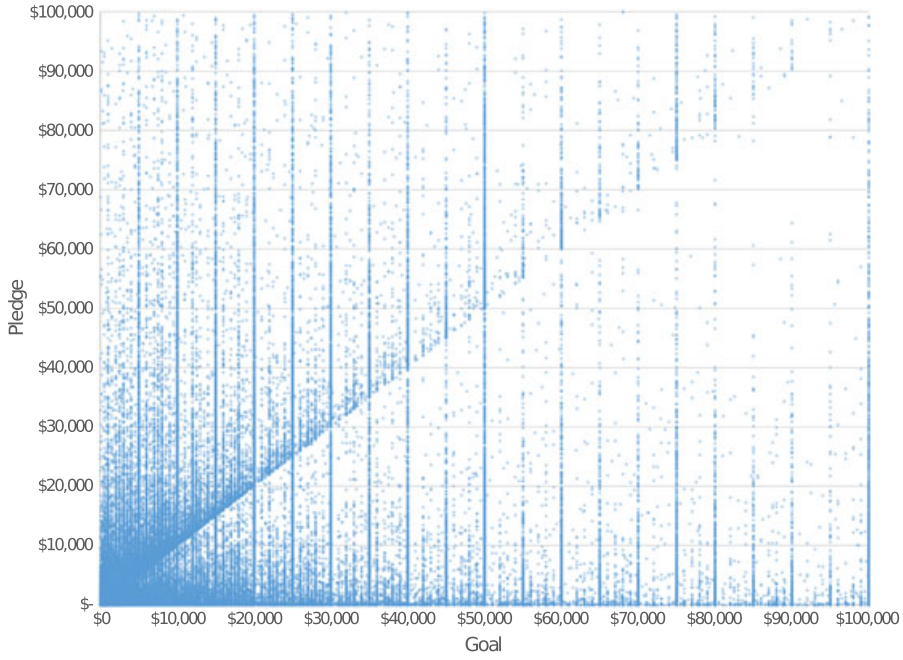

FIGURE 1 Kickstarter projects with both a goal and a pledge amount of up to $100 000 [Colour figure can be viewed at wileyonlinelibrary.com]

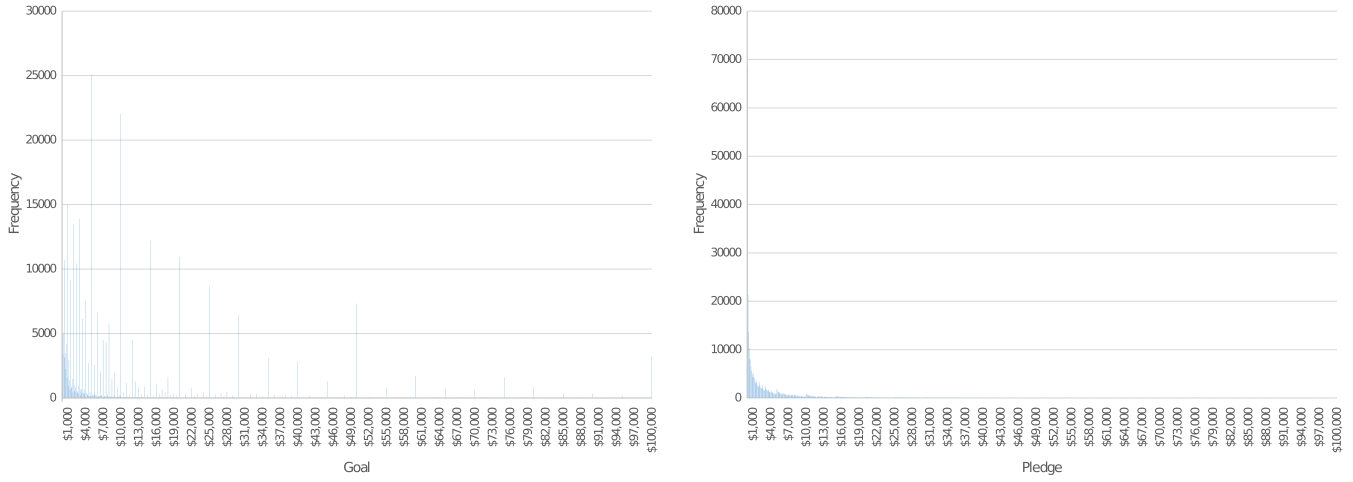

FIGURE 2 One-dimensional frequencies for the goal and pledge dimensions for projects with both a goal and a pledge amount of up to $100 000 [Colour figure can be viewed at wileyonlinelibrary.com]

expected that many projects have pledges that are very close to the project's goal. Kuppuswamy and Bayus[23] study the dynamics of a project's support over time and conclude that people pledge support for Kickstarter projects when they believe that their contribution will make an impact, and as such predict the support increases as the pledges reaches the project's goal and decreases once it has been met. According to these conclusions, it is expected for there to be modes with pledges right above the projects' goals.

Our analysis uses the projects from the platform's launch through January 2nd 2018 and of these 378 661 projects, we focus on the 311 605 where there were actually funds pledged and both the project's goal and funding were up to $100 000, shown in Figure 1 with their respective frequencies for the goal and pledge dimensions shown in Figure 2.

In this example when we have initially taken $b_1 = 3$ and $r = 1000$, with these parameters, we have found 32 modes across the goal axis, as shown in Figure 3; and 22 modes across the Pledge axis, as shown in Figure 4. Out of the modes in the goal axis, eight are located under the $1000, 23 are under $10 000; whereas there are also modes within $100 of the $15 000, $20 000, $25 000,$30 000, $35 000, $40 000, $50 000, and $100 000 goals. As such, taking only the *Goal* as a

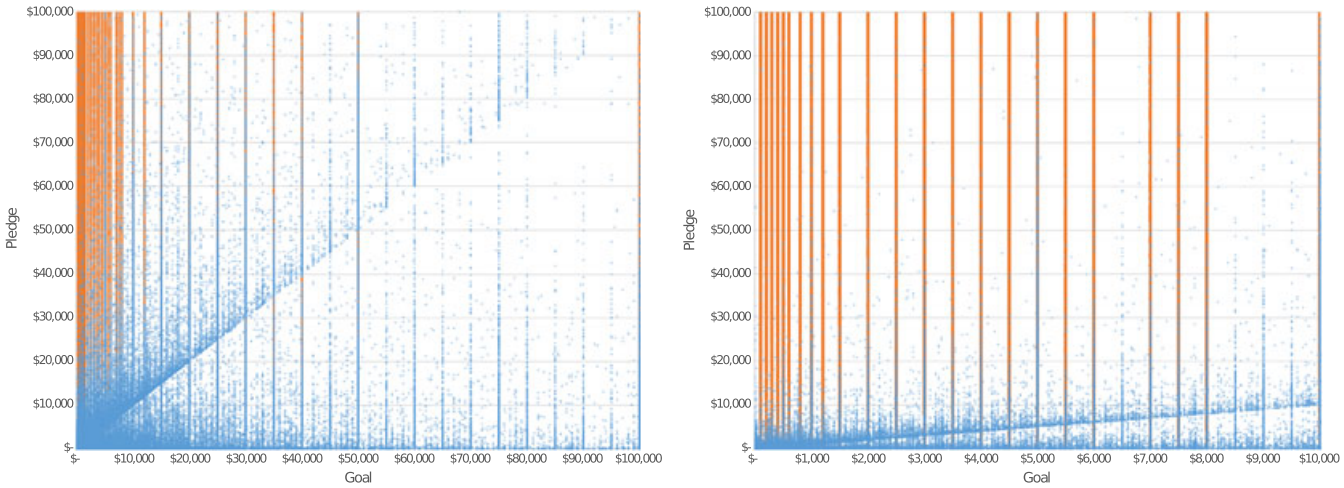

FIGURE 3 One-dimensional modes for the goal dimension for projects with both a goal and a pledge amount of up to $100 000 using a $b_1 = 3$ [Colour figure can be viewed at wileyonlinelibrary.com]

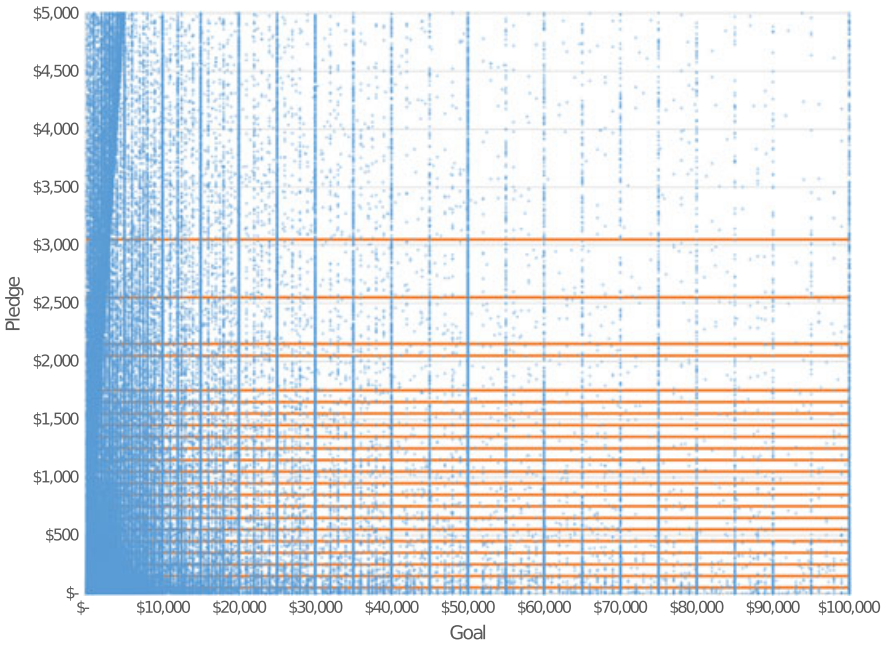

FIGURE 4 One-dimensional modes for the pledge dimension for projects with both a goal and a pledge amount of up to $100 000 using a $b_1 = 3$ [Colour figure can be viewed at wileyonlinelibrary.com]

decision making variable, it may be desirable for project creators to select a goal that is not in these modes and as such would not compete within a possibly saturated goal.

The modes in the Pledge axis are all under $3100, with modes in every box under $1800, and modes within $100 of $2100, $2200, $2600, and $3100; this should be sobering information for project creators as it shows how pledges cluster at lower pledge amounts.

We have also taken $b_1 = 4$ and $b_1 = 5$, with $b_1 = 4$, we have found 19 modes across the goal axis, as shown in Figure 5; and 10 modes across the Pledge axis, as shown in Figure 6. In this case, out of the modes in the goal axis, four are located under the $1000, 14 are under $10 000; whereas there are also modes within $100 of the $15 000, $20 000, $25 000, $30 000, and $50 000 goals. The modes in the Pledge axis are all under $1100, with modes in every box under $900, as well as modes within $100 of $1100. With $b_1 = 5$, we have found nine modes across the goal axis, as shown in Figure 7; and four modes across the Pledge axis, as shown in Figure 8. In this case, out of the modes in the goal axis, four are located within $100 of the $500, $1000, $2000, $2500, $3000, $5000, $10 000, $15 000, and $20 000 goals. The modes in the Pledge axis are in every box under $400.

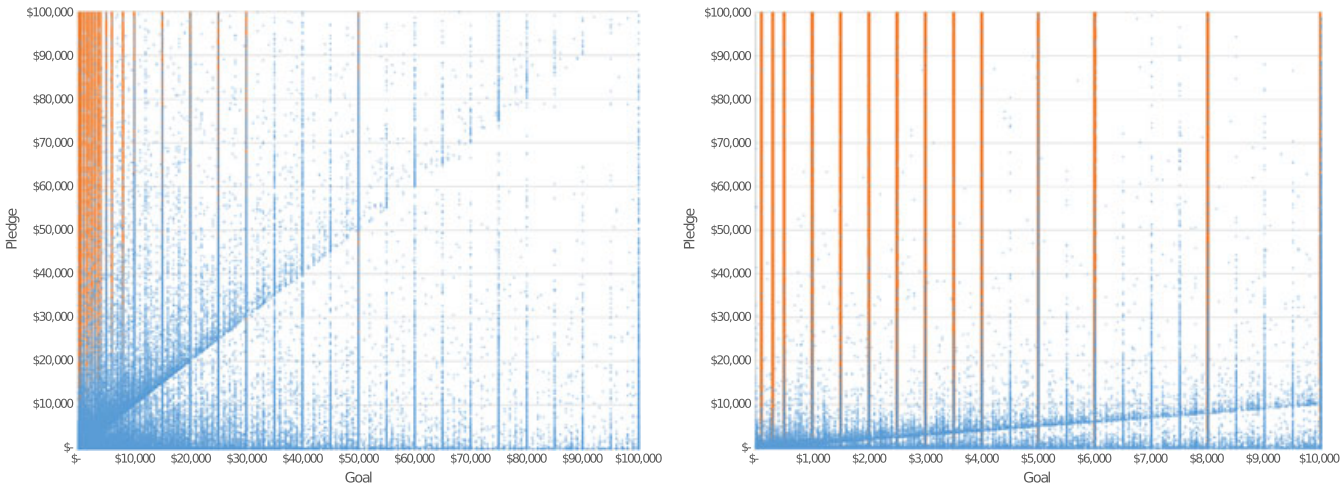

FIGURE 5 One-dimensional modes for the goal dimension for projects with both a goal and a pledge amount of up to $100 000 using a $b_1 = 4$ [Colour figure can be viewed at wileyonlinelibrary.com]

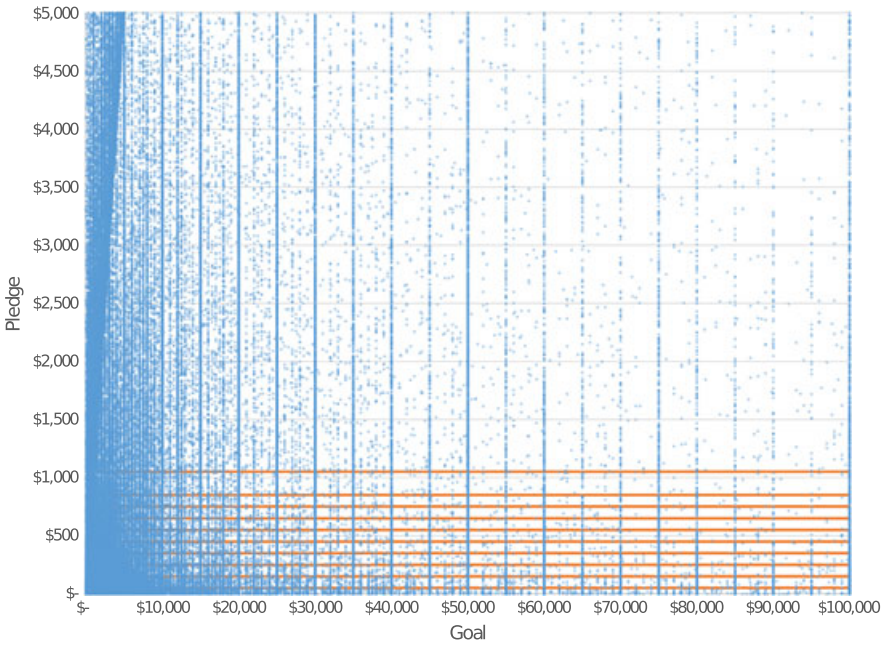

FIGURE 6 One-dimensional modes for the pledge dimension for projects with both a goal and a pledge amount of up to $100 000 using a $b_1 = 4$ [Colour figure can be viewed at wileyonlinelibrary.com]

When using $b_1 = 3$, there are 704 resulting two-dimensional hyper-rectangles; in this case, when using $b_2 = 9$, we have found 141 two-dimensional modes, as shown in Figure 9. In this case, we can see that the mode for successful projects with both the highest goal and pledge is located within $100 of $3000 in terms of the goal and $3100 in terms of the pledge. Up to this point, there are modes of successful projects at every one-dimensional goal mode as such, there are a significant number of projects that have a goal under $3000 that are successful, yet merely funded. The mode where the projects have the highest goal and barely miss getting funded is those with a goal of $1500, where there is a mode with pledges within $100 of the goal. There are, of course, various modes for unfunded projects but none that are close to funding beyond this goal. Additionally, we can also see that at the $1000 goal level, there are more modes with at points where the projects funded successfully than modes at the points where the projects did not succeed. Taking only both the *Goal* and *Pledge* as decision variables for creators, they may have better chances of succeeding with smaller goals (under $1000) whereas losing feasibility with goals beyond $3000.

When using $b_1 = 3$ and $b_1 = 10$, we have found 86 two-dimensional modes, as shown in Figure 10. In this case, we also see that the mode for successful projects with both the highest goal and pledge is located within $100 of $3000 in

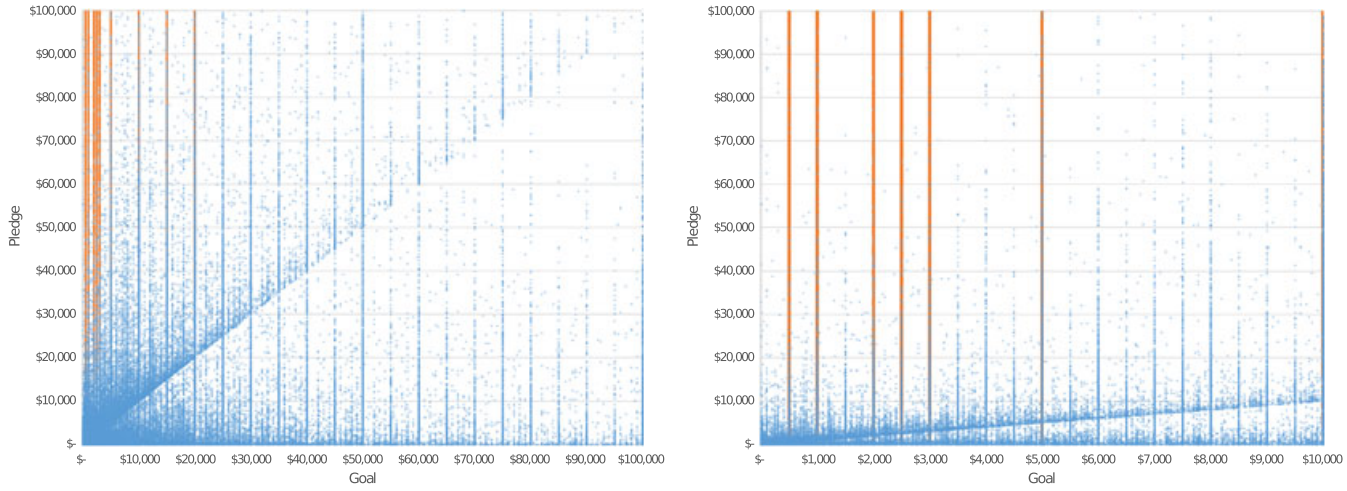

FIGURE 7 One-dimensional modes for the goal dimension for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 5$ [Colour figure can be viewed at wileyonlinelibrary.com]

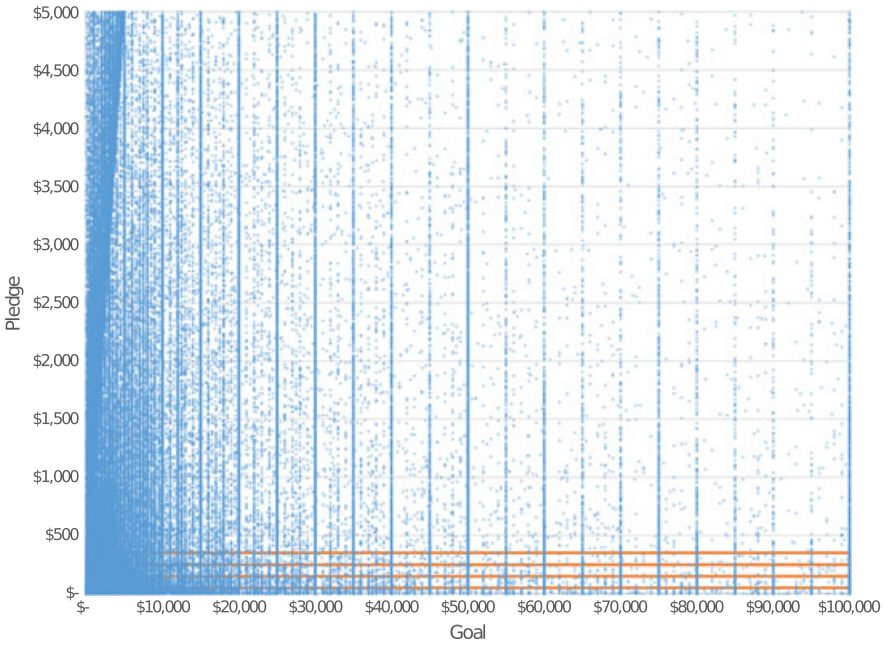

FIGURE 8 One-dimensional modes for the pledge dimension for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 5$ [Colour figure can be viewed at wileyonlinelibrary.com]

terms of the goal and \$3100 in terms of the pledge. This time, unlike when $b_2 = 9$, there is no mode with successfully funded projects at the \$1200 one-dimensional goal mode, however, there is still a significant number of projects that have a goal under \$3000 that are successful, yet merely funded. In this case as with $b_1 = 9$, we can also see that at the \$1000 goal level, there are more modes with at points where the projects funded successfully than modes at the points where the projects did not succeed.

When using $b_1 = 4$, there are 190 resulting two-dimensional hyper-rectangles, in this case, when using $b_2 = 9$, we have found 123 two-dimensional modes, as shown in Figure 11; whereas when $b_2 = 10$, there are only 76 two-dimensional modes (shown in Figure 12). Comparing these two cases, we find the greatest difference in pledges of less than \$400, where the number of two-dimensional modes drops from 75 when $b_2 = 9$ to 59 when $b_2 = 10$. In the case, when $b_1 = 5$, there are only 36 resulting two-dimensional hyper-rectangles, and there are two-dimensional modes in every one of them when $b_2 = 9$, as shown in Figure 13; whereas when $b_2 = 10$, there are only 2 hyper-rectangle in which there are no modes which are within \$100 of the \$400 pledge level with goals of \$100 and \$20 000, respectively, as shown in Figure 14.

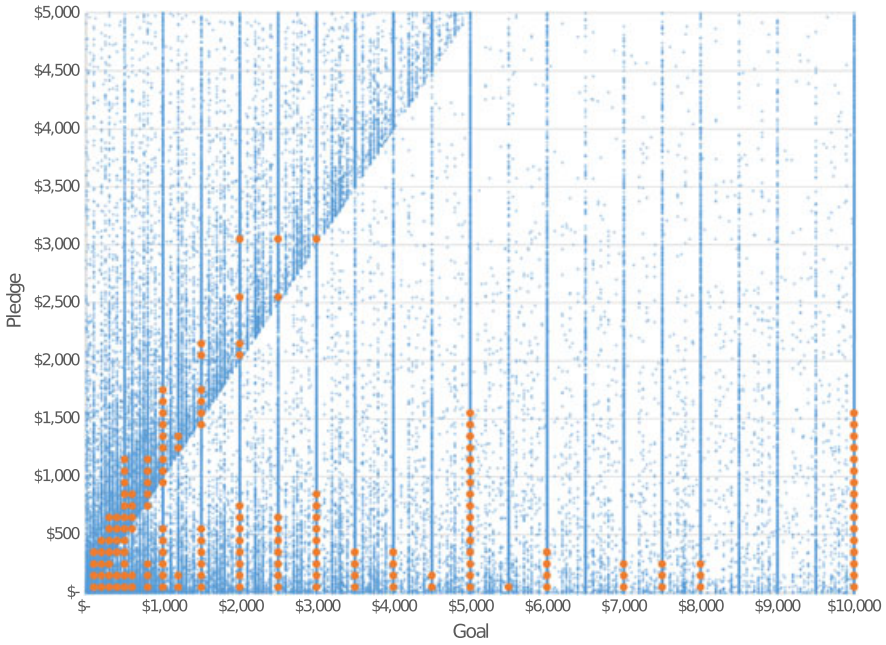

FIGURE 9 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 3$ and $b_2 = 9$ [Colour figure can be viewed at wileyonlinelibrary.com]

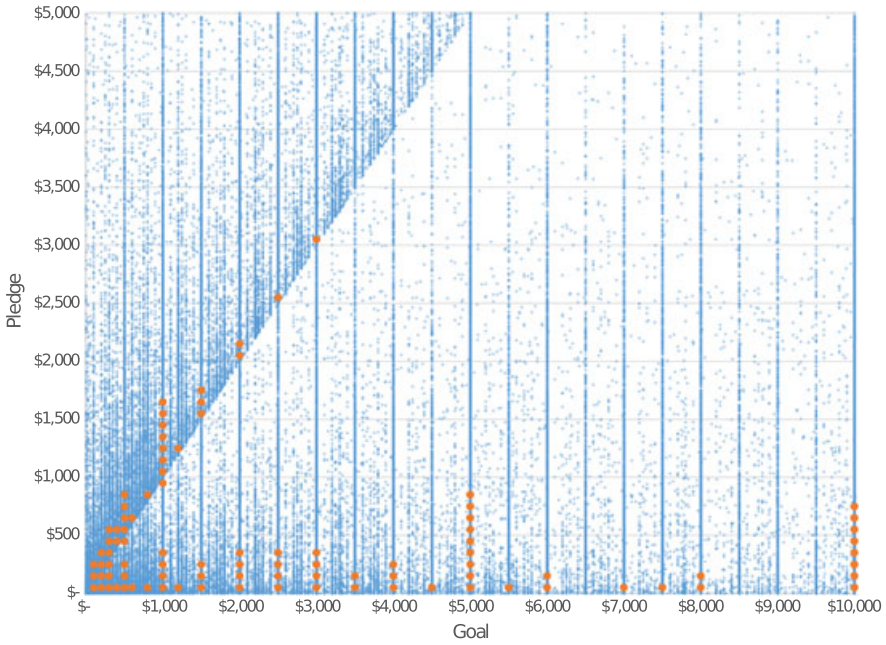

FIGURE 10 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 3$ and $b_2 = 10$ [Colour figure can be viewed at wileyonlinelibrary.com]

We can see how in the case when $b_1 = 5$, the restriction on the AI that this imposes on the first dimension leads to a very reduced number of two-dimensional hyper-rectangles where most of them have two-dimensional modes.

8 | SIMULATION

This section shows the results for the simulation of 1000 observations in a 150-dimensional space of independent random variables as shown in Table 1.

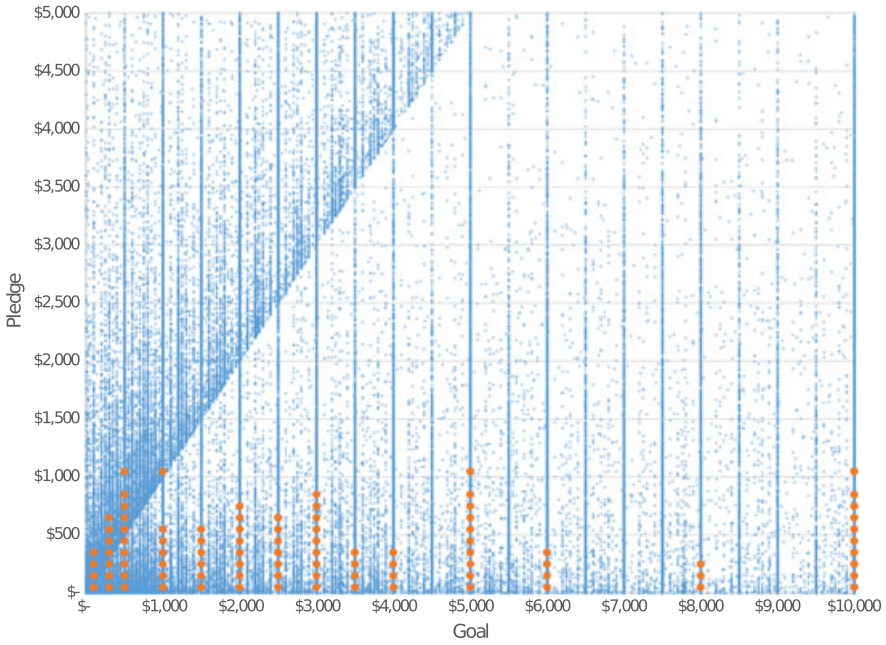

FIGURE 11 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 4$ and $b_2 = 9$ [Colour figure can be viewed at wileyonlinelibrary.com]

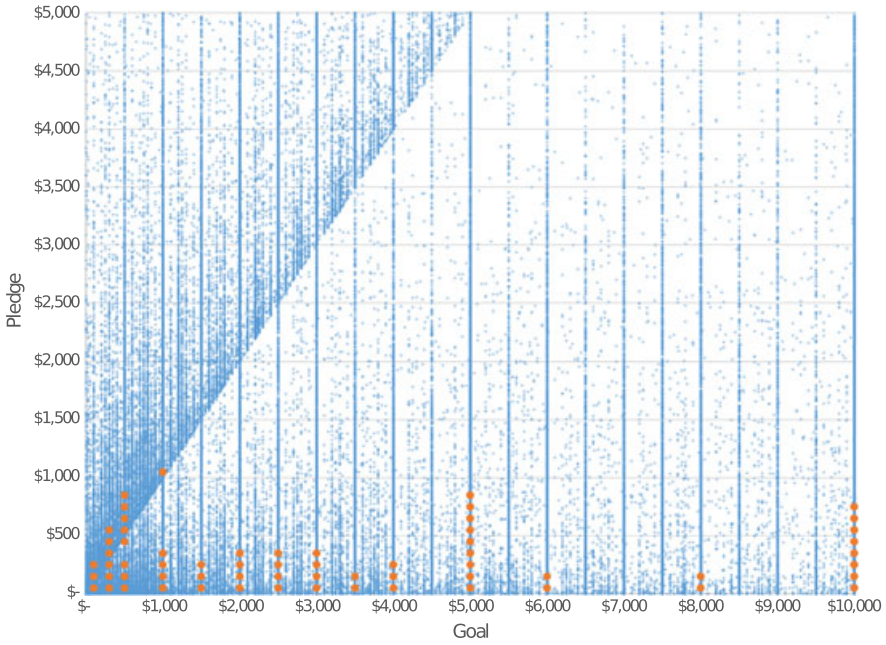

FIGURE 12 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 4$ and $b_2 = 10$ [Colour figure can be viewed at wileyonlinelibrary.com]

To determine the AI, every dimension was compared against a uniform between the min and the max. For dimension 1, we took $b_1 = 1.5$ and each r was chosen segmenting the space in 10 intervals of the same length, rounding each value to the first decimal position. After this first stage, we selected only the following six variables with their respective intervals and AI:

The purpose of the Index column is to use as nomenclature in the next table. Notice that none of the 100 uniform distributions were selected, which illustrates the robustness of the procedure. In addition, two intervals of the same variable were selected. Thus, the analysis will not go beyond 5 (< 6) dimensions, as explained in the third remark of Section 3.

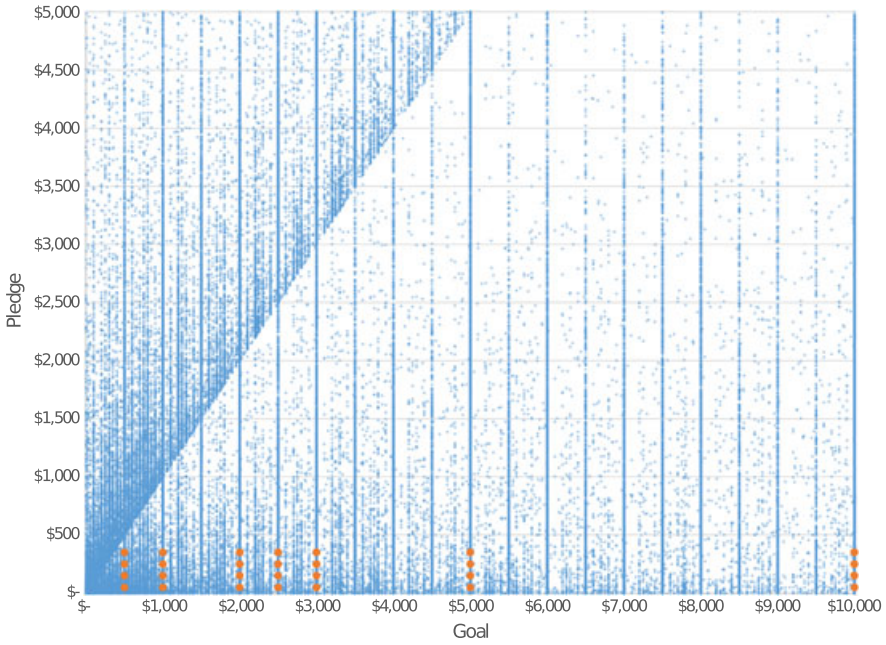

FIGURE 13 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 5$ and $b_2 = 9$ [Colour figure can be viewed at wileyonlinelibrary.com]

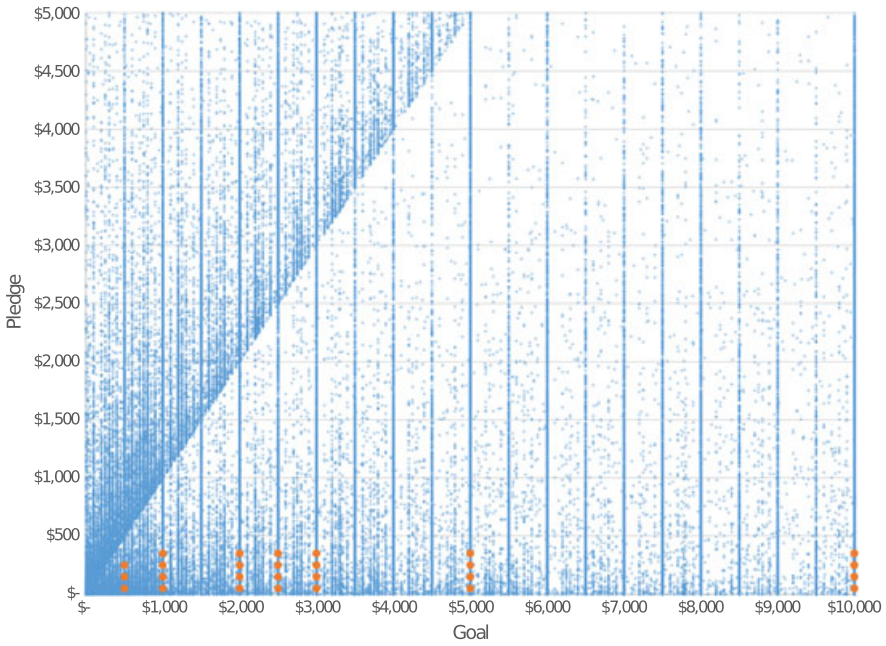

FIGURE 14 Two-dimensional modes for the pledge and goal dimensions for projects with both a goal and a pledge amount of up to \$100 000 using a $b_1 = 5$ and $b_2 = 10$ [Colour figure can be viewed at wileyonlinelibrary.com]

For the bidimensional analysis, we then select only among the selected intervals in dimension 1, with the variables defined by the indexes of Table 2. For $b_2 = 3$. The chosen rectangles are shown in Table 3.

In Table 3, we have 3 bidimensional rectangles with AI over 3. To construct the 3-dimensional cubes, we use the rectangles of Table 3 crossing them with the intervals of Table 2, and we select those with AI threshold $b_3 = 4.5$; the result is presented in Table 4.

Again, in Table 4, the Variables column is selected using the indexes in Tables 2 and 3.

Now, with $b_4 = 6$, we perform the 4-dimensional analysis, and it is presented in Table 5.

Finally, we construct the only two possible options in 5 dimensions, making $b_5 = 7.5$, we show in Table 6 that both

hyper-cubes are selected.

The selection of the b's in each dimension was always arbitrary.

TABLE 1 Description of the random variables used for simulation

Distribution	Variable
Unif$[-1, 1]$	1–100
N(0, 1)	101, 111, 121, 131, 141
N(0,2)	102, 112, 122, 132, 142
N(0,3)	103, 113, 123, 133, 143
N(0,4)	104, 114, 124, 134, 144
N(0,5)	105, 115, 125, 135, 145
N(0,6)	106, 116, 126, 136, 146
N(0,7)	107, 117, 127, 137, 147
N(0,8)	108, 118, 128, 138, 148
N(0,9)	109, 119, 129, 139, 149
N(0,10)	110, 120, 130, 140, 150

TABLE 2 Selected intervals in one dimension with AI > 1.5

Index	Variable	Distribution	Interval	AI
1.1	111	N(0,1)	$[-0.2, 0.5]$	1.5260
1.2	132	N(0,2)	$[-1, 0.5]$	1.5558
1.3	138	N(0,8)	$[-3, 3]$	1.5410
1.4	141	N(0,1)	$[-0.2, 0.5]$	1.5558
1.5	146	N(0,6)	$[-3.6, 1]$	1.5310
1.6	146	N(0,6)	$[1, 5.6]$	1.5008

TABLE 3 Selected rectangles in 2D with AI > 3

Index	Variables	AI
2.1	1.1 × 1.4	3.1858
2.2	1.3 × 1.5	3.0356
2.3	1.3 × 1.6	3.0356

TABLE 4 Selected cubes in 3D with AI > 4.5

Index	Variables	AI
3.1	1.1 × 2.2	4.7548
3.2	1.1 × 2.3	4.9068
3.3	1.2 × 2.2	4.8579
3.4	1.4 × 2.2	4.7548

TABLE 5 Selected hyper-cubes in 4D with AI > 6

Index	Variables	AI
4.1	1.2 × 3.1	6.6438
4.2	1.2 × 3.2	6.3219
4.3	1.4 × 3.1	6.9068
4.4	1.4 × 3.3	6.6438

TABLE 6 Selected hyper-cubes in 5D with AI > 7.5

Index	AI
5.1	9.2288
5.2	9.2288

9 | 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. We have shown an asymptotic result that AIMH is consistent. We have shown that AIMH is more reliable in high dimensions because information increases logarithmically in the interval $(0, 1]$ helping to overcome the curse of dimensionality. We have illustrated how it works in a theoretical example, showing that it finds bumps where PRIM cannot. We have applied it to a real dataset example. We performed a simulation, in which we also showed how it reduces the dimensionality with no recurrence to principal components. The AIMH algorithm takes elements from both parametric and nonparametric 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.

ORCID

Jean-Eudes Dazard https://orcid.org/0000-0003-4720-3684

REFERENCES

1. Hall P, Minotte MC, Zhang C. Bump hunting with non-Gaussian kernels. *Ann Stat*. 2004;32:2124-2141.
2. Mostat E. *Mode Hunting and Density Estimation With the Focussed Information Criterion* [master's thesis]. Oslo, Norway: University of Oslo; 2013.
3. Ooi H. Density visualization and mode hunting using trees. *J Comput Graph Statist*. 2002;11(2):328-347.
4. Sommerfeld M, Heo G, Kim P, Rush ST, Marron JS. Bump hunting by topological data analysis. *Stat*. 2017;6:462-471.
5. Izbicki R, Lee AB. Converting high-dimensional regression to high-dimensional conditional density estimation. *Electron J Stat*. 2017;11:2800-2831.
6. Díaz-Pachón DA, Dazard JE, Rao JS. Unsupervised bump hunting using principal components. In: Ahmed SE, ed. *Big and Complex Data Analysis: Methodologies and Applications*. New York, NY: Springer; 2016.
7. Hadi S, Ling RF. Some cautionary notes on the use of principal components regression. *Am Stat*. 1998;52(1):15-19.
8. Joliffe I. A note on the use of principal components in regression. *J R Stat Soc Ser C Appl Stat*. 1982;31(3):300-303.
9. Friedman JH, Fisher NI. Bump hunting in high-dimensional data. *Stat Comput*. 1999;9:123-143.
10. Polonik W, Wang Z. PRIM analysis. *J Multivar Anal*. 2010;101(3):525-540.
11. Dazard JE, Rao JS. Local sparse bump hunting. *J Comput Graph Stat*. 2010;19(4):900-929.
12. Wolpert DH, MacReady WG. No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010. Santa Fe, NM: Santa Fe Institute; 1995.
13. Wolpert DH, MacReady WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput*. 1997;1(1):67-82.
14. Dembski WA, Marks RJ. Bernoulli's principle of insufficient reason and conservation of information in computer search. In: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics; 2009; San Antonio, TX.
15. Dembski WA, Marks RJ. Conservation of information in search measuring the cost of success. *IEEE Trans Syst Man Cybern A Syst Hum*. 2009;5(5):1051-1061.
16. Marks RJ, Dembski WA, Ewert W. *Introduction to Evolutionary Informatics*. Singapore: World Scientific; 2017.
17. Díaz-Pachón DA, Marks RJ. Maximum entropy and active information. 2017. Submitted.
18. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer Science; 2009.
19. Lamidi A. Predicting the success of Kickstarter campaigns. 2017. https://towardsdatascience.com/predicting-the-success-of-kickstarter-campaigns-3f4a976419b9
20. Barbi M, Bigelli M. Crowdfunding practices in and outside the US. *Res Int Bus Finance*. 2017;42:208-223.
21. Kaur H, Gera J. Effect of social media connectivity on success of crowdfunding campaigns. *Inf Technol Quant Manag*. 2017;122:767-774.

22. Zvilichovsky D, Danziger S, Steinhart Y. Making-the-product-happen: a driver of crowdfunding participation. *J Interact Mark*. 2018;41:81-93.

23. Kuppuswamy V, Bayus BL. Does my contribution to your crowdfunding project matter? *J Bus Ventur*. 2017;32:72-89.

How to cite this article: Díaz-Pachón DA, Sáenz JP, Rao JS, Dazard J-E. Mode hunting through active information. *Appl Stochastic Models Bus Ind*. 2019;1–18. https://doi.org/10.1002/asmb.2430