Are your data gathered? The Folding Test of Unimodality
Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, Christine Largouët

To cite this version:
Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, Christine Largouët. Are your data gathered? The Folding Test of Unimodality. KDD 2018 - 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Aug 2018, London, United Kingdom. pp.2210-2218, 10.1145/3219819.3219994. hal-01951676

HAL Id: hal-01951676
https://hal.science/hal-01951676
Submitted on 11 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Understanding data distributions is one of the most fundamental research topics in data analysis. The literature provides a great deal of powerful statistical learning algorithms to gain knowledge on the underlying distribution given multivariate observations. We are likely to find out a dependence between features, the appearance of clusters or the presence of outliers. Before such deep investigations, we propose the folding test of unimodality. As a simple statistical description, it allows to detect whether data are gathered or not (unimodal or multimodal). To the best of our knowledge, this is the first multivariate and purely statistical unimodality test. It makes no distribution assumption and relies only on a straightforward $p$-value. Through real world data experiments, we show its relevance and how it could be useful for clustering.

Unimodality tests exist in the literature (for example the dip test [13] or the Silverman test [22]), however they are restricted to one dimensional data. Up to now, there exists no purely statistical unimodality test able to tackle multi-dimensional data without distribution assumption. One may be tempted to use unidimensional unimodality tests on each dimension of multidimensional data (in a way similar to [17]), however the simple example of Figure 1 shows the limits of such an approach. On the bottom left side, one can see that the bidimensional data is bimodal. However, its projection of the X (top) and Y (right) axis are perfectly unimodal, preventing to detect the multimodality of that dataset with unidimensional unimodality tests.

Our main contribution is to propose the first statistical unimodality test designed for multidimensional data. Our approach relies on a novel descriptive statistics for (multidimensional) data, that provides a measure of its unimodality character (level of unimodality). The proposed approach keeps the same elegant properties as well-known unimodality tests like dip: it is independant from the distribution of the data and has only one parameter (a $p$-value threshold to tune the confidence in the results). We provide an algorithm for computing our test and show that it can be adapted to the streaming context through an incremental version.

The outline of the paper is as follows: in Section 2, we present related work on unimodality tests. We then describe our folding test of unimodality in Section 3, and its experimental validation in Section 4. Section 5 concludes the paper.
2 RELATED WORK

Unimodality tests. Many statistical tests have been proposed in the literature and Stoecker gave recently a nice survey of them [23]. There are two kinds of tests: testing unimodality versus bimodality and testing unimodality versus multimodality. Once we know that a distribution is \( k \)-modal, many algorithms can be used to learn this distribution from a few samples [3, 4].

To test the “unimodality character” of a distribution, the main approaches aim at estimating the distribution. They include histograms, kernel density estimates and mixture models.

Silverman’s Test is a parametric test that uses the bandwidth of the kernel density estimate to test multimodality. A sufficiently large bandwidth gives a smooth unimodal estimate. If we need a large amount of smoothing to find an unimodal estimate, this indicates multimodality. This test has several drawbacks [10, 23] even if we know that the input distribution is a mixture of gaussians. Some corrections have been proposed in [10] and a procedure to use non-gaussian kernels is given in [9].

The Excess Mass Test of Müller et al. [18] tests unimodality versus bimodality. The idea is to estimate the excess mass of the modes which can be seen as the area between the density \( f(x) \) and a level \( L \). The test compares the excess mass under unimodal and bimodal assumptions (through a statistics noted \( \Lambda \)). Actually, only a simple histogram is needed to estimate the density. Some properties about the distribution of \( \Lambda \) have been discovered by Cheng and Hall [1] leading to a bootstrap procedure to compute \( p \)-values.

The Dip Test by Hartigan and Hartigan [13] measures the distance between the empirical cumulative distribution function (ecdf) and the set of the unimodal distributions \( \mathcal{U} \). The ecdf of the set \( \mathcal{U} \) are well characterized: they are convex on an interval \( [−\infty, x_l] \), then constant on \( [x_l, x_u] \) and finally concave on \( [x_u, +\infty] \). In short, the dip of an empirical cdf \( F \) corresponds roughly to the distance (with infinity norm) between \( F \) and the set \( \mathcal{U} \), i.e:

\[
dip(F) = d_\infty(F, \mathcal{U}) = \min_{U \in \mathcal{U}} \| F - U \|_\infty.
\]

In [13], the uniform distribution is considered as the least favorable unimodal distribution. For an empirical cdf \( F_n \), the dip test compares \( \text{dip}(F_n) \) with the quantiles of \( \text{dip}(U_n) \) where \( U_n \) is the ecdf of \( n \) iid uniform random variables (these quantiles can be pre-computed with Monte-Carlo simulations). The dip test reports a \( p \)-value \( p \) which is the probability to have an ecdf \( U_n \) with \( \text{dip}(U_n) > \text{dip}(F_n) \). The common threshold \( \alpha = 0.05 \) is used to make the decision: if \( p < \alpha \) the ecdf \( F_n \) is probably multimodal. This statistics and the test which results from it are rigorously detailed in [13] and a good analogy is given in [17]. Even if the excess mass and the dip statistics are equivalent [1] the latter is the most commonly used statistics.

The main drawback of these tests is that they are restricted to real valued random variable (dimension 1). Moreover they need to estimate the distribution (density or cdf) and may require several passes over the data. Some multivariate extensions have been proposed like the RUNT test [12] or the MAP test [20]. Unfortunately, these methods rely on the construction of several minimal spanning tree and the use of expensive optimization steps. So they are far more complex than our purely statistical approach.

Usage and related fields. Testing unimodality is not an end in itself. On the contrary, it may be used as a tool for related purposes. Particularly, some clustering algorithms use such approaches to find relevant groups of data. In [17], the dip statistics is used to find clusters in noisy data (skinny-dip algorithm). The authors idea is to find all the bumps in the ecdf because they represent “high” density regions. This information comes precisely from the dip computation. As the dip cannot be computed in the multivariate case, the authors apply it on each dimension, with the issue we raised before.

In [15], Kalogeratos and Likas present a combination of the dip statistics with the \( k \)-means algorithm (\( dip-means \) algorithm). More precisely, the dip test is performed on the pairwise distances of the points within a cluster. Indeed, a cluster is acceptable if it is unimodal (unimodality assumption). If \( k \)-means provides a cluster which is not unimodal, dip-means re-run the algorithm assuming an additional cluster (with an initial state carefully chosen).

Other approaches to estimate the \( k \) of \( k \)-means exist but they are likely to run the clustering algorithm several times with different numbers of cluster so as to optimize a predefined criterion [14]. Examples of criteria are the Bayes Information Criterion (BIC), the Akaike Information Criterion (AIC), the Minimum Message Length (MML) [7], the Minimum Description Length (MDL) [11] or the Gap Statistics [24].

We may think that these tasks are similar with testing unimodality but they are slightly different. First, clustering is stronger because it provides the groups of data while unimodality testing decides whether data are gathered within a single group or not. However clustering needs more information: the number of clusters and/or some parameters to define the similarity between the observations. Thus, unimodality testing may be viewed as a macroscopic description of the data while clustering inspects it deeper.

Definition of unimodality. As presented above, unimodality is well-defined in dimension 1. In a few words, a distribution is unimodal iff the cdf is first convex, then flat, and finally concave. Unfortunately this definition does not generalize in higher dimensions. Thus, several unimodality notions have been developed but in nonequivalent ways. They are rigorously detailed in [5].

In our work, unimodality refers to star unimodality (def 2.1 of [5]). Simply speaking, a density \( f \) is unimodal about the mode \( m \) if \( f \) decreases with the distance to \( m \).

3 CONTRIBUTION

This section presents our theoretical contribution. First we give the general intuition of our test of unimodality (§3.1). The next part (§3.2) deals with the mathematical formulation. It allows us to develop theoretical results (§3.3), leading to our folding test of unimodality (§3.4 and §3.5). Finally we deal with the incremental computation of the required statistics for the test (§3.6).

Notations. In the next parts we will use the following notations: \( \| \cdot \| \) denotes the euclidean norm. In the general case, \( X \) is a random vector of \( \mathbb{R}^d \) \((d \in \mathbb{N}^*)\), we note \( \mathbb{E}[X] \in \mathbb{R}^d \) its expected value. We assume that \( X \) is \( 3 \)-integrable: it means that for all its components \( X_i, \mathbb{E}[X_i^3] \) exists. We write \( \Sigma \in \mathbb{R}^{d \times d} \) its covariance matrix (we may use \( \text{Var} X \) in dimension 1). We assume that \( \Sigma \) is non-degenerated.
is very similar except that the absolute value are replaced by the euclidean norm. Then we will consider $\text{Var} \|X - s^*\|$ where $X$ is a random vector of $\mathbb{R}^d$ and $s^* \in \mathbb{R}^d$ is the pivot.

The figure 3 gives an empirical example of the folding mechanism in two dimensions. Given a trimodal distribution (figure 3a), and the right $s^*$, the folding $X \mapsto \|X - s^*\|$ turns the multi-modal bivariate distribution into a univariate distribution (figure 3b) which is likely to have a “low” variance. But obviously we have to mention about which reference this variance is low.

3.1 General intuition

In this section, we present the general intuition of our approach aimed at testing whether a distribution is unimodal or not.

Univariate case. Here we restrict the description of our approach to univariate distributions. The generalization will be made in the next paragraph.

Let us have a look at a bimodal density (figure 2a). In this case, the variance is likely to be “high” because the two modes make the data far from the expected value. Our idea is the following: if we fold up a mode to the other (with respect to the right pivot $s^*$), the resulting density (figure 2b) will have a far lower variance. Intuitively, this phenomenon will not appear for unimodal distributions (actually not with the same amplitude).

Let us sum up the approach: (1) find the right pivot $s^*$, (2) fold up the distribution along $s^*$, (3) compute the variance of the folded distribution and (4) compare it with the initial variance.

More formally, if we assume we get the pivot $s^*$, the folding step is performed with the transformation $X \mapsto |X - s^*|$, finally we will compute the folding ratio:

$$
\varphi(X) = \frac{\text{Var} \|X - s\|}{\text{Var} X}.
$$

Higher dimensions. How can this approach be generalized? Actually, the transformation made on the distribution (the folding)
observations \( ||X - s_1||, ||X - s_2|| \) and \( ||X - s_3|| \). In this case, we may notice that \( s_1 \) seems to reduce the variance the best (so it may be the best pivot \( s^* \)). Actually, in the unimodal case, the best pivot \( s^* \) is likely to be close to the mode although in the multimodal case, it is likely to stand “between” the modes.

### 3.2 Formal approach

The previous paragraph introduced our main idea. Here, we develop it more formally. We introduce the function \( v_X \) defined by

\[
v_X: \mathbb{R}^d \to \mathbb{R} \;
\text{by}
\begin{array}{c}
\text{Var} \|X - s\|.
\end{array}
\]

By definition, \( v_X \) is lower-bounded by 0, so we can define the folding ratio as below

\[
\phi(X) = \inf_{s \in \mathbb{R}^d} \frac{\text{Var} \|X - s\|}{\mathbb{E} (\|X - \mathbb{E}[X]\|^2)}.
\]

Thus, the computation of \( \phi(X) \) requires a minimization step, which is expensive. The ideal would be to have an expression of \( s^* = \arg \min_{s \in \mathbb{R}^d} v_X(s) \). Unfortunately, its existence is not clear because \( v_X \) can be minimum to the infinity (we have noticed this divergence on some concrete and non absurd configurations).

In our approach, we aim to find a pivot even if the real one is not defined. Obviously this approximate pivot should produce a folding ratio which has intuitively the same properties as the theoretical one \( \phi(X) \). To circumvent the existence problem we rather use the following function:

\[
v_X^{(2)}: \mathbb{R}^d \to \mathbb{R} \;
\text{by}
\begin{array}{c}
\text{Var} (\|X - s\|^2).
\end{array}
\]

If we find the \( s \) which minimizes \( v_X^{(2)} \), we are likely to have a good candidate to minimize \( v_X \). Moreover, the properties of \( v_X^{(2)} \) are richer than the properties of \( v_X \).

**Lemma 3.1.** For all \( s \in \mathbb{R}^d \) we have:

\[
v_X^{(2)}(s) = 4 s^T \Sigma s - 2 s^T \text{Cov}(\|X\|^2, X) + \text{Var}(\|X\|^2).
\]

Lemma 3.1 tells us that \( v_X^{(2)} \) is a quadratic form of \( \mathbb{R}^d \). Thus, thanks to the properties of the covariance matrix \( \Sigma \), \( v_X^{(2)} \) is strictly convex and then it has a unique minimum, noted \( s_X^* \). Theorem 3.1 gives its analytical expression.

**Theorem 3.1.** The function \( v_X^{(2)} \) has a unique minimum \( s_X^* \) given by:

\[
s_X^* = \frac{1}{2} \Sigma^{-1} \text{Cov}(X, \|X\|^2).
\]

In particular, if \( X \) is a real random variable (dimension 1) then

\[
s_X^* = \mathbb{E}[X] + \frac{1}{2} \frac{M_3(X)}{M_2(X)}.
\]

With the previous result, we have always a pivot allowing to compute an approximate of the folding ratio

\[
\bar{\phi}(X) = \frac{\text{Var} \|X - s_X^*\|}{\mathbb{E} (\|X - \mathbb{E}[X]\|^2)}.
\]

The previous expression leads to a straightforward algorithm to compute \( \bar{\phi}(X) \) (algorithm 1). For simplicity reason, we will use “folding ratio” in the next paragraphs to express its approximate version.

![Algorithm 1 Batch computation of \( \bar{\phi}(X) \)](image)

**Input:** \( X \)

**Output:** \( \bar{\phi}(X) \)

\[
D = \mathbb{E} (\|X - \mathbb{E}[X]\|^2) \\
\quad \text{or} \quad D \leftarrow \text{Tr}(\Sigma) \\
s_X^* \leftarrow \frac{1}{2} \Sigma^{-1} \times \text{Cov}(X, \|X\|^2) \\
\text{return} \quad \text{Var} \|X - s_X^*\| / D
\]

We give some details about complexity. Let us consider a dataset of \( n \) observations in \( \mathbb{R}^d \). The covariance matrix \( \Sigma \) needs \( O(n \cdot d^2) \) operations and additional \( O(d^3) \) operations are required for its inversion. Computing the norm of the observations and the covariance \( \text{Cov}(X, \|X\|^2) \) costs \( O(n \cdot d) \) operations. Finally, another \( O(d^2) \) operations are required for \( s_X^* \) and \( O(n \cdot d) \) for the variance \( \text{Var} \|X - s_X^*\| \). In a nutshell, the complexity of the batch computation is linear for the number of observation \( n \) (\( d^3 + n \cdot d^2 + n \cdot d \)).

### 3.3 Unidimensional case study

In this paragraph, we treat the unidimensional case with some common distributions. We show that the approximate folding ratio \( \bar{\phi}(X) \) is relevant to rank the distributions according to their “unimodal character”.

The expression of \( s_X^* \) is very convenient and allows us to calculate analytically its value (and also \( \bar{\phi}(X) \)) for some well-known distributions. We have summarized some of these results in table 1.

<table>
<thead>
<tr>
<th>Distribution of ( X )</th>
<th>( s_X^* )</th>
<th>( \bar{\phi}(X) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential ( E(\lambda) )</td>
<td>( \frac{2}{\lambda} )</td>
<td>( 1 - 4e^{-2} - 4e^{-4} \approx 0.385 )</td>
</tr>
<tr>
<td>Normal ( N(\mu, \sigma^2) )</td>
<td>( \mu )</td>
<td>( 1 - 2 \pi \approx 0.363 )</td>
</tr>
<tr>
<td>Uniform ( U(a, b) )</td>
<td>( \frac{a + b}{2} )</td>
<td>( \frac{1}{\pi} \approx 1 )</td>
</tr>
</tbody>
</table>

**Table 1:** Analytical approximate folding ratio for some common unimodal distributions

The results above show two things: first we can notice that the folding ratios do not depend on the distribution parameters. This property is very interesting because it characterizes a whole distribution class through a single value. Second, if we merely compare the values, the distributions seem to be well-ranked according to their “unimodal character”. Obviously, we concede this is not a well-defined notion (more a visual aspect) but we can sketch what we mean: a distribution is more unimodal when the relative density at its mode is higher.

Table 1 presents only unimodal distributions. Actually, analytical expressions for more complex distributions are more difficult to get. However, we may present a bimodal example: let \( \delta \geq 0 \) and \( X_\delta \sim \frac{1}{2} N(0, 1) + \frac{1}{2} N(\delta, 1) \). It means that \( X_\delta \) follows a bimodal normal distribution with a gap of \( \delta \) between the two modes. In this
In this short study, we have shown that the approximate folding ratio is a relevant statistics to evaluate the “unimodality character”. Furthermore, the uniform distribution seems to be the right reference to make a decision. In the next part, we will use this analysis to build our unimodality test.

3.4 The folding test of unimodality

In the previous parts, we have developed the approximate folding ratio: a relevant statistics to rank the distributions according to their unimodality character. Henceforth, we want to make a decision: is the distribution of $X$ unimodal or not? To answer, we need to compare the folding ratio $\tilde{\varphi}(X)$ to a reference.

As claimed by Hartigan’s work [13] and also observed in the paragraph above, the uniform distribution is likely to be this reference. Indeed, if we consider a sample of size $n$ (from the uniform distribution), we need a larger $n$ to assess its unimodality. More formally, Hartigan and Hartigan have shown that the dip statistics of the uniform distribution is asymptotically larger than other distributions ones, meaning that an empirical sample drawn from the uniform distribution need more observations to be considered as unimodal (worst case of unimodality).

In our work, we generalize this approach for all dimensions $d \in \mathbb{N}$. We consider the uniform distribution within the $d$–ball as the limit case of unimodality. We do not need to precise the radius of this $d$–ball because the folding ratio does not depend on it (see proposition 3.1).

**Proposition 3.1.** Let $R > 0$ and $B_d(R) = \{x \in \mathbb{R}^d \mid \|x\| \leq R\}$ the $d$–dimensional ball of radius $R$. The approximate folding ratio of the uniform random vector $U_d \sim \mathcal{U}(B_d(R))$ is:

$$\tilde{\varphi}_d = \tilde{\varphi}(U_d) = \frac{1}{(d + 1)^2}.$$  

The power of our method relies on its independence to distribution parameters, allowing to catch whole classes through a single value. As of now, we are able to build an unimodality test based on the comparison of folding ratios.

**Definition 3.1 (Folding test of unimodality).** Let $X$ be a 3–integrable random vector of $\mathbb{R}^d$. We define the folding statistics $\Phi(X)$ by

$$\Phi(X) = \tilde{\varphi}(X) = (1 + d)^2 \tilde{\varphi}(X),$$  

leading to the folding test of unimodality: if $\Phi(X) \geq 1$, the distribution of $X$ is unimodal, if $\Phi(X) < 1$ it is multimodal.

3.5 Statistical significance

**Theory.** Obviously, we do not have the same level of confidence in the test whether we have 100 or 1 billion observations. Here we precise the decision bounds when the sample size is $n$.

Let us take a sample $X_1, \ldots, X_n \in \mathbb{R}^d$. We have seen how we can compute $\Phi(X_1, \ldots, X_n)$. According to this value we can decide whether the distribution is unimodal or multimodal. Now, let us imagine that the sample $U_1, \ldots, U_n$ is drawn from the uniform distribution: the test becomes very uncertain because the computed folding statistics would be close to 1. So we may understand that the test is more significant if we are far from the uniform case.

This classical statistical hypothesis testing problem leads us to provide $p$–values. For instance, let $q > 0$ and let us assume we have computed $\Phi(X_1, \ldots, X_n) = 1 - q < 0$, so the distribution is considered as multimodal. The probability to have a uniform example $U_1, \ldots, U_n$ with a lower folding statistics is given by:

$$\mathbb{P}(\Phi(U_1, \ldots, U_n) < 1 - q) = \mathbb{P}(1 - \Phi(U_1, \ldots, U_n) > q).$$

Conversely, we can imagine that we have computed $\Phi(X_1, \ldots, X_n) = 1 + q$. In this case, we can estimate the probability to have a uniform sample with a higher folding statistics:

$$\mathbb{P}(\Phi(U_1, \ldots, U_n) > 1 + q) = \mathbb{P}(\Phi(U_1, \ldots, U_n) - 1 > q).$$

Obviously we want these two probabilities to be low. It means that the $p$–value $p = \mathbb{P}(|\Phi(U_1, \ldots, U_n) - 1| > q)$ must be as low as possible. In a nutshell, if we want a significant test (usually $p \leq 0.05$), it implies to choose $q$ high enough, making the uncertainty area $1 \pm q$ wider. Conversely, if the test outputs a value $\Phi(X_1, \ldots, X_n)$, it leads to a decision whose significance can be estimated by $p$ (the lower it is, the more significant it will be).

**Numerical quantiles.** The ideal would be to know the distribution of $Y_{d,n} = |\Phi(U_1, \ldots, U_n) - 1|$. However $Y_{d,n}$ does not follow a common distribution, leading us to compute some quantiles with Monte-Carlo simulations. Table 2 presents the quantiles $q$ according to...
to $d$ and $n$ at a significance level $p = 0.05$ (10000 simulations for each tuple with Marsaglia’s sampling method [16]).

<table>
<thead>
<tr>
<th>$n/d$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.22</td>
<td>0.28</td>
<td>0.33</td>
<td>0.35</td>
<td>0.38</td>
</tr>
<tr>
<td>200</td>
<td>0.15</td>
<td>0.2</td>
<td>0.23</td>
<td>0.25</td>
<td>0.27</td>
</tr>
<tr>
<td>500</td>
<td>0.1</td>
<td>0.13</td>
<td>0.15</td>
<td>0.16</td>
<td>0.17</td>
</tr>
<tr>
<td>1000</td>
<td>0.07</td>
<td>0.09</td>
<td>0.1</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>2000</td>
<td>0.05</td>
<td>0.06</td>
<td>0.07</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>5000</td>
<td>0.03</td>
<td>0.04</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>10000</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>20000</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 2: Quantiles $q$ according to $d$ and $n$ at significance level $p = 0.05$

For example, let us take a dataset of 1000 observations in 2 dimensions $X_1, \ldots, X_d$ ($n = 1000$, $d = 2$). The distribution is considered significantly unimodal (at level $p = 0.05$) if $\Phi(X_1, \ldots, X_d) \geq 1.09$ (and significantly multimodal if it is lower than 0.91). Given a set of similar tables for different values of $p$, we can infer the significance of any test output $\Phi$.

Obviously we notice that the higher is $n$, the tighter are the bounds but the dimension increases the uncertainty. Unfortunately, we cannot provide reliable quantiles for higher dimensions with $X_{cov}$ (and significantly multimodal if it is lower than $V_{inv}$).

In the paragraph 3.2 we proposed a batch (or offline) version for the computation of $\bar{\Phi}(X_1, \ldots, X_n)$. However, in the streaming context, we can accelerate it with an incremental computation of $s_2^2$.

As seen in the Section 3.2, we have an analytical expression for the pivot: $s^2 = \frac{1}{n} X_{cov} ||X||^2$. Furthermore this expression can easily be computed incrementally or even updated over a sliding window. This is a very important property if we work on streaming data. The algorithm 2 details how to perform an update.

As an input we have a new incoming observation $X_{new}$ and the square of its norm $R_{new}$. The idea is to compute their respective basic moments (cumulative sums $S_X$ and $S_R$) and their co-moment (line 4) to finally compute an estimate of $\text{Cov}(X, ||X||^2)$ (line 7). About the inverse of the covariance matrix, we use the Sherman-Morrison formula [21] to compute the inverse of the co-moment of $X$ (line 5) and then $\Sigma^{-1}$ (line 6). We recall:

$$\text{SM}(A^{-1}, u, v) = \left( A + u \times v^T \right)^{-1} = A^{-1} - \frac{A^{-1} \times u \times v^T \times A^{-1}}{1 + v^T \times A^{-1} \times u}.$$

Here we present an update, so we have to deal with the initialization. Basically, the variables $n, S_X, S_R, V_{X,R}$ and $C$ are set to 0. Unfortunately, the initialization of $V_{X}^{inv}$ and $\Sigma^{inv}$ is not as simple because it requires several observations $X_i$ (at the beginning $V_{X}^{inv}$ and $\Sigma^{inv}$ are singular matrices). In practice, we start from a small batch of data $X_{init}$. Therefore, we can compute $V_{X} \leftarrow X_{init}^2 \times X_{init}$ and then $V_{X}^{inv} \leftarrow V_{X}^{-1}$ (so $\Sigma^{inv}$). Finally, this update can easily be extended to sliding window requiring to store the quantities $X_i$ and $R_i$, but here, we prefer not to overload the reader with additional formulas.

Is it possible to go further by computing $\bar{\Phi}(X)$ incrementally too? Unfortunately, this is not as straightforward. There is no problem for the denominator $\mathbb{E} \left[ ||X - \mu||^2 \right]$ because this is the trace of $\Sigma$.

The main issue comes from $\text{Var} ||X - s_2||$: a variance can easily be computed incrementally (like in the algorithm 2) but $s_2$ changes at each iteration, so an update of all the distances $||X_i - s_2^2||$ is needed.

In the paragraph 3.2, we discussed about the complexity of the batch computation. In table 3, we compare its complexity with the incremental computation in 3 cases:

- single computation: only $\bar{\Phi}(X_1, \ldots, X_n)$
- cumulative: $\bar{\Phi}(X_1), \bar{\Phi}(X_1, X_2), \ldots, \bar{\Phi}(X_1, \ldots, X_n)$
- sliding window: $\bar{\Phi}(X_1, \ldots, X_{1+\ell}), \ldots, \bar{\Phi}(X_{n}, \ldots, X_{n+\ell})$

<table>
<thead>
<tr>
<th>Computation(s)</th>
<th>Batch</th>
<th>Incremental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>$d^3 + n \cdot d^2$</td>
<td>$(+ n \cdot d)$</td>
</tr>
<tr>
<td>Cumulative</td>
<td>$n \cdot d^3 + n^2 \cdot d^2$</td>
<td>$d^3 + n \cdot d^2 + n^2 \cdot d$</td>
</tr>
<tr>
<td>Sliding window</td>
<td>$n \cdot d^3 + n \cdot w \cdot d^2$</td>
<td>$d^3 + n \cdot d^2 + n \cdot w \cdot d$</td>
</tr>
</tbody>
</table>

Table 3: Complexity of each method (in big O notation)

Without delving into the details, we may notice two phenomena. First, the incremental computation of $s_2^2$ decreases the dependency on the dimension $d$ (actually the update of the matrix costs $O(d^2)$ instead of $O(d^3)$ in the batch version). Second, as the computation of $\text{Var} ||X - s_2||$ cannot be done efficiently, the computation of the folding statistics cannot be cut down any more.

4 EXPERIMENTS

In this section, we highlight the relevance of the folding test of unimodality. We experiment it on real world multidimensional data to emphasize its correctness and its practical uses. Particularly, we show that it provides a paramount statistical description necessary for data analysis. We make our python3 code available for reviewers.

4.1 Should I try to cluster Pokémon?

In this section, we will show that our test is able to avoid a useless clustering step. Particularly, we analyze the Pokémon Stats

1 git clone https://scm.gforge.inria.fr/anonscm/git/sharedcode/sharedcode.git
Dataset from kaggle\(^2\). This dataset gathers statistics of the Poké-
mon until the 6\(^{th}\) generation: it includes 21 variables per each of the
721 Pokémons. In this short experiment, we keep only the 6
basic fight statistics: *Attack*, *Defense*, *HealthPoints*, *SpecialAttack*,
*SpecialDefense* and *Speed*.

A priori, in such a 6-dimensional space, this is not easy to claim
whether some clusters arise. Either all the features pairs could be
plotted, or a clustering algorithm could be run. Through the first
method (figure 6), the distributions of the features and their pairs
lead us to think that the whole distribution is rather unimodal.

When we compute the folding statistics to this dataset, we get
\(\Phi(X) = 5.04\). As guessed above, this result claims that the distribu-
tion is unimodal.

Figure 6: Pairs-plot of the Pokémon six fight characteristics

To show that our approach is relevant, we compare the result of
our test with the output of some well-known clustering algorithms.
Unfortunately, most of them needs to set precisely the number of
clusters, so we only test the approaches having some procedures to
find it.

First we try to model the data with gaussian mixtures. The num-
ber of clusters is found by minimizing either the Akaike’s or the
Bayesian Information Criterion (AIC, BIC). Then we use the *gap*
statistics [24] of Tibshirani *et al.*, which is a common statistics to
estimate the number of clusters. We test also the Mean Shift al-
gorithm [2] with different quantiles \(q \in [0.3, 0.8]\). This parameter
tunes the "similarity" of the points (it sets the bandwidth of the un-
derlying gaussian kernel). Finally, we run the Affinity Propagation
algorithm [8] with the euclidean distance as similarity. For these
two last approaches, we show only "stable" results, so we do not
present neither absurd parameter choices (the mean shift algorithm
with \(q \to 1\) would make all the points similar) nor absurd outputs
of some algorithms (the affinity propagation algorithm with the
gaussian similarity did not output less than 49 clusters). Finally we
compute the average silhouette score [19] to estimate the clustering
quality (the closer to 1, the better).

The results are given in the table 4. We can notice that there is
no consensus among the different methods: they do not output
the same number of clusters. At the very least, this indicates that
the dataset does not contain obvious clusters. Now regarding the
returned clusters, the average silhouette scores are mostly low, even
close to zero, meaning that these clusters are not well separated.
The highest silhouette score is reached by Mean Shift, however the
algorithm returned 3 clusters of size 717-3-1, so in practice it found
only one cluster. These results are in favor of the unimodality of
the data, agreeing with the result of our folding test of unimodality.

Before running clustering algorithms, our simple test can thus
be a crucial step to decide whether clustering is relevant or not.

4.2 Stock market behaviors

The aim of this section is twofold. Firstly, we analyze the behav-
ior of the folding statistics over a sliding window. Secondly, we show
how the folding test of unimodality can help the clustering
parametrization step.

*catching the behavior jumps*. In this experiment we analyze a
part of the DJIA 30 Stock Time Series dataset hosted on kaggle\(^3\). In
particular we study the historical stock data from NYSE (between
2006-01-03 to 2017-29-12) of four major companies: Cisco (CSCO),
Intel (INTC), IBM and Microsoft (MSFT). We consider the highest
price daily reached, so we get a dataset with \(n = 3018\) observations
in dimension 4 (number of companies).

The figure 7 shows the evolution of the stock prices for these
four companies. We may notice two different behaviors: IBM stocks
vary more than Cisco, Intel and Microsoft ones.

Now, we compute the folding statistics \(\Phi\) over a sliding window
of size 650 (about two years and half, because there is no data on
Saturday and Sunday). It means that at every iteration, we take a
snapshot of 650 observations in dimension 4, and we compute \(\Phi\)
on it. The results are presented in the figure 8. We add significan-
cess bounds at level 0.05 (\(q = 0.15\)). One can roughly see three "moments
of unimodality" (around iterations 500, 1500 and 1800).

What does it mean? If the distribution at these moments is uni-
modal, it means that the companies stocks are individually quite
the same: their behavior is stationary within the window. At the

\(^2\)https://www.kaggle.com/alopez247/pokemon
\(^3\)https://www.kaggle.com/zzrlee/stock-time-series-20050101-to-20171231
When the four stocks are the most stationary, (figure 10), we can confirm these behaviors (to ease the plot we have overlaid the scatter-plots IBM vs MSFT, IBM vs CSCO and IBM vs INTC). On figure 10a, we can observe about three clusters whereas no clusters really arises on the figure 10b. However we clearly see three little groups of data on this latter figure. In fact, they are not statistically substantial (not real modes).

Let us analyze the unimodal area around iteration 1500. All the instances of DBSCAN output nearly a single cluster. But around it, 1800 (unimodal area too), they all return 2 clusters. In the first unimodal zone (around it, 500) DBSCAN($r = 7$) is the only one not returning one cluster. Eventually, if we look at the first multimodal area (before it, 330), only the instances DBSCAN($r = 7$) and DBSCAN($r = 8$) output several clusters. We may extend the analysis to all the regions, making the instance DBSCAN($r = 8$) the more faithful to the folding test of unimodality.

Running time. We compare the running time of two methods: computing the statistics on the complete window each time (batch computation), or using the incremental approach presented in section 3.6. The complete computation (2400 iterations) took 5.5 seconds with batch computation, and 4.5 seconds with the incremental algorithm (around 18% acceleration). Note that the variance computation part is identical in both methods: if we only compare the computation of $\Phi$, the times are 3.4 and 0.9 seconds, giving a 73% decrease on running time. We can conclude that even on this relatively simple data (low dimension, few iterations), the incremental computations has a significant impact on running time.

Helping clustering algorithms. We have mentioned in the Poké-mon experiment (section 4.1) that unimodality testing provides a different piece of information from clustering. In that case, it was relevant to avoid over-clustering. Here, we use it as a parametrization support for clustering.

Let us consider the DBSCAN algorithm [6]. It requires two parameters: the neighborhood radius $r$ and the minimum requested numbers of neighbors to be considered as a core point $m$. Our idea is to tune the radius parameter so as to make DBSCAN output “1 cluster” only when $\Phi > 1$. Obviously, we cannot ensure the quality of the clustering in the other regions however with this criterion we can narrow down the parameter research domain.

An example with $m_r = 80$ (12.5% of the observations in the window) is presented on the figure 11. We plot the folding statistics (top) and the number of clusters output by DBSCAN with different neighborhood radius ($r = 6, 7, 8, 9$ and $10$). Moreover we add all the unimodal areas (where $\Phi > 1 + q$) and the first multimodal area (where $\Phi < 1 - q$).

Let us analyze the unimodal area around iteration 1500. All the instances of DBSCAN output nearly a single cluster. But around it, 1800 (unimodal area too), they all return 2 clusters. In the first unimodal zone (around it, 500) DBSCAN($r = 7$) is the only one not returning one cluster. Eventually, if we look at the first multimodal area (before it, 330), only the instances DBSCAN($r = 7$) and DBSCAN($r = 8$) output several clusters. We may extend the analysis to all the regions, making the instance DBSCAN($r = 8$) the more faithful to the folding test of unimodality.
This paper introduces the folding test of unimodality. To the best of our knowledge, this is the first time a multivariate and purely statistical unimodality test has been proposed. The test is based on a new folding statistics, which provides a score related to the “level of unimodality” of a distribution. This statistics does not depend on the parameters of the input distribution. The only parameter of the unimodality test is a natural \( p \)–value giving the desired significance level of the result.

Among the perspectives opened by this work, we envision to exploit our folding statistics to discover \( k \) values for the \( k \)–means algorithm. This was explored before in the dip-means approach, by using unidimensional unimodality tests on all pairwise distances. Using our folding unimodality test allows to immediately validate each cluster of \( k \)–means in its multi-dimensional space (unimodality assumption). This significantly reduces the complexity compared to dip-means.

5 CONCLUSION

We insist on the non-completeness of our test. We cannot claim that DBSCAN gives the correct clusters in the multimodal regions. Moreover, we know that other phenomenons should be taken into account for clustering like the influence of the second parameter and/or data normalization. We merely show that our lightweight test is able to provide some paramount information about data. The latter may be useful for a potential clustering stage.

REFERENCES


Figure 11: Number of clusters output by DBSCAN according to the radius parameter \( r \)