SmokeOut: An Approach for Testing Clustering Implementations

Vincenzo Musco
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
vincenzo.a.musco@njit.edu

Xin Yin
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
xy258@njit.edu

Iulian Neamtiu
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
ineamtiu@njit.edu

***Abstract*—Clustering is a key Machine Learning technique, used in many high-stakes domains from medicine to self-driving cars. Many clustering algorithms have been proposed, and these algorithms have been implemented in many toolkits. Clustering users assume that clustering implementations are correct, reliable, and for a given algorithm, interchangeable. We challenge these assumptions. We introduce SmokeOut, an approach and tool that pits clustering implementations against each other (and against themselves) while controlling for algorithm and dataset, to find datasets where clustering outcomes differ when they shouldn't, and measure this difference. We ran SmokeOut on 7 clustering algorithms (3 deterministic and 4 nondeterministic) implemented in 7 widely-used toolkits, and run in a variety of scenarios on the Penn Machine Learning Benchmark (162 datasets). SmokeOut has revealed that clustering implementations are fragile: on a given input dataset and using a given clustering algorithm, clustering outcomes and accuracy vary widely between (1) successive runs of the same toolkit; (2) different input parameters for that tool; (3) different toolkits.**

***Index Terms*—Clustering, Machine Learning, Differential Testing, Software Reliability**

I. Introduction

Cluster analysis, a.k.a. *Clustering*, is an unsupervised learning technique used to group together entities that are related or share similar characteristics. Clustering has many high-stakes applications: medicine/disease prediction [1]–[3], self-driving cars [4], criminology/criminal justice [5], finance [6], etc. End-users that run such applications (or are affected by decisions made with the support of such applications) should be able to assume that the applications are reliable.

Moreover, clustering (and Machine Learning in general) is seeing increased adoption in software products, so it is imperative that clustering implementations be reliable. Prior research efforts have produced many clustering algorithms, and these algorithms have been implemented in numerous toolkits, but prior work has not questioned or investigated the clustering implementations' correctness or reliability. For example, developers use clustering implementations as "black boxes" and might over-optimistically assume that algorithms' implementations are correct, accurate, and generally reliable. However, even specifying clustering correctness remains a challenge, which complicates validating or verifying clustering implementations. We are not aware of any study that has questioned the reliability of clustering implementations, or tool for testing such aspects.

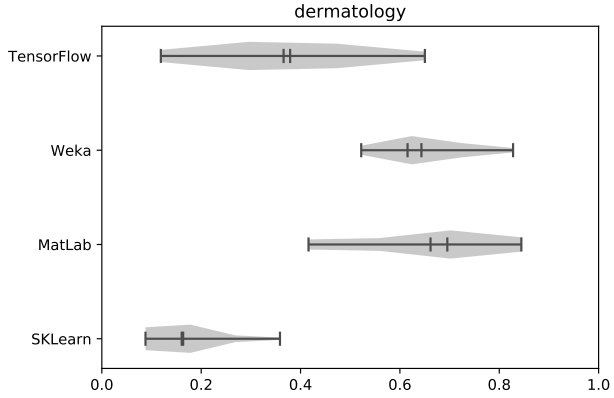

Fig. 1. Expectation Maximization: Accuracy Distributions for 4 Toolkits On Dataset dermatology.

Example: clustering accuracy in dermatology. Consider the dermatology dataset [7] used for "differential diagnosis of erythemato-squamous [skin] diseases". Güvenir et al. [8] have used this dataset to construct a classification approach for diagnosing new patients; they explicitly state "The main requirement for such a system is prediction accuracy". We set the algorithm to the widely used (and in our experiments, the most accurate across-the-board) Expectation Maximization algorithm, and ran 4 different toolkits implementing the algorithm (TensorFlow, Weka, Matlab, SKLearn) to cluster this dataset; each toolkit was run 30 times. In Figure 1 we show toolkits' accuracy distributions. We make two observations: (1) there are wide variations across runs for the same toolkit, e.g., from 0.1 to 0.7 for TensorFlow; (2) all of SKLearn's runs, including its "best" runs at 0.35 accuracy, are *worse* than any of Matlab or Weka's runs, whose minimum accuracies were 0.41 and 0.52, respectively. Note that, while *variation* across runs is expected for nondeterministic algorithms, *consistently poor performance* is problematic.

Therefore, there is a need for reliable clustering implementations. Users, from life science researchers to software engineers should be able to (reasonably) assume that clustering implementations are reliable and interchangeable, i.e., for a given algorithm, its implementation is correct and has no negative impact on the clustering outcome.

We introduce SmokeOut,[1] a tool that leverages the wide availability of clustering implementations and datasets with ground truth to test clustering implementations (while controlling for datasets and algorithms). Crucially, SmokeOut does not require an explicit specification associated with an implementation. SmokeOut uses a suite of differential clusterings coupled with a statistics-driven approach to help developers measure the determinism and accuracy (absolute, as well as relative to other toolkits) of a given implementation. Section II describes SmokeOut's architecture.

In Section III we describe the setup. SmokeOut's input consisted of 162 datasets from the Penn Machine Learning Benchmark (described in Section III-A). To measure accuracy, we use the Adjusted Rand Index (ARI), ranging from -1 (worst, or lowest accuracy) to $+1$ (best, or highest accuracy); note that $ARI = 0$ corresponds to "independent" clusterings when comparing toolkits or roughly random clustering when comparing to Ground Truth; ARI details are in Section III-B.

We chose 7 widely-used clustering toolkits: MATLAB, mlpack, R, Scikit-learn, Shogun, TensorFlow, WEKA (Section III-C). However, some algorithms are not implemented by all 7 toolkits: in total, we have 27 algorithm/toolkit combinations. We analyzed 7 widely-used clustering algorithms, described in Section III-D. Of these, 3 are deterministic: Hierarchical clustering - agglomerative, Affinity Propagation, and DBSCAN. The other 4 are nondeterministic: K-means and its K-means++ variant, Spectral Clustering, and Expectation-Maximization (Gaussian Mixture).

To characterize clustering outcomes and present the results in an intuitive way, we introduce a concise, yet effective and statistically rigorous, 5-label system that captures distribution shapes (Section IV).

Section V presents the SmokeOut results. We now present a few highlights for our findings:

Deterministic algorithms have non deterministic implementation across toolkits. Hierarchical clustering is a completely deterministic procedure and there should be no variation in the results obtained by a given implementation for a fixed dataset. However, SmokeOut has revealed toolkits disagreements where there should be no disagreement.

Non-deterministic algorithms have a wide range of outcomes: the variations across toolkits and variation across runs can be severe. While variation across runs is expected for nondeterministic algorithms, consistently poor performance is problematic. Many toolkits achieve $ARI = 1$, while some toolkits' best runs are around 0 (i.e., random).

Different implementations of the same algorithm cluster points differently. Note that similar accuracy does not imply similar clusters; toolkits often disagree on cluster composition.

II. SmokeOut Architecture

Figure 2 shows SmokeOut's architecture. Let CTT be a new "Clustering Toolkit under Test" (bottom left of the figure), that is, an implementation of a specific clustering algorithm. CTT is tested as follows:

[1] https://github.com/v-m/SmokeOut

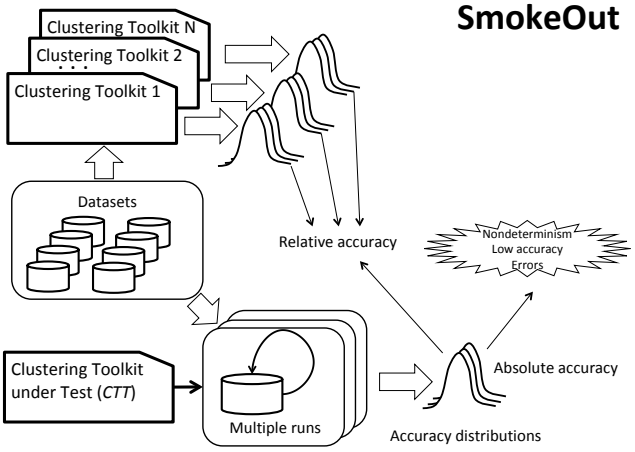

Fig. 2. SmokeOut architecture.

1) CTT is run multiple times on the same dataset to gauge (potential) non-determinism. For this we run statistical analyses on the accuracy distributions.
2) CTT's accuracy distributions are compared with other implementations of CTT's algorithm ("Clustering Toolkit 1" ... "Clustering Toolkit N"); which allows us to measure CTT's relative accuracy, or compare accuracy when ground truth is not available (e.g., via measures such as objective or silhouette).

III. Setup

Clustering is defined as follows. Given a set S of n points (d-dimensional vectors in the $\mathcal{R}^d$ space), the objective of clustering is to partition S into K non-overlapping subsets (clusters) $S_1, \ldots, S_i, \ldots, S_K$ such that intra-cluster distance between points (that is, within individual S_i's) is minimized.

A. Datasets

Appropriate datasets are crucial when benchmarking Machine Learning implementations. To that end, we chose PMLB (Penn Machine Learning Benchmark) [9], a benchmark suite carefully designed to include representative datasets, suitable for evaluating ML implementations.[2] PMLB is a collection of 166 datasets, of which we used 162; we excluded connect-4, poker, mnist, and kddcup due to their excessive size – running these hundred of times would be prohibitive.

The following table contains descriptive statistics: datasets have, on average, 809 instances (points to be clustered) and the mean number of features (number of attributes, or dimensions d) is 15. PMLB comes with Ground Truth, which allows us to measure clustering accuracy. About half the datasets have two clusters ($K = 2$), while for the rest we have $3 \leq K \leq 26$.

	Min	***Max***	***Geometric Mean***
Instances	32	105,908	809.25
Features (attributes)	2	1,000	15.41
K (# of clusters)	2	26	3.18

[2] PMLB was specifically designed to, among others, avoid pitfalls of other publicly available datasets as well as to "compare and contrast ML methods" [9].

TABLE I
CATEGORIES FOR THE PMLB DATASETS.

Category	*Percentage*
Medical/Health	24%
Biology, Biochemistry, Bioinformatics	15%
Physics, Math, Astronomy	11%
Social, Census	10%
Sports	7%
Financial	7%
Image recognition	6%
Synthetic datasets	6%
IT, AI	4%
Linguistics	3%
Miscellaneous	7%

We categorized the nature of each dataset and present the category breakdown in Table I. We point out several things: the datasets are quite representative, as they cover a wide range of domains, from scientific to social to financial; medical data (discussed next) has the highest proportion, 24%; and the presence of synthetic datasets, 6%, to increase the variety of data density distributions.

To illustrate the need for clustering reliability, we note that 38 of the real-world datasets in PMLB are clustering tasks from the medical/health domain, e.g., contain patient data and outcomes. For example, four datasets are dedicated to breast cancer; three are focused on heart disease; other datasets involve predicting diabetes, hypothyroidism, appendicitis, etc.

B. Measuring Accuracy

The *adjusted Rand index* (ARI), introduced by Hubert and Arabie [10] is an effective and intuitive measure of clustering outcomes: it allows two different partitioning schemes of an underlying set D to be compared. Multiple surveys and comparisons of clustering metrics have shown that ARI is the most widely used [11], most effective, as well as very sensitive [12]. Concretely, assuming two clusterings (partitionings) U and V of S, the ARI measures how similar U and V are. The ARI varies between -1 and $+1$, where $ARI = +1$ indicates perfect agreement, $ARI = 0$ corresponds to independent/random clustering, and $ARI = -1$ indicates "perfect disagreement", that is, completely opposite assignment.

C. Toolkits

We chose 7 widely-used ML toolkits: MATLAB, mlpack, R, Scikit-learn,[3] Shogun, TensorFlow, WEKA. The popularity of these toolkits is apparent in many ways: multi-million user bases, e.g., MATLAB and R[4]; TensorFlow's 1,600+ GitHub contributors [15] or the abundance of S&P 500 companies that use TensorFlow [16]; Scikit-learn is used by popular services such as Spotify, Evernote, or Booking.com [17]; and so on.

D. Algorithms

We chose 3 deterministic clustering algorithms: Hierarchical clustering ("agglomerative" variant), DBSCAN,[5] and Affinity Propagation. *For a given dataset, the clustering outcomes for these algorithms are not supposed to vary across runs, or across toolkits.*

We chose 4 nondeterministic algorithms: K-means and its K-means++ variant; Spectral Clustering, and Expectation-Maximization (Gaussian Mixture). *The clustering outcomes for these algorithms are expected to vary across runs or toolkits, but consistently poor performance is problematic.*

[3]Scikit-learn and R use "Gaussian kernel" density by default. In addition, Scikit-learn can also use k-nearest neighbors, a faster scheme, hence we use the term "SKlearnFast" to refer to this implementation. We use the term "SKlearn0T" to denote Scikit-learn's zero-tolerance configuration.

[4]MATLAB: more than 3 million users in 2017 [13]. R: over 2 million users in 2014 [14].

[5]While in rare scenarios DBSCAN could be "mildly" nondeterministic due to input order, we used a fixed input order to ensure determinism.

IV. DISTRIBUTION SHAPES

Since clustering implementations are used as "black boxes", we want to give users an idea of what to expect from a certain toolkit or algorithm: *will clustering performance be consistently good? will it be consistently bad? will it be mostly good with an occasional "bad" run? will it be mostly bad with an occasional "good" run? will it be good for half the runs, and bad for the other half?*

There are many statistical parameters that characterize a distribution, but no single parameter to give us the answers to the previous questions. To this end, we introduce a simple, concise, five-label system that can succinctly characterize a distribution along the lines drawn above. The labels capture distribution shapes (Figure 3) and are defined as follows:

- **R**, which stands for outliers to the Right of the distribution; that is, clustering accuracy can sometimes be high; put prosaically, some runs are "good".
- **L**, which stands for outliers to the Left of the distribution; that is, clustering accuracy can sometimes be low; put prosaically, some runs are "bad".
- **LR**, when both good and bad outliers exist.
- **B**, i.e., Bimodality – the distribution is bimodal, where a set of values is low and one is high.
- **U**, aka Uniform values – no outliers.

V. RESULTS

A. SmokeOut Methodology

SmokeOut was run 30 times for each algorithm, so we can draw meaningful statistical conclusions; we used default settings for all toolkits. In all, across all algorithms and toolkits, there were 152,276 runs. We use the following format: for each of these 30 runs, we obtain 30 clustering outcomes. We compare these clusterings against Ground Truth, and measure the ARI. Next, we characterize the ARI distribution by indicating the *min* value, the *max* value, and the *shape* (that is, one of **B**, **R**, **L**, **LR**, **U**). Let us take the first row of Table II as an example, where K-means was run on dataset collins using SKlearn, R, MLpack, MATLAB, Shogun and TensorFlow. For SKlearn, across the 30 runs, we observed a minimum accuracy of 0.54, a maximum accuracy of 0.7, and the distribution shape is **LR** (both left and right outliers). That is, the expected accuracy is in the interval $[0.54, 0.7]$, with both left and right outliers possible. The next row, confidence, however, has a bimodal distribution with minimum 0.36 and maximum 0.71 hence running the toolkit repeatedly will yield accuracy values either in the neighborhood of 0.36 or in the neighborhood of 0.71, i.e., a $2\times$ variation from run to

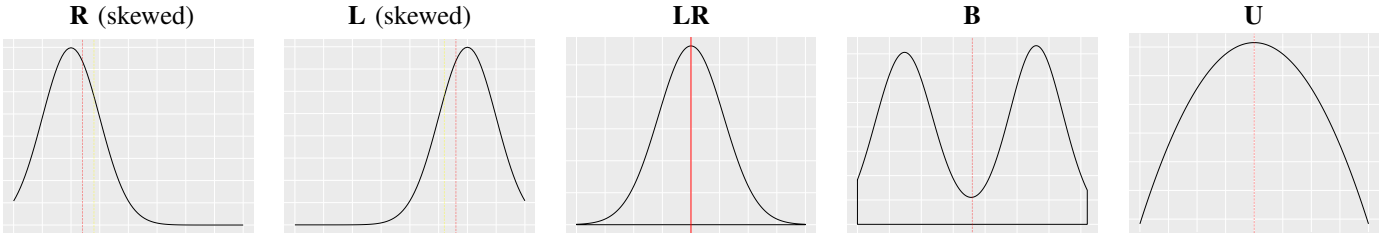

Fig. 3. Distribution shapes and their corresponding labels; the red dotted vertical line indicates the median while the yellow dotted line is the mean.

TABLE II
K-MEANS: VARIATION DUE TO STARTING POINTS

Dataset	SKlearn			R			R100iter			MLpack			Matlab			Shogun			TensorFlow			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
collins	.54	.70	LR	.56	.65	B	.54	.70	LR	.54	.70	LR	.54	.70	LR	.54	.70	LR	.54	.70	LR	.50	ALL
confidence	.36	.71	B	.36	.71	B	.36	.71	B	.36	.71	B	.36	.71	B	.36	.71	B	.36	.71	B	.30	ALL
corral	0	.38	B	0	.38	B	0	.38	B	-.01	.38	B	0	.38	B	0	.38	B	0	.38	B	-.08	ALL
ecoli	.47	.70	B	.47	.69	B	.47	.70	B	.47	.70	B	.47	.70	B	.47	.70	B	.47	.70	B	.51	ALL
house-votes-84	-.02	.54	B	.52	.54	U	-.02	.54	B	-.02	.54	B	-.02	.54	B	-.02	.54	B	-.02	.54	B	0	ALL
iris	.41	.73	B	.41	.78	B	.41	.73	B	.41	.73	B	.41	.73	B	.41	.73	B	.41	.73	B	.41	ALL
mfeat-karhunen	.48	.76	LR	.47	.76	R	.48	.76	LR	.48	.76	LR	.48	.76	LR	.48	.76	LR	.48	.76	LR	.42	ALL
mfeat-pixel	.50	.78	LR	.51	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.41	MT/R
monk3	.09	.39	B	.09	.39	B	.09	.39	B	.09	.39	B	.09	.39	B	.09	.39	B	.09	.39	B	-.01	ALL
mushroom	0	.37	B	0	.37	B	0	.37	B	0	.37	B	0	.37	L	0	.37	L	0	.37	L	-.01	ALL
new-thyroid	.16	.60	B	.21	.60	B	.16	.60	B	.16	.60	B	.16	.60	B	.16	.60	B	.16	.60	B	.14	ALL
optdigits	.57	.75	LR	.52	.75	LR	.57	.67	B	.57	.75	LR	.57	.75	LR	.57	.75	LR	.57	.75	LR	.55	MT/R/R1/S/SH/T
promoters	1	1	U	1	1	U	1	1	U	1	1	U	1	1	U	1	1	U	1	1	U	1	ALL
shuttle	.17	.45	B	.13	.45	LR	.24	.45	B	.24	.45	B	.24	.45	B	.24	.45	B	.24	.45	B	.19	MT/R/R1/SH/T
solar-flare_1	.08	.45	LR	.08	.45	LR	.08	.45	LR	.08	.45	LR	.08	.45	LR	.08	.45	LR	.08	.45	LR	.19	ALL
solar-flare_2	.12	.54	LR	.12	.54	LR	.12	.54	LR	.12	.57	LR	.12	.54	LR	.12	.54	LR	.12	.54	LR	.24	ALL
waveform-40	.25	.46	B	.25	.25	U	.25	.25	U	.25	.46	B	.25	.46	B	.25	.46	B	.25	.46	B	.37	ALL
soybean	.29	.49	LR	.29	.44	B	.29	.49	B	.29	.49	LR	.29	.49	LR	.29	.49	LR	.29	.49	LR	.47	ALL
satimage	.28	.52	B	.30	.53	B	.28	.52	B	.28	.52	B	.28	.52	B	.28	.52	B	.28	.52	B	.38	MP/MT/R/R1/SH/T
pendigits	.48	.62	B	.47	.62	B	.48	.62	B	.48	.62	B	.48	.62	B	.48	.62	B	.48	.62	B	.52	MT/R/R1/S/SH/T
mofn-3-7-10	-.06	.38	B	-.06	.38	B	-.06	.37	B	-.06	.38	B	-.06	.38	B	-.06	.38	B	-.06	.38	B	-.04	ALL
mfeat-pixel	.50	.78	LR	.51	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.50	.78	LR	.41	MT/R
led7	.31	.49	LR	.31	.50	LR	.23	.49	LR	.23	.49	LR	.22	.49	LR	.31	.49	LR	.22	.49	LR	.23	MP/MT/R1/S/SH/T
haberman	-.01	-.01	U	-.01	.01	U	-.01	-.01	U	-.01	.17	B	-.01	-.01	U	-.01	-.01	U	-.01	-.01	U	-.01	ALL
flags	-.01	.15	B	0	.11	LR	-.01	.35	B	-.02	.15	B	-.01	.15	B	-.01	.15	B	-.01	.15	B	-.04	MP/MT
dna	.32	.45	B	.24	.47	LR	.33	.45	B	.32	.45	B	.32	.45	B	.32	.45	B	.32	.45	B	.54	MT/R
balance-scale	.04	.29	LR	.04	.29	LR	.04	.29	LR	.04	.33	LR	.04	.29	LR	.04	.29	LR	.04	.29	LR	-.01	ALL
Median	*.28*	*.52*		*.29*	*.53*		*.25*	*.52*		*.25*	*.52*		*.25*	*.52*		*.28*	*.52*		*.25*	*.52*		*.30*	
Geometric Mean	*.25*	*.52*		*.27*	*.51*		*.25*	*.52*		*.25*	*.53*		*.25*	*.52*		*.25*	*.52*		*.25*	*.52*		*.26*	
Median (all)	*.01*	*.07*		*.01*	*.07*		*.01*	*.07*		*.01*	*.07*		*.01*	*.07*		*.01*	*.07*		*.01*	*.07*		*.30*	
Geometric Mean (all)	*.09*	*.16*		*.09*	*.16*		*.09*	*.16*		*.09*	*.17*		*.09*	*.16*		*.09*	*.16*		*.09*	*.16*		*.26*	

run! Note that the numbers discussed so far compare the clustering outcome against Ground Truth. The final set of columns, "T.A." (toolkit agreement),[6] indicates the ARI when comparing toolkits against each other; in other words, we want to know if toolkits agree with each other (though they might disagree with Ground Truth). This is computed pairwise, toolkit vs. toolkit, hence we have $30 \times 30 = 900$ comparisons per toolkit pair. Based on these comparisons, we compute the ARI and indicate the minimum ARIs, as well as the toolkit combinations[7] exhibiting that minimum. Referring again to the first row, collins, we see the minimum cross-toolkit agreement, $ARI = 0.50$ (attained by ALL toolkit combinations).

Finally, each table has three sets of rows: 25 "regular" rows on top where we show results for the top-25 *highest* accuracy for that algorithm. The *Median* and *Geometric Mean* are computed over the 25 datasets shown in the table. The last two rows, *Median (all)* and *Geometric Mean (all)*, are computed over all 162 datasets.

B. K-means with Varying Starting Points

The K-means algorithm requires "starting points", that is, initial cluster centers – with different starting points, the algorithm may converge to different minima. We explored the variation in outcome by randomly picking different starting points from the dataset to compare difference between toolkits.

Specifically, in each run we pick K (according the number of clusters in the dataset) points from the datasets and use them as initial centroids (cluster centers). We ran R in the default configuration (the 'R' grouped columns) as well as 100-iterations configuration ('R100iter' grouped columns) because by default R stops iterations early. Table II shows the results; we now proceed to discuss the results.

A bad choice of starting point can be *worse than random*. Note the fifth row, house-votes-84: all toolkits except R have a minimum value of *-0.02* (recall that $ARI = 0$ corresponds to random clustering); moreover, these toolkits' distributions are bimodal (marked with **B**) meaning the minimum is not an outlier (which would be marked as **L**).

[6]Assuming two clusterings $\overline{U}$ and $\overline{V}$ produced by two toolkits implementing the same algorithm and on the same input, equal accuracies, i.e., $ARI_{\overline{U}} = ARI_{\overline{V}}$, do not imply that $\overline{U}$ and $\overline{V}$ are equivalent up to a permutation. For example, even for K-means and Hierarchical where different toolkits have similar accuracy ranges, the toolkits still disagree on cluster composition.

[7]For space reasons, in the last column we use the abbreviations: 'MT' for MATLAB, 'T' for TensorFlow, 'W' for WEKA, 'S' for SKlearn, 'S0' for SKlearn0t, 'R1' for R100iter, 'SH' for Shogun, and 'MP' for MLpack.

TABLE III
K-MEANS++

Dataset	SKlearn			R			R100iter			MLpack			Matlab			Weka			Shogun			TensorFlow			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
collins	.56	.65	B	.55	.65	B	.57	.65	B	.54	.70	B	.65	.70	U	.27	.38	LR	.50	.70	LR	.49	.64	LR	.20	MP/W
confidence	.58	.58	U	.58	.65	B	.58	.61	U	.36	.71	B	.57	.69	B	.39	.71	LR	.35	.71	B	.36	.71	LR	.05	MP/T
corral	.13	.31	B	-.01	.31	B	.13	.31	B	-.01	.31	B	.13	.18	B	-.01	.37	B	-.01	.34	B	-.01	.38	B	-.09	MP/SH/W
ecoli	.46	.53	B	.46	.53	B	.47	.53	B	.42	.70	B	.68	.70	U	.44	.72	B	.42	.70	B	.44	.72	B	.43	MP/W
house-votes-84	.54	.54	U	.54	.54	U	.54	.54	U	.01	.54	B	.54	.54	U	-.02	.54	B	-.02	.54	B	-.02	.54	B	-.01	MP/SH/T/W
iris	.73	.73	U	.73	.73	U	.73	.73	U	.41	.73	B	.73	.73	U	.42	.71	B	.41	.73	B	.71	.73	U	.42	MP/SH/W
mfeat-karhunen	.56	.76	LR	.55	.70	LR	.56	.75	LR	.50	.70	LR	.55	.66	B	.45	.64	LR	.50	.75	LR	.51	.74	B	.35	MP/W
mfeat-pixel	.55	.69	B	.56	.69	B	.56	.75	LR	.48	.67	LR	.57	.61	U	.49	.69	LR	.49	.75	LR	.52	.72	B	.41	MP/W
monk3	.09	.09	U	.09	.09	U	.09	.09	U	-.01	.39	B	.07	.09	U	-.01	.23	B	-.01	.39	B	.01	.09	B	-.01	SH/T
mushroom	.11	.11	U	.11	.11	U	.11	.11	U	0	.11	B	.05	.11	B	0	.23	B	0	.11	B	0	.37	B	-.07	T/W
new-thyroid	.24	.62	B	.57	.59	U	.57	.59	U	.16	.60	B	.16	.59	B	.43	.62	B	.16	.59	B	.16	.62	B	.13	MP/MT/SH/T/W
optdigits	.66	.67	U	.67	.67	U	.67	.67	U	.52	.75	B	.59	.67	B	.50	.76	LR	.57	.75	LR	.57	.75	LR	.44	SH/W
promoters	1	1	U	1	1	U	1	1	U	1	1	U	1	1	U	-.01	.40	B	1	1	U	1	1	U	-.01	ALL
shuttle	0	0	U	.25	.27	U	.16	.32	LR	.14	.37	LR	0	.26	B	.18	.27	B	.24	.41	B	0	.45	B	-.01	MP/T
solar-flare_1	.26	.44	B	.25	.28	U	.25	.45	B	.07	.33	B	.22	.25	U	.03	.17	LR	.07	.45	LR	.06	.45	LR	-.01	SH/W
solar-flare_2	.33	.56	LR	.25	.57	LR	.12	.55	LR	.10	.48	LR	.15	.28	B	.07	.16	LR	.09	.55	LR	.13	.57	B	.04	SH/W
vote	.57	.58	U	.58	.58	U	.58	.58	U	0	.58	B	.57	.58	U	.57	.58	U	.57	.58	U	.57	.58	U	-.01	MP/S/SH/T/W
spambase	.03	.03	U	.03	.03	U	.03	.03	U	.03	.03	U	.03	.03	U	-.03	.35	B	.03	.03	U	0	.03	U	-.06	ALL
dermatology	.02	.02	U	.02	.02	U	.02	.02	U	.01	.06	U	.01	.06	U	.31	.91	B	.01	.08	B	.01	.06	B	-.02	MP/W
crx	0	0	U	0	0	U	0	0	U	0	0	U	0	0	U	0	.50	B	0	0	U	0	0	U	-.01	MP/MT/S/SH/T/W
credit-a	0	0	U	0	0	U	0	0	U	0	0	U	0	0	U	-.01	.50	B	0	0	U	0	0	U	-.01	MP/S/SH/T/W
buggyCrx	0	0	U	0	0	U	0	0	U	0	0	U	0	0	U	0	.50	B	0	0	U	0	0	U	-.01	MP/MT/S/SH/T/W
australian	0	0	U	0	0	U	0	0	U	0	0	U	0	0	U	-.01	.50	B	0	0	U	0	0	U	-.01	MP/MT/SH/T/W
appendicitis	.31	.33	U	.31	.31	U	.31	.31	U	-.06	.37	B	.29	.29	U	-.06	.37	B	.29	.37	B	.29	.37	B	.07	MP/SH/T/W
Median	*.28*	*.48*		*.28*	*.42*		*.28*	*.49*		*.05*	*.44*		*.19*	*.28*		*.05*	*.50*		*.12*	*.49*		*.10*	*.50*		*-.01*	
Geometric Mean	*.29*	*.35*		*.31*	*.36*		*.31*	*.37*		*.17*	*.39*		*.28*	*.34*		*.16*	*.48*		*.21*	*.41*		*.21*	*.41*		*.08*	
Median (all)	*.02*	*.05*		*.02*	*.05*		*.03*	*.05*		*.01*	*.06*		*.02*	*.05*		*0*	*.15*		*.01*	*.07*		*.01*	*.07*		*-.01*	
Geometric Mean (all)	*.12*	*.14*		*.12*	*.14*		*.12*	*.14*		*.08*	*.15*		*.11*	*.14*		*.08*	*.20*		*.09*	*.16*		*.09*	*.16*		*.08*	

MAX performance (best case). No toolkit outperforms consistently. For example, on dataset flags, *all toolkits except* R100iter have max accuracy of 0.11–0.15. However, R100iter achieves three times better accuracy: .35! For haberman, with the exception of MLpack, all toolkits' max hovers around 0; hence, except MLpack, all toolkits' top performance is close to random.

MIN performance (worst case). house-votes-84 shows the danger of local minima: all toolkits, except R, have mins of -0.02 (worse than random). Occasionally, these toolkits will achieve high accuracy. However, R users are much better "protected": their min is essentially the same as their max (.52 min, .54 max).

Instability, as revealed by distribution shapes. Recall that **U** indicates "predictable" performance. However, the table shows the abundance of bimodality and outlier-prone outcomes, i.e., **B**, **L**, **R**, **LR**. The last row shows the median values for min- and max- accuracy, respectively. Notice how accuracy can vary from .25–.29 (min) to .52–.53 (max), indicating a large degree of instability.

C. K-means++

Table III shows the clustering outcomes when running K-means with starting points generated according the K-means++ initialization algorithm. However, for K-means++ we do not control how the starting points are chosen, as K-means++ is supposed to improve upon K-means with a better initialization. We now discuss the findings.

No real improvement compared to random starting points. Despite the fact that K-means++ was devised to improve upon K-means, in our experiments K-means++ does not achieve higher accuracy compared with the random starting points (Section V-B). Indeed, in the last rows, showing the median values for min- and max- accuracy, respectively, we observe that we are around the same values: .22–.31 to .29–.54. However, there is an improvement in terms of stability – comparing the shape labels in Table II and Table III we see more stability (more **U**'s).

TABLE IV
HIERARCHICAL

Dataset	SKlearn			R			Matlab			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
Hill_Valley_with_noise	0	0	U	0	0	U	0	0	U	1	ALL
analcatdata_germangss	.01	.01	U	.01	.01	U	.01	.01	U	1	ALL
balance-scale	.16	.16	U	.17	.17	U	.12	.12	U	.29	MT/S
vote	.49	.49	U	.49	.49	U	.49	.49	U	1	ALL
breast-cancer-wisconsin	.28	.28	U	.28	.28	U	.28	.28	U	1	ALL
analcatdata_aids	-.02	-.02	U	-.02	-.02	U	-.02	-.02	U	1	ALL
cmc	.01	.01	U	.03	.03	U	.01	.01	U	.46	R/S
analcatdata_happiness	.10	.10	U	.10	.10	U	.10	.10	U	1	ALL
backache	-.01	-.01	U	-.01	-.01	U	-.01	-.01	U	1	ALL
buggyCrx	0	0	U	0	0	U	0	0	U	1	ALL
analcatdata_japansolvent	0	0	U	0	0	U	0	0	U	1	ALL
mfeat-karhunen	.57	.57	U	.57	.57	U	.57	.57	U	1	ALL
ionosphere	.18	.18	U	.18	.18	U	.18	.18	U	1	ALL
car	0	0	U	0	0	U	-.01	-.01	U	.17	MT/R
solar-flare_1	.25	.25	U	.25	.25	U	.24	.24	U	.61	R/S
pima	.10	.10	U	.10	.10	U	.10	.10	U	1	ALL
house-votes-84	.59	.59	U	.67	.67	U	.33	.33	U	.36	MT/S
allhypo	.02	.02	U	.02	.02	U	.02	.02	U	1	ALL
tic-tac-toe	0	0	U	0	0	U	-.02	-.02	U	-.08	MT/R
pendigits	.55	.55	U	.55	.55	U	.55	.55	U	.99	ALL
waveform-21	.31	.31	U	.31	.31	U	.31	.31	U	1	ALL
analcatdata_asbestos	.11	.11	U	.11	.11	U	.11	.11	U	1	ALL
flags	.02	.02	U	.02	.02	U	.03	.03	U	.97	ALL
soybean	.40	.40	U	.38	.38	U	.38	.38	U	.82	MT/S
analcatdata_authorship	.76	.76	U	.77	.77	U	.77	.77	U	.94	ALL
Median	*.10*	*.10*		*.10*	*.10*		*.10*	*.10*		*1*	
Geometric Mean	*.20*	*.20*		*.20*	*.20*		*.19*	*.19*		*.79*	
Median (all)	*.04*	*.04*		*.03*	*.03*		*.02*	*.02*		*1*	
Geometric Mean (all)	*.13*	*.13*		*.13*	*.13*		*.12*	*.12*		*.72*	

Weka differs from the other toolkits. When using K-means++, we were able to add Weka to our study (Weka does not permit specifying starting points hence its absence from Section V-B). Weka has interesting behavior, markedly different from the other toolkits. For example, if we look at credit-a through australian datasets (lower half of the table) we can see that no algorithm can break 0 (they achieve 0 min/max

TABLE V
EM/GAUSSIAN

Dataset	SKlearn			SKlearn0T			Matlab			Weka			TensorFlow			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
prnn_crabs	0	0	U	.02	.02	U	-.01	.97	B	-.01	-.01	U	-.01	1	B	-.02	MT/T
analcatdata_creditscore	-.04	.26	B	-.05	.26	B	-.03	.95	B	0	0	U	-.03	.25	B	-.08	S/S0/T
twonorm	.90	.90	U	.90	.90	U	0	.91	B	.91	.91	U	-.01	.90	B	-.05	MT/T
analcatdata_authorship	.59	.90	LR	.50	.90	LR	.04	.79	LR	.95	.95	U	-.01	.41	B	-.09	MT/T
wdbc	.81	.81	U	.81	.81	U	0	.75	B	.67	.67	U	.03	.76	B	-.01	MT/T
breast-cancer-wisconsin	.81	.81	U	.81	.81	U	0	.72	B	.67	.67	U	.03	.76	B	-.01	MT/T
ionosphere	.39	.40	U	.39	.40	U	-.02	.43	B	.25	.25	U	0	.77	B	-.04	MT/T
wine-recognition	.45	.60	B	.44	.61	B	.32	.94	B	.91	.91	U	.02	.49	B	-.06	MT/T
breast	-.01	.70	B	-.01	.70	B	-.01	.58	B	.76	.76	U	.48	.67	B	-.01	MT/S/S0/W
dermatology	.08	.35	B	.02	.36	B	.41	.84	B	.52	.82	B	.11	.65	LR	-.06	MT/T
new-thyroid	.86	.86	U	.86	.86	U	.41	.90	B	.89	.89	U	.40	.86	LR	.24	MT/T
vote	.47	.54	B	.47	.54	B	0	.62	B	.47	.57	B	-.03	.45	B	-.03	MT/T
iris	.90	.90	U	.90	.90	U	.56	.56	U	.75	.75	U	.51	.90	B	.50	T/W
house-votes-84	-.02	.49	B	-.02	.49	B	-.03	.56	B	.55	.55	U	-.02	.57	B	-.05	MT/T
biomed	.18	.54	B	.18	.57	B	.01	.55	B	.54	.54	U	0	.57	B	-.02	MT/T
dna	.26	.50	LR	.33	.49	B	-.02	.10	LR	.30	.72	B	-.03	.01	U	-.03	T/W
waveform-40	.25	.25	U	.53	.53	U	.25	.56	B	.25	.25	U	0	.53	B	-.03	MT/T
ecoli	.27	.61	B	.25	.61	B	0	0	U	.34	.73	B	.53	.75	B	0	ALL
confidence	.32	.60	B	.32	.67	B	.30	.66	LR	.57	.75	B	.35	.62	B	.27	S0/T
promoters	1	1	U	1	1	U	-.01	.25	B	.45	.62	B	-.01	.03	U	-.05	MT/T
optdigits	.41	.57	B	.36	.61	LR	.45	.66	LR	.22	.61	B	.29	.53	LR	.12	T/W
waveform-21	.25	.25	U	.57	.57	U	.15	.58	B	.25	.25	U	.15	.57	B	.06	MT/T
splice	.02	.35	B	.02	.36	B	-.02	.26	B	.23	.34	LR	-.03	.49	B	-.07	MT/T
mushroom	0	.12	B	0	.38	B	-.01	.45	B	.07	.07	U	-.01	.49	B	-.10	MT/T
shuttle	.04	.23	B	.03	.20	B	.03	.50	B	.19	.32	B	.08	.27	B	.01	S/S0
Median	*.29*	*.55*		*.37*	*.59*		*0*	*.58*		*.46*	*.62*		*0*	*.57*		*-.03*	
Geometric Mean	*.35*	*.54*		*.37*	*.58*		*.09*	*.57*		*.44*	*.53*		*.09*	*.53*		*.01*	
Median (all)	*.01*	*.10*		*.01*	*.09*		*0*	*.16*		*.03*	*.10*		*-.01*	*.16*		*-.02*	
Geometric Mean (all)	*.10*	*.18*		*.11*	*.18*		*.04*	*.21*		*.13*	*.18*		*.03*	*.22*		*.01*	

with a uniform distribution) whereas Weka has a bimodal distribution with accuracies of up to 0.5. Unfortunately, for the "easy" promoters set where all algorithms achieve a 1 score, Weka can only manage between 0.1 and 0.41.

Similarly, the agreement columns show that Weka has, in some cases, minimum agreement with other toolkits (e.g., solar-flare_1, but it is never present when agreeing with the maximum values. Even worse, on a large number of cases (not shown due to space limits), all toolkits report a 100% agreement with each other except with Weka!

We reached out to WEKA developers who suggested that we change WEKA's default configuration (turn normalization off) to improve its performance on this particular dataset [18]. While turning off normalization improved the performance on this dataset. we believe it is important for uniformity to run all toolkits with default parameters, as per-dataset tweaking might affect behavior negatively for other datasets.

MAX performance (best case). Similarly to K-means with random starting points, for K-means++ no toolkit outperforms consistently. For example, on monk3, Shogun and MLPack have a maximum accuracy of 0.39 where other algorithms do not get higher than 0.09 (with the exception of Weka which have a maximum score of 0.23), that is four times lower!

MIN performance (worst case). The minimum shows that even if considering using a specific algorithm for drawing our starting points, the difference min/max can be important. MLPack seems to be really sensitive as its min/max can range greatly. A clear examples is solare_flare_2 with a minimum of 0.1 where the maximum was 0.49; similarly for vote where minimum is 0.01 and maximum is 0.59.

D. Hierarchical/Agglomerative

Table IV shows the results obtained with the hierarchical (agglomerative) clustering.

Deterministic runs. Unlike the previous algorithms, hierarchical is deterministic, hence we expect no variation between runs. Indeed we find that for a given toolkit, distribution is uniform (all **U**'s).

Difference across toolkits. What is concerning however, is the difference between toolkits, e.g., on sets house-votes-84 (max is .59 for SKLearn, .67 for R, .33 for Matlab) or balance-scale (max's were .16, .17, and .12 respectively). For a deterministic algorithm there should be no such variation.

Toolkit (dis)agreement. Excepting some specific cases, all toolkits agree on their outcome as we have a large number of 1 as minimum values. A few datasets however show some disagreement, e.g., car and solar-flare_1. However, for a deterministic algorithm, there should be no such disagreement.

We emphasize that finding the root causes of these determinism violation is orthogonal to this paper, and a direction we leave to future work.

E. EM/Gaussian

Table V shows the results on Gaussian mixture.

MAX performance (best case). This algorithm stands out in that multiple toolkits achieve max performance of 0.9 or higher (Matlab, for instance, does so on four datasets, while SKlearn and Weka do so on three!).

MIN performance (worst case). house-vote-84 poses difficulties for all toolkits (minimum is around -0.03 to -0.01) except Weka, which achieves a minimum of 0.56!

The tolerance parameter from SKlearn has limited impact on results. Only for waveform-40, SKlearn with default tolerance attains a 0.25 max, whereas SKlearn0t attains double the accuracy (0.53).

Overall, best performance. EM/Gaussian is the only algorithm where multiple toolkits (Matlab and TensorFlow) exceed a 0.2 geometric mean across the 162 datasets.

TABLE VI
SPECTRAL CLUSTERING

Dataset	SKlearn			SKlearnFast			R			T.A.			
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*	*Max*	*Algo*
breast-w	.10	.10	U	.80	.80	U	.05	.81	LR	-.03	R/S	.93	R/SF
mfeat-pixel	0	0	U	.92	.92	U	.55	.92	LR	-.01	R/S	.96	R/SF
dermatology	.01	.01	U	.17	.17	U	.47	.86	LR	-.04	R/S	.18	R/SF
breast-cancer-wisconsin	-.01	0	U	.41	.41	U	0	.53	B	-.01	R/S	.18	R/SF
wdbc	-.01	.32	B	.41	.41	U	.02	.53	B	-.01	R/S	.43	R/S
analcatdata_lawsuit	-.07	0	B	.03	.03	U	.31	.69	B	-.08	R/S	.05	R/SF
analcatdata_creditscore	-.05	.26	B	.84	.84	U	-.03	-.01	U	-.07	R/S	.19	S/SF
confidence	.01	.01	U	.70	.70	U	.32	.68	B	-.14	R/S	.86	R/SF
appendicitis	.35	.35	U	.46	.46	U	-.04	.45	B	-.09	R/SF	.76	R/SF
corral	.48	.48	U	.13	.13	U	-.01	.38	B	-.02	R/S	.55	R/SF
collins	-.01	0	U	.61	.63	U	.40	.66	LR	-.02	R/S	.60	R/SF
new-thyroid	-.12	-.08	U	.27	.27	U	.23	.50	B	-.12	R/S	.23	R/SF
mfeat-fourier	.54	.56	U	.56	.56	U	.41	.62	LR	.47	R/S	.66	R/S
mfeat-factors	-.01	0	U	.67	.67	U	.47	.66	B	-.01	R/S	.83	R/SF
mfeat-zernike	0	0	U	.56	.57	U	.47	.66	B	-.04	R/S	.80	R/SF
mfeat-morphological	0	.01	U	.23	.30	B	.17	.45	B	-.03	R/S	.37	R/SF
solar-flare_2	.08	.10	U	-.02	.04	B	.11	.41	B	-.10	R/SF	.61	R/SF
analcatdata_bankruptcy	-.02	.18	B	.45	.45	U	.04	.30	B	-.12	R/S	.43	R/S
iris	.74	.74	U	.75	.75	U	.55	.70	B	.58	R/S	.94	S/SF
ecoli	.60	.62	U	.50	.50	U	.58	.73	LR	.47	R/SF	.70	R/SF
balance-scale	-.01	.31	LR	.13	.13	U	0	.21	B	0	R/S	.80	S/SF
soybean	.01	.03	U	.19	.29	B	.25	.46	B	.01	R/S	.58	R/SF
threeOf9	-.01	.29	B	-.01	.12	B	-.01	.09	B	-.01	ALL	.51	S/SF
analcatdata_authorship	-.01	.01	U	.96	.96	U	.63	.72	B	-.01	S/SF	.75	R/SF
lupus	-.02	0	U	.19	.21	U	-.02	.25	B	-.04	R/S	1	R/SF
Median	*0*	*.03*		*.45*	*.45*		*.23*	*.53*		*-.02*		*.61*	
Geometric Mean	*.08*	*.15*		*.41*	*.43*		*.22*	*.51*		*.01*		*.57*	
Median (all)	*-.01*	*0*		*.03*	*.03*		*0*	*.04*		*-.01*		*.57*	
Geometric Mean (all)	*.03*	*.05*		*.12*	*.12*		*.07*	*.15*		*.02*		*.52*	

F. Spectral

Table VI shows the performance for SKlearn (Gaussian), SKlearnFast (k-nearest) and R (Gaussian).

SKlearnFast is a solid all-around choice. SKlearnFast outperforms other implementations, e.g., analcatdata_authorship reports a performance of .96 where it is around .63 to .72 for R and 0 for SKlearn! Similar for mfeat-pixel. More than having a high global performance, it is also quite stable (a lot of **U**'s) despite the fact that it is supposed to sacrifice accuracy in the name of efficiency (compared to Gaussian).

MAX performance (best case). The max values range from 0 to .92 depending on the dataset and the toolkit. However, SKlearn shows the worst performance as it is the only one to perform consistently worse than random (9 times, its max is around 0; across all datasets it has a min/max of 0, too), whereas the max for SKlearnFast and R are much higher.

SKlearnFast agrees more with R than with SKlearn. Last columns show that R generally agrees with SKlearnFast regarding maximums (18 times), where it generally agrees with SKlearn for the minimums (20 times).

G. DBSCAN

Recall that DBSCAN is deterministic; Table VII shows the results.

Low performance (with default parameters). We noticed that DBSCAN can suffer from low accuracy with default parameters. For example, on datasets new-thyroid and analcatdata_lawsuit, accuracy can be as low as -0.2 and -0.1, respectively: *lower than random and lower than K-means++*. To gauge the impact of (small) variations in defaults, we also ran experiments where we varied its $minPoints$ and ε parameters.[8] This leads to wide-spread bimodality and outliers – note the **B**'s and **LR**'s. For example, the accuracy varies widely for the same toolkit across different runs: this range can be as large as 0.95 (min 0, max 0.95) for analcatdata_creditscore, 0.89 (min -0.2, max 0.69) for new-thyroid or 0.84 for breast-w (min -0.07, max 0.77). This finding is especially worrisome for a deterministic algorithm.

[8]These control the minimum cluster size (our range was $1 \leq minPoints \leq 10$) and maximum neighborhood size (our range was $0 < \varepsilon < 10$.

TABLE VII
DBSCAN

Dataset	SKlearn			R			MLPack			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
analcatdata_creditscore	0	.95	B	0	.95	B	0	.95	B	-.06	ALL
breast-w	-.07	.77	B	-.07	.77	B	-.07	.77	B	-.34	ALL
new-thyroid	-.20	.69	B	-.20	.69	B	-.21	.69	B	-.26	ALL
ionosphere	0	.65	B	0	.65	B	0	.66	B	-.12	ALL
collins	0	.63	B	0	.63	B	0	.63	B	-.08	ALL
iris	0	.56	B	0	.56	B	0	.56	B	0	ALL
vote	-.01	.47	B	-.01	.47	B	-.02	.45	B	-.12	ALL
spect	-.08	.32	B	-.08	.32	B	-.10	.32	B	-.17	ALL
analcatdata_lawsuit	-.10	.29	B	-.10	.29	B	-.10	.29	B	-.24	ALL
led7	0	.32	B	0	.32	B	0	.32	B	0	ALL
house-votes-84	-.03	.30	B	-.03	.30	B	-.02	.31	B	-.16	ALL
soybean	-.02	.27	B	-.02	.27	B	-.02	.30	B	0	ALL
mfeat-fourier	0	.29	B	0	.29	B	0	.29	B	0	ALL
titanic	0	.27	B	0	.27	B	0	.27	B	0	ALL
spectf	-.02	.26	B	-.02	.26	B	-.02	.26	B	0	ALL
prnn_fglass	0	.26	B	0	.26	B	0	.26	B	0	ALL
glass	0	.26	B	0	.26	B	0	.26	B	0	ALL
dermatology	0	.21	B	0	.21	B	0	.21	B	-.15	ALL
dna	-.06	.18	B	-.06	.18	B	-.06	.18	B	-.12	ALL
lymphography	0	.21	B	0	.21	B	-.04	.17	B	-.17	ALL
page-blocks	-.01	.19	LR	-.01	.19	LR	-.01	.20	LR	-.05	ALL
haberman	-.08	.16	LR	-.08	.16	LR	-.06	.16	B	-.22	ALL
agaricus-lepiota	-.01	.19	B	-.01	.19	B	-.01	.19	B	-.02	ALL
clean2	-.09	.15	LR	-.09	.15	LR	-.09	.15	LR	-.01	ALL
tic-tac-toe	0	.17	B	0	.17	B	0	.17	B	0	ALL
Median	*-.01*	*.27*		*-.01*	*.27*		*-.01*	*.29*		*-.06*	
Geometric Mean	*-.03*	*.35*		*-.03*	*.35*		*-.04*	*.35*		*-.10*	
Median (all)	*-.01*	*.01*		*-.01*	*.01*		*-.01*	*.01*		*-.01*	
Geometric Mean (all)	*-.02*	*.06*		*-.02*	*.06*		*-.02*	*.07*		*-.05*	

TABLE VIII
AFFINITY PROPAGATION

Dataset	SKlearn			R			T.A.	
	Min	*Max*	*Shape*	*Min*	*Max*	*Shape*	*Min*	*Algo*
collins	.19	.63	B	.21	.62	B	.11	ALL
breast-w	.03	.47	B	.07	.17	B	.10	ALL
iris	.42	.67	B	.44	.64	B	.47	ALL
cleveland-nominal	-.02	.31	B	.02	.03	U	.11	ALL
tokyo1	-.01	.31	L	.10	.30	B	0	ALL
promoters	0	.28	B	.23	.28	U	0	ALL
mfeat-morphological	0	.28	B	0	.27	B	-.01	ALL
ecoli	.21	.37	B	.21	.24	U	.15	ALL
titanic	-.01	.24	B	-.01	.09	B	0	ALL
soybean	.21	.34	LR	.21	.25	U	.56	ALL
wine-recognition	.17	.30	B	.17	.21	U	.45	ALL
analcatdata_cyyoung9302	0	.19	B	.18	.19	U	0	ALL
analcatdata_creditscore	-.03	.16	LR	-.03	.16	LR	.01	ALL
dermatology	0	.15	B	.14	.15	U	.05	ALL
confidence	.24	.31	B	.24	.31	B	.50	ALL
new-thyroid	.04	.17	B	.12	.17	B	0	ALL
segmentation	0	.14	B	.12	.14	U	0	ALL
mfeat-fourier	0	.13	B	.12	.13	U	0	ALL
balance-scale	0	.06	LR	.03	.15	B	0	ALL
analcatdata_bankruptcy	.04	.15	B	.04	.15	B	.07	ALL
solar-flare_1	.08	.17	B	.09	.15	B	.13	ALL
prnn_synth	.07	.17	B	.07	.11	U	.38	ALL
solar-flare_2	.01	.13	B	.02	.12	B	0	ALL
prnn_fglass	.13	.20	B	.13	.17	U	.10	ALL
glass	.13	.20	B	.13	.17	U	.10	ALL
Median	*.02*	*.22*		*.12*	*.17*		*.07*	
Geometric Mean	*.07*	*.26*		*.12*	*.21*		*.12*	
Median (all)	*0*	*.04*		*.01*	*.03*		*.05*	
Geometric Mean (all)	*.02*	*.08*		*.04*	*.07*		*.18*	

H. Affinity Propagation

Table VIII shows the results. Recall that this algorithm (AP) is deterministic. AP uses a $dampingFactor$ parameter.[9]

Variation across toolkits. Given that this algorithm is deterministic, we should not see variation across toolkits when toolkits are run with the same parameter (damping factor). However, max performance differed substantially, e.g., on breast-w max was .47 for SKLearn and .17 for R; for cleveland-nominal max was .31 for SKLearn and .03 for R.

Variation across runs. Our experiments show that the damping factor induces substantial differences between min and max performances across runs. This was the case for both SKLearn and R, e.g., for SKLearn we had collins (min .19, max .63) or breast-w (min .03, max .47) and for balance-scale on R we had (min 0.03, max .15).

VI. Related Work

While clustering is a richly explored field, prior clustering research efforts have not questioned or investigated clustering reliability or correctness. For example, Software Engineering research has used clustering as a tool rather than as an object of study; Machine Learning and Data Mining research can be split into theoretical research on clustering properties, or experiments on improving clustering; in both cases, the research literature assumes correct algorithm implementations.

The study closest to us in breadth of algorithm/toolkit combinations is Kriegel et al.'s [19]. They have also pointed out the peril of assuming that "toolkits don't matter": an algorithm's implementation is *not* standardized across all toolkits. They have compared several algorithm and implementations on a narrower benchmark set (a single dataset of 500k Twitter locations, and subsets thereof) but their goal was different: runtime efficiency. They found orders-of-magnitude differences across toolkits for the same algorithm and same input dataset.

Ben-Hur et al. [20] have investigated hierarchical clustering on several datasets: varying K to find the value for which the algorithm is most stable. Our goal is toolkit dependability, and our focus is on datasets with ground truth and fixed K.

Fred [21] has proposed voting K-means, an improvement upon standard K-means by choosing clusters on a majority voting policy, to weed out outliers. They use consistency (similarity of partitionings for multiple runs of K-means on the same dataset and the same K) which is akin to our notion of determinism. Their experiments were run with varying K on two datasets. Our use of ARI is more robust, and our goal is toolkit dependability, rather than improving K-means.

VII. Conclusions

Clustering is widely used, but its reliability has not been questioned yet. Moreover, verification and validation approaches for clustering are scarce. We propose SmokeOut, an approach for testing clustering implementations that leverages the current abundance of datasets and clustering toolkits. We applied SmokeOut to quantify clustering outcomes across different algorithms, toolkits, and multiple runs. Our findings show large variations across all these dimensions, including violations of determinism. Our approach has the potential to improve the state-of-the-art in clustering by allowing software engineering practitioners and researchers to test and improve clustering implementations, which in turn benefits the larger classes of clustering users.

[9]The factor controls oscillations; in our experiments $0.5 < dampingFactor < 1$.

Acknowledgments

This material is based upon work supported by the NSF Grant No. CCF-1629186. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] H. P. Ng, S. H. Ong, K. W. C. Foong, P. S. Goh, and W. L. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, March 2006, pp. 61–65.

[2] W. Vogt and D. Nagel, "Cluster analysis in diagnosis." *Clinical Chemistry*, vol. 38, no. 2, pp. 182–198, 1992.

[3] P. G. Sun, L. Gao, and S. Han, "Prediction of human disease-related gene clusters by clustering analysis," *International journal of biological sciences*, vol. 7, no. 1, pp. 61–73, 01 2011.

[4] W. Wang, A. Ramesh, and D. Zhao, "Clustering of driving scenarios using connected vehicle datasets," *CoRR*, vol. abs/1807.08415, 2018.

[5] T. Brennan and W. L. Oliver, "The emergence of machine learning techniques in criminology," *Crim. & Public Policy*, vol. 12, 08 2013.

[6] F. Cai, N.-A. Le-Khac, and T. Kechadi, "Clustering approaches for financial data analysis: a survey," 09 2016.

[7] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[8] H. A. Güvenir, G. Demiröz, and N. Ilter, "Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals," *Artificial Intelligence in Medicine*, vol. 13, p. 147, 1998.

[9] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "Pmlb: a large benchmark suite for machine learning evaluation and comparison," *BioData Mining*, vol. 10, no. 1, p. 36, Dec 2017.

[10] L. Hubert and P. Arabie, "Comparing partitions," vol. 2, pp. 193–218, 02 1985.

[11] D. Steinley, "Properties of the hubert-arable adjusted rand index." *Psychological methods*, vol. 9, no. 3, p. 386, 2004.

[12] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behavioral Research*, vol. 21, no. 4, pp. 441–458, 1986.

[13] "Mathworks fast facts," https://www.mathworks.com/company/aboutus.html.

[14] M. Hornick, "Oracle r technologies overview," https://www.oracle.com/assets/media/oraclertechnologies-2188877.pdf.

[15] "Tensorflow github," https://github.com/tensorflow/tensorflow.

[16] "Companies using tensorflow," https://www.tensorflow.org/.

[17] "Who is using scikit-learn?" http://scikit-learn.org/stable/testimonials/testimonials.html.

[18] "Weka mailing list," September 2018, http://weka.8497.n7.nabble.com/Weka-clustering-diverges-from-other-toolkits-td43955.html.

[19] H.-P. Kriegel, E. Schubert, and A. Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?" *Knowl. Inf. Syst.*, vol. 52, no. 2, pp. 341–378, Aug. 2017.

[20] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Proceedings of the 7th Pacific Symposium on Biocomputing*, 2002, pp. 6–17.

[21] A. Fred, "Finding consistent clusters in data partitions," in *In Proc. 3d Int. Workshop on Multiple Classifier*. Springer, 2001, pp. 309–318.