Pattern Matching in the Era of Big Data: A Benchmark of Cluster Quality Metrics

Ole Kristian Ekseth, Per Jarle Furnes, and Svein-Olaf Hvasshovd
Department of Computer Science (IDI)
NTNU & Eltorque
Trondheim, Norway
email: oekseth@gmail.com and sophus@ntnu.no

***Abstract*—In today's quest for knowledge, there is a need for accurate and fast measures for pattern matching. While numerous new metrics and algorithms are published every year, researchers are unaware of which metric to choose. There does not exist an established strategy for pattern matching of cluster algorithms, which may explain why new hypothesis and algorithms are often forgotten. In this work, we address this issue. The paper presents a new benchmark for automated evaluation of pattern matching algorithms. From key characteristics of training data, the benchmark deduce fast and accurate cluster quality metrics, hence enabling pattern searches in big data. The benchmark address key issues in pattern analysis: while recent algorithms improve prediction accuracy by less than 2x, there is a 5x+ inaccuracy in established pattern matching algorithms. The evaluation of 100+ real-life data-sets reveals how the benchmark manages to identify patterns which are otherwise hidden, hence paving the ground for improved quality in the field of big-data pattern matching.**

***Keywords: patterns; clustering; similarity metrics; data analysis*.**

I. Introduction

Data mining and big data analysis have experienced a surge of interest in the recent years [1]. In data analysis, it is essential to know the trustworthiness of other's findings. An example is seen in the work of [2], where the authors report the patterns to have a difference of 1.67x: is the improvement reported by [2] sufficient to discard the earlier ground-truth?

The motivation of many research papers is to demonstrate that some algorithms are superior to other [3]–[5]. In contrast, our hypothesis is that the choice of clustering algorithms depends on both the data and the local configurations of the clustering algorithms (code-listing 1): the established strategy results in a 100x+ prediction error.

Of importance is to identify the reasons for why measurement data, presented in different research papers, diverge. The diverging recommendations, which seems to be the rule in benchmarking of algorithms, raises several questions. To exemplify, is the rarely discussed choice of *benchmark data* the determining factor?

While the choice of validity metrics (*e.g.*, Silhouette) determines the outcome of experiments (*best column* in Table II), the established strategies have a poor trustworthiness in seperating between *false* versus *true* hypothesis (Section VI). For some data-sets the VRC metric [6] is unable to spot differences in data predictions. Hence, perturbations in data will remain unknown to VRC users. The issue of VRC inaccuracy arises from how metrics weight differences in data (sub-section V-A).

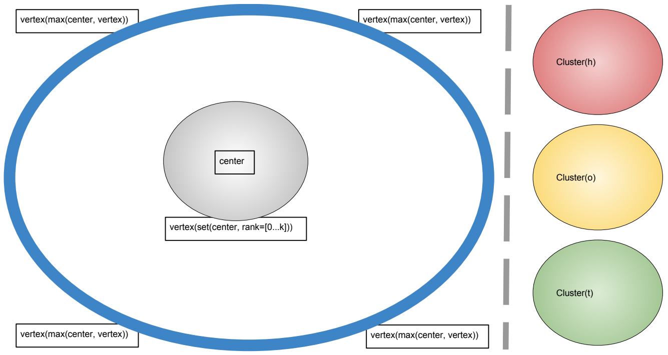
vertex(max(center, vertex))
vertex(max(center, vertex))
center
vertex(set(center, rank=[0...k]))
vertex(max(center, vertex))
vertex(max(center, vertex))
Cluster(h)
Cluster(o)
Cluster(t)

Fig. 1. Ambiguities in cluster analysis. Above figure demonstrates why it is impossible to use a a single number (*e.g.*, '0.8') to describe pattern similarities between two cluster partitions, *e.g.*, when comparing a null-hypothesis to the results of a clustering algorithm.

To address this challenge, this paper constructs an automated method for capturing the bias in metrics and training data. The method explores the parameter space of *pattern matching algorithms* (Fig. 1). The measurements reveals how the approach addresses issues in [2]–[5].

The results demonstrate how the proposed framework increases the accuracy of data classification by 5x+ (sub-section VI-C). The benchmark captures peculiarities in turf specific data-sets. This knowledge is important in a number of domains:

1) trustworthiness: the significance of differences when no hypothesis is valid;
2) experimental design: the accurate and efficient exploration of a hypothesis in large data volumes;
3) new metrics: the automated identification of new metrics from training data (code-listing 1).

The remainder of the paper is organized as follows. Section II identifies the contributions of this paper. Section III briefly surveys related approaches, and Section IV describes a new algorithm for automated evaluation and identification of metrics. Section V describes an approach to evaluate the influence of strategies for pattern matching, a method which is applied in the result Section VI. This paper ends with a brief summary

TABLE I
APPLICABILITY OF ALGORITHMS FOR PATTERN MATCHING. THE TABLE CAPTURES THE ACCURACY AND RESOLUTION OF PATTERN MATCHING ALGORITHMS.

Year	Name	Cite	Gold	Diff. / Equal	Equal: Worst / best	Diff.: Worst / best
1971	Rand's Index		x	100x	-	-
1974	VRC		-	100x	-	
1974	Dunn		-			
1974	Dunn		x			
1979	Davis-Bouldin		x	∞	1x	1x
1979	Chi-squared		x	∞	4.5x	
1982	SSE		x	1x		
1982	SSE		-	79x		
1983	FM		x	20x		
1987	Silhouette		x		-	
1987	Silhouette			∞	-	
2001	R-squared		-	2x		
2001	ARI		x	∞		
2001	Mirkin		-	∞		
2003	Fred & Jain		x	25x	5x	
2003	Strehl & Gosh		x	25x	5x	
2007	Wallace		-	1x		
2007	VOI		-	25x	10x	
2015	MMM		x	1x		
2017	Dogen		x	∞		
2010	RMSSTD		-	78x		

of observations in Section VII.

II. CONTRIBUTIONS

This paper presents a benchmark identifying the drawbacks of clustering metrics. The work manages to both quantity the trustworthiness–threshold for algorithms, and provide a software for automated benchmarking of users own data. The results reveal how the proposed benchmark provides users with software which identifies fast and accurate cluster algorithms (Fig. 2). Hence, a new approach which deduces fast and accurate cluster quality metrics.

Today, it is impossible to check if results (*e.g.*, produced by new algorithms) makes sense. This due to configuration parameters not described in research papers, *i.e.*, hidden factors which are left unexplained to the users. Of importance is therefore to identify benchmarks, and a software tool, for capturing the algorithms which produce the best cluster predictions at lowest possible execution time.

To address this issue, this paper provides users with a tool for big-data analytics. The tool covers a big assortment of metrics and databases. The results are presented through generalization of algorithms and metrics, hence easing the accessibility (of the findings) across a wide spectrum of research domains.

The proposed method may be applied to existing algorithms and clustering (code-listing 1). Hence, the work avoids the pitfall of proposing new algorithms (instead of improving the existing). The work analysis the patterns used in clustering algorithms, and the clustering evaluation, for which a *Pareto Boundary* is identified. The *Pareto Boundary* provides the means to identify when a given algorithm provides improves beyond the noise threshold (Fig. 2).

To summarize, the paper presents a new algorithm and software. The the software enables users to select metric combinations with high prediction accuracy at low execution time, hence paving the ground for improved quality in the field of big-data pattern matching.

III. RELATED WORK

Application of data analysis requires accurate strategies to capture the similarities across measurements, hypothesis, data perturbations, etc. There are more than 30+ metrics for capturing the patterns of data, for which a subset is listed in Table I. Below section demonstrates how researchers are unaware of the ambiguity in “Cluster Comparison Metrics (CCMs)”. While accurate algorithm configuration increases prediction quality by 10x+ (Fig. 2), new algorithms provides less than a 2x prediction improvement. Prediction improvements are measured through algorithms such as “Rand’s Index”

Accuracy of data analysis is fundamental in all parts of research, such as bio-medicine [1], language processing [7], image recognition and reconstruction [5], [8], etc. An application of data analysis is to establish the significance of new findings: to identify the degree of correlation between hypothesis and experimental outcomes. Examples of widely used metrics are “Sum of Squared Error (SSE)”, Silhouette, “Rand’s Index”, etc.

Pattern matching in big-data represents the performance crux in drug discovery [1], epidemiology [9], etc. Clustering is able to group mixed data into groups, called clusters, focusing on the similarity between the data points [10]. The requirements for *big data* differ from other application and domains. The work of [11] observes how “big data analytics requires technologies to efficiently process large quantities of data” [11]. Software for pattern matching suffers from high execution time, as observed for “Sci-kit learn” [12] and the “Moa” software [13].

Partitioning of data into clusters involves assumptions of the data: to use metrics for similarities between groups to partition data. For example, the default “k-means” implementation uses Euclidean distance to cluster numerical data points [14], “k-modes” groups categorical data [15], while “k-prototypes” uses cost functions to group mixed data [16].

A challenge concerns how to evaluate and interpret the identified patterns (Fig. 1). The ambiguity of CCMs is due to its purpose: from variance and agreements inside each cluster, and between multiple clusters, to infer a representative number (to capture the fit between hypothesis and data) [17].

However, the ambiguities of CCMs are not reflected in their application. In pattern analysis, there does not exist any agreement in which CCMs to use. To exemplify, [3] combines “ARI” [18] with “Silhouette index”, “Jaccard index”,“Minkowski measure”, “Silhouette index”, “Dunns index” and “Davies-Bouldin index” to judge the cluster-accuracy of their proposed algorithm. [20] combines “FJ” with “ARI” in order to validate their new-proposed algorithm. The work of [4] combines “Rand’s Index” with “Calinski-Harabasz (VRC)” [6], “Silhouette Index” and “logSS”.

When research agrees in which CCMs to use, they disagree in how to interpret the CCM scores. To exemplify, the work of [2] asserts that a change in ARI [18] prediction score of *0.46* into *0.76* implies a significant difference in cluster accuracy. However, the authors do not discuss ambiguities in their gold standard, nor the significance of the scores. Measurements reveal how the difference may be explained by the unawareness of the metrics sensitivity (Table I).

To summarize, the established metrics for pattern matching are applied irrespective of their inaccuracy. The choice of strategy for pattern matching is applied without discussing the dependency between CCMs (Fig. 3), hence it is unknown when methods and algorithms are better than others.

Algorithm 1 An algorithm for unbiased selection of best-performing *pattern matching algorithm* in real-life data-sets. To simplify, the *selectMax(r_g, a, t, n, s)* method is omitted from the evaluation, a method which identifies the best-performing algorithm permutation.

```
procedure EVALUATE(ENSEMBLE)
    for each $a \in clustAlg$ do
        for each $t \in [0, 1]$ do    ▷ are we to t(transpose)?
            for each $n \in normMetrics$ do
                for each $s \in simMetrics$ do
                    $r_M$ = ccmMatrix(ensemble, a, t, n, s)
                    selectMax($r_M$, a, t, n, s)
                    $r_g$ = ccmGold(ensemble, a, t, n, s)
                    selectMax($r_g$, a, t, n, s)
procedure CCMMATRIX(ENSEMBLE, A, T, N, S)
    $ranks = [][]$ Ranks for 'not gold' CCM (Table I)
    for each $data \in Ensemble$ do
        clusters = a(data, t, n, s)
        for each $ccm \in matricCCM$ do
            ranks[ccm,data] = ccm(clusters, data)
        ranks[ccm] = rank(ranks[ccm])
    return ranks
procedure CCMGOLD(ENSEMBLE, A, T, N, S)
    $ranks = [][]$ Ranks for each 'gold' CCM (Table I)
    $clusters_0$ = a(Ensable[0], t, n, s)
    for each $data \in Ensemble$ do
        clusters = a(data, t, n, s)
        for each $ccm \in ccmGold$ do
            ranks[ccm,data] = ccm($clusters_0$, clusters))
        ranks[ccm] = rank(ranks[ccm])
    return ranks
```

IV. Method: new big-data tools for data perturbation and algorithm identification

This section describes a new method for capturing the trustworthiness of "Cluster Comparison Metrics (CCMs)" (code-listing 1):

1) data perturbations: a new approach and API to capture the accuracy of CCMs (Table I);
2) execution time: a strategy to reduce the time cost, hence a methodology supporting big-data analytic;
3) unbiased evaluation: a new algorithm which combines clustering with metric permutations to avoid bias in gold-data from influencing the prediction outcome.

The method enables the automated identification of best-performing metrics in an ensemble of data, hence its broad applicability. The algorithms are integrated into the "hpLysis" machine learning software [21]. Hence, users are provided with software for fast classification of large data-sets.

A. Data perturbations: a new API to detect accuracy and resolution of CCMs

Motivation is to trap the differences among CCMs. A large number of data topologies and CCMs argues for an automated approach, for which a new method and API for synthetic evaluation of CCMs is designed:

1) cluster shapes: construct different co-occurrence matrices and cluster partitions;
2) perturbations: compare *cluster shape* with exactly similar data topologies;
3) CCMs: apply permutations of the 30+ CCMs, and then select the extreme cluster predictions.

The strategy enables an automated and unbiased quantification of differences in CCM prediction.

B. Execution Time: the feasibility of big-data evaluation

The large number of CCMs requires an approach to reduce the computational complexity. The crux in CCM computation concerns the time cost of computing similarity metrics. The computation of CCMs involves the steps of 1) compute a covariance matrix (time: $O(n^3)$), and 2) similarities between features in Fig. 1 (time: $O(n^2)$).

The performance $O(n^3)$ issue is addressed through application of optimized implementation of matrix multiplication, as discussed in our earlier work [22]. The "hpLysis" software [21] provides fast access to the 320+ pairwise similarity metrics.

C. An algorithm for unbiased exploration of data

Code-listing 1 describes an algorithm for enabling the qunaitifciaotn of pattern matching algorithms. The algorithm takes as input data-sets with a well-defined order of predictions. Example input is a feature matrix combined with multiple hypotheses representing different data segmentation.

To avoid bias in clustering from hampering the prediction accuracy, the 20+ cluster algorithms supported by the hpLysis software [21] are combined with the 320+ established similarity metrics. Hence, the approach address issues in regression analysis.

The algorithm makes use of heuristics to reduce its execution time. The choice of the metrics is tuned towards different data ensembles. When partitioning the data-sets into clusters *three categories of cluster algorithms* are explored: threshold based cluster algorithms (*e.g.*, "DBSCAN" [23]); hierarchical cluster algorithms (*e.g.*, "SLINK"); randomized cluster algorithms (*e.g.*, "k-means"). For computation of the cluster algorithms the hpLysis software [21] is used, hence ensuring fast execution.

file	k means avg (Low)	k means avg (High)	k means rank (Low)	k means rank (High)	k means medoid (Low)	k means medoid (High)	hpCluster (Low)	hpCluster (High)	disjoint kdTree (Low)	disjoint kdTree (High)	disjoint kdTree CCM (Low)	disjoint kdTree CCM (High)	HCA single (Low)	HCA single (High)	HCA max (Low)	HCA max (High)	HCA average (Low)	HCA average (High)	HCA centroid (Low)	HCA centroid (High)	Kruskal HCA (Low)	Kruskal HCA (High)	k means altAlg miniBatch (Low	k means altAlg miniBatch (Higl	random best (Low)	random best (High)
msq	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0
chorSub	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
pulpfiber	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
randu	0.0	0.7	0.0	0.7	0.7	0.7	0.0	0.7	0.0	0.7	0.0	0.7	0.0	0.7	0.7	0.7	0.0	0.0	0.0	0.7	0.0	0.7	0.7	0.7	0.0	0.7
cf	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
airquality	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.5	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.7
UScrime	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
pottery	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
Hedonic	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
Melanoma	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
affect	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
Holzinger	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.8
smoking	0.2	0.7	0.2	0.7	0.7	0.7	0.2	0.7	0.2	0.8	0.2	0.8	0.2	0.7	0.7	0.7	0.2	0.2	0.2	0.8	0.2	0.8	0.7	0.7	0.2	0.7
airquality	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7	0.1	0.7
bfi	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.9	0.0	1.0	0.0	1.0	0.0	0.9
Hartnagel	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
attitude	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.7	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
gilgais	0.1	0.7	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
cancer	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
burt	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
votes	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9
attitude	0.1	0.6	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.6	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.6	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9
msq	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0
Arbuthnot	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8
LifeCycleSavings	0.1	0.5	0.1	0.8	0.8	0.8	0.1	0.8	0.1	0.5	0.1	0.8	0.1	0.8	0.8	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.8	0.8	0.1	0.8
phosphate	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8
alcohol	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.7	0.1	0.9	0.1	0.9	0.1	0.8	0.1	0.7	0.1	0.8	0.1	0.8	0.1	0.8	0.1	0.8
aldh2	0.1	0.8	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.2	0.9	0.1	0.9	0.1	0.9	0.1	0.9

Fig. 2. How the choice of metrics influences algorithms accuracy. The above table compares 20 clustering algorithms.

V. Evaluation Framework

The section identifies an experimental benchmark setup capturing the effect of data and metric perturbations. The framework provides an unbiased strategy for measuring the accuracy of different pattern matching algorithms (Section III).

A. Axis of variability: hypothesis testing & result ranking

An accurate benchmarking of CCMs for big-data analytic requires:

1) representative: investigate the CCMs applied in big-data analytic, *e.g.*, SSE [25] and Silhouette
2) features variation: data with different spread and skewness in both column features and row features;
3) configurations: different sizes of rows, columns, clusters, and score density.

Importantly, all of the proposed metrics share a set of common artifacts.

Fig. 1 exemplifies the *axis of variability* through the use of different shapes (Table I). The figure captures complexities in cluster analysis: to correctly describe the distance between vertices versus the arbitrary clusters *h*, *o*, and *t*.

Fig. 1 identifies how the *cluster within distance* is computed through permutations of $\sum d(vertex(...), vertex(...))$: when metrics such as VRC and Dunn's Index agrees in the prediction, it is due to the agreements between different interpretations of *minimum(between–within)* distance. Hence, for controlled topologies, it is sufficient to evaluate a small subset of the proposed cluster algorithms and metrics.

B. Core characteristics captured through representative data

Motivation is to identify representative data ensembles. Therefore, data-sets are explored for different perspectives:

1) controlled: synthetic clusters and hypothesis to evaluate the the linear relationship between hypothesis, data topology, and cluster segmentation;
2) real-life: an evaluation of 100+ real-life data-sets taken from [26].

The 100+ real-life data-sets. are modified through increased use of Gaussian noise. The application of linearly distorted data enables users to capture the effects of randomness, *e.g.*, when evaluating the CCMs ability to correctly rank different hypothesis.

The CCMs are evaluated through comparison of outputs from different algorithms. As input, the evaluation takes both randomized data and randomized cluster partitions. An issue concerns the different scores (and ranges) provided by metrics (Table I). To address the issue of different scales, the prediction results are ranked separately for each $[data\ permutations]$ x $metric$.

ISBN: 978-1-61208-708-5

Zero clusters for the SSE CCM:

vertices	1	2	3	4	5	6	7	8	9
0	0.58	0.22	0.17	0.18	0.17	0.16	0.17	0.17	0.17
100	0.56	0.17	0.11	0.09	0.08	0.08	0.08	0.08	0.08
300	0.56	0.16	0.09	0.07	0.06	0.05	0.05	0.05	0.05
600	0.56	0.15	0.08	0.06	0.05	0.04	0.04	0.04	0.04
1000	0.56	0.15	0.08	0.05	0.04	0.04	0.03	0.03	0.03
1500	0.56	0.15	0.07	0.05	0.04	0.03	0.03	0.03	0.03

Zero clusters for the Silhouette CCM:

vertices	1	2	3	4	5	6	7	8	9
10	0.57	0.72	0.67	0.65	0.6	0.6	0.6	0.6	0.6
100	0.5	0.89	0.87	0.87	0.87	0.87	0.87	0.87	0.86
300	0.5	0.93	0.92	0.92	0.92	0.92	0.92	0.92	0.92
600	0.5	0.95	0.95	0.94	0.94	0.94	0.94	0.94	0.94
1000	0.5	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95
1500	0.5	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.96

Zero clusters for the Dunn CCM:

vertices	1	2	3	4	5	6	7	8	9
10	0.43	0.04	0.13	0.04	0.04	0.04	0.04	0.04	0.04
100	0.5	0	0	0.01	0.01	0.01	0.01	0.01	0.01
300	0.5	0	0	0	0	0	0	0	0
600	0.5	0	0	0	0	0	0	0	0
1000	0.5	0	0	0	0	0	0	0	0
1500	0.5	0	0	0	0	0	0	0	0

Zero clusters for the VRC CCM:

vertices	1	2	3	4	5	6	7	8	9
10	19.55	1.22	0.62	0.39	0.27	0.13	0	0	0
100	141.6	0.79	0.62	0.49	0.41	0.33	0.27	0.23	0.19
300	379.45	0.52	0.46	0.41	0.35	0.3	0.26	0.23	0.2
600	739.44	0.38	0.36	0.33	0.3	0.27	0.24	0.22	0.19
1000	1,221.6	0.3	0.3	0.29	0.27	0.25	0.22	0.2	0.18
1500	1,819.44	0.25	0.25	0.25	0.24	0.22	0.21	0.19	0.18

Fig. 3. Prediction difference when no clusters are present. The above figure captures the result (of pattern matching algorithms) when no clusters are present in the input data.

Table I introduce the notation of *Gold*. The *Gold* column refers to cases where two hypothesis (*e.g.*, for the use-case where cluster partitions are compared), which is an alternative to comparing a hypothesis with a feature matrix. On the other hand, the table's *Equal Clusters (Equal)* column identifies the metrics sensitivity to comparing two equal cluster partitions, an effect is compared to the case where they do not *(Diff.)*. In the measurements, each algorithm is evaluated across multiple feature matrices and multiple hypothesis (*e.g.*, the result of a cluster algorithm).

VI. RESULT: EMPIRICAL EVALUATION

This paper presents an automated approach for unbiased evaluation of 30+ CCMs. For brevity, the details of the 30+ CCMs are included in the benchmark scripts (appended into the hpLysis software). The results reveal how the proposed method outperforms established metrics and algorithms (Fig. 2), answering questions such as:

1) 30x+: How CCMs differ in their prediction scores? (Fig. 3);
2) 4x+: How differences in data size influence the CCM score? (Table II);
3) 0x–79x: Is SSE able to separate between *false* versus *true* hypothesis? (Table I).

While the above results are specific for the evaluated topologies, they capture the pitfall of making strong conclusions from inaccurate pattern metrics.

A. How disagreements in CCMs capture topological features

The motivation of CCMs is to grasp the differences between data using a few numeric indicators (Table II): to apply independent metrics to capture similarities and distortions in data.

The correct applicability of CCMs depends on both the datasets and the gold standards (Table I). A linear increase in random perturbations is not recognized in the Davids-Bouldin metric. In contrast, SSE and Silhouette detect a variation in data perturbed with Gaussian noise. When compared to Silhouette and SSE, the VRC metric is distinctively different. The results demonstrate how the combination of different CCMs provides users with a unique ability to reject a false hypothesis: there is no uniform agreement in which CCMs to select.

TABLE II
HOW CATEGORIZATION OF CCMS REVEALS UNDERLYING DATA TOPOLOGY. THE TABLE SUMMARIZES THE OBSERVATIONS FROM FIG. 3: WHILE $n = 10$ REFERS TO A MATRIX WITH ROWS=COLUMNS=10, $n = 1500$ CAPTURES THE RESULT OF EVALUATING A MATRIX WITH ROWS=COLUMNS=1500; "THE *n=10 – n=1500*" DESCRIBES THE RELATIVE DIFFERENCE BETWEEN *worst–best*; THE "*worst–best*" COLUMNS IDENTIFIES THE SPREAD IN CCM SCORE; THE "*best:column*" IDENTIFIES THE HYPOTHESIS WHICH IS FARTHEST AWAY FROM THE INPUT DATA.

CCM	n=10 – n=1500	n=10: worst–best	n=1500: worst–best	best: column
SSE	2.6x – 3.5x	0.22 – 0.58	0.56 – 0.15	6
Silhouette	1.3 – 2x	0.7 – 0.57	0.97 – 0.50	2
Dunn	10.5 – ∞	0.42 – 0.04	0.58 – 0.00	*all*
VRC	16.1x – ∞	19.6 – 1.22	1819.4 – 0.18	9

B. The correct choice of CCM provides accurate predictions

The performance of CCMs is determined by:

1) metric choice: 10x+ difference when using "Davids-Bouldin" instead of "Dunn's" or "Euclidean" (Table I);
2) topology sensitivity: while metrics such as VRC are highly sensitive to score perturbations, metrics such as Silhouette and SSE provides higher granularity (as derived from underlying measurements);
3) score difference: a 100x+ score-difference between matrix with rows=columns=[10, 1500] (Fig. 3).

The above differences is due to the metrics definition, hence the importance of relating CCMs to topolgoical traits. An example is an assumption that the *within cluster distance* has a uniform distribution, for which CCMs, such as VRC and Dunn's Index, becomes overlapping. When evaluating the algorithm for CCM identification (code-listing 1), the results demonstrates how the new-identified CCMs outperforms metrics for regression analysis. Hence, the benchmark of cluster quality metrics improves the broad turf of *regression analysis*.

Fig. 3 demonstrates how CCMs have different score sensitivity: while VRC indites an $= 1819.4/0.18 = \infty$ separation between *correct hypothesis* versus *wrong hypothesis*, SSE has a sensitivity of $0.56/0.04 = 14x$, as summarized in Table II.

C. Summary: pattern recognition versus data topologies

The measurements identifies the importance of applying *independent CCMs* to capture differences in data-sets and cluster predictions. Hence, the choice of CCMs should reflect the given use-case. For the same data, there is a 5x prediction difference between "Fred & Jain" [27] versus Davies-Bouldin (Table I). Similarly, SSE and Silhouette disagrees in which of the cluster prediction is the best (Fig. 3).

While the established strategy is to apply CCMs irrespective of the topologies, this paper has demonstrated how the implicit assumptions of data topology directly influences the accuracy of pattern matching (Fig. 2). While this knowledge is known among authors of algorithms (*e.g.*, [23]), users of pattern matching are unaware of these findings (Section III).

VII. Conclusion and Future Work

In this paper, we have identified how established pattern matching strategies in big data suffers from bias. The 30x+ inaccuracy of metrics for pattern recognition goes undetected in large research projects (Fig. 3). The proposed benchmark software enables quantification of the trustworthiness of established recommendations (Section IV).

Importantly, the approach may be applied for big data-sets, which is due to the combination of optimized software implementation and heuristics deduced from metrics (code-listing 1). The benchmark deduces fast and accurate cluster quality metrics: code-listing 1 identifies the metric combination to be used (for a given data ensemble), hence enabling an increase in the accuracy of algorithms.

The new methodology and software provide users with a tool enabling the insight into when and how data is captured by hypothesis: to identify patterns which are otherwise hidden. The results highlight the importance of not always following the established guidelines for cluster validity.

In the future, we plan to apply the proposed method and benchmark to the 1000+ recently proposed cluster algorithms, hence easing the applicability of our findings into all turfs relying on pattern matching algorithms.

Acknowledgements

The authors would like to thank Dr. O.V. Solberg at SINTEF, Dr. B.H. Helleberg at NTNU–medical, K.T. Dragland at NTNU, and MD K.I. Ekseth at UIO for their support.

References

[1] Ekseth, O.K., Meyer, J.C., Hvasshovd, S.O.: A new database for drug discovery through application of data-integration and semantics. In: Semantic Computing (ICSC), 2018 IEEE 12th International Conference On, pp. 403–410 (2018). IEEE

[2] Hahsler, M., Bolaños, M.: Clustering data streams based on shared density between micro-clusters. IEEE Transactions on Knowledge and Data Engineering **28**(6), 1449–1461 (2016)

[3] Chiu, T.-Y., Hsu, T.-C., Yen, C.-C., Wang, J.-S.: Interpolation based consensus clustering for gene expression time series. BMC bioinformatics **16**(1), 1 (2015)

[4] Lord, E., Diallo, A.B., Makarenkov, V.: Classification of bioinformatics workflows using weighted versions of partitioning and hierarchical clustering algorithms. BMC bioinformatics **16**(1), 1 (2015)

[5] Yazdani, M., Chow, J., Manovich, L.: Quantifying the development of user-generated art during 20012010. PLOS ONE **12**(8), 1–24 (2017). doi:10.1371/journal.pone.0175350

[6] Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 265–276 (2000). Springer

[7] Garla, V.N., Brandt, C.: Semantic similarity in the biomedical domain: an evaluation across knowledge sources. BMC bioinformatics **13**(1), 261 (2012)

[8] Solberg, O.V., Lindseth, F., Torp, H., Blake, R.E., Hernes, T.A.N.: Freehand 3d ultrasound reconstruction algorithmsa review. Ultrasound in medicine & biology **33**(7), 991–1009 (2007)

[9] Bhaskaran, K., Smeeth, L.: What is the difference between missing completely at random and missing at random? International journal of epidemiology **43**(4), 1336–1339 (2014)

[10] Ferrari, D.G., De Castro, L.N.: Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. Information Sciences **301**, 181–194 (2015)

[11] Lau, L., Yang-Turner, F., Karacapilidis, N.: Requirements for big data analytics supporting decision making: A sensemaking perspective. In: Mastering Data-Intensive Collaboration and Decision Making, pp. 49–70. Springer, ??? (2014)

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**(Oct), 2825–2830 (2011)

[13] Fan, W., Bifet, A.: Mining big data: current status, and forecast to the future. ACM sIGKDD Explorations Newsletter **14**(2), 1–5 (2013)

[14] Szalkai, B.: Generalizing k-means for an arbitrary distance matrix. arXiv preprint arXiv:1303.6001 (2013)

[15] He, Z., Deng, S., Xu, X.: Approximation algorithms for k-modes clustering. In: International Conference on Intelligent Computing, pp. 296–302 (2006). Springer

[16] Ji, J., Bai, T., Zhou, C., Ma, C., Wang, Z.: An improved k-prototypes clustering algorithm for mixed numeric and categorical data. Neurocomputing **120**, 590–596 (2013)

[17] Ekseth, O.K., Gribbestad, M., Hvasshovd, S.-O.: Inventing wheels: why improvements to established cluster algorithms fails to catch the wheel. In: The International Conference on Digital Image and Signal Processing (DISP19), Springer (2019)

[18] Yeung, K.Y., Ruzzo, W.L.: Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics **17**(9), 763–774 (2001)

[19] Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics **4**(1), 95–104 (1974)

[20] Zhao, W., Chen, J.J., Perkins, R., Wang, Y., Liu, Z., Hong, H., Tong, W., Zou, W.: A novel procedure on next generation sequencing data analysis using text mining algorithm. BMC bioinformatics **17**(1), 1 (2016)

[21] Ekseth, Ole Kristian: hpLysis: a high-performance software-library for big-data machine-learning. https://bitbucket.org/oekseth/hplysis-cluster-analysis-software/. Online; accessed 06. June 2017

[22] Ekseth, O.K., Hvasshovd, S.-O.: How an optimized DBSCAN implementation reduce execution-time and memory-requirements for large data-sets. (2017)

[23] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al.*: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)

[24] Sibson, R.: Slink: an optimally efficient algorithm for the single-link cluster method. The computer journal **16**(1), 30–34 (1973)

[25] Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

[26] Arel-Bundock, V.: Rdatasets r datasets: An archive of datasets distributed with r, 2014. URL http://vincentarelbundock. github. io/Rdatasets

[27] Ana, L., Jain, A.K.: Robust data clustering. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference On, vol. 2, p. 128 (2003). IEEE