

How far are we from a satisfactory theory of Clustering ?

Shai Ben-David
University of Waterloo

“Hammers and Nails”, Weizmann Institute, July 2017

High level view of Science

***“The purpose of science is
to find **meaningful simplicity**
in the midst of
disorderly complexity”***

Herbert Simon

*This can also serve to describe the goal of
clustering*

The Theory-Practice Gap

Clustering is one of the most widely used tool for exploratory data analysis.

Social Sciences

Biology

Astronomy

Computer Science

•

•

All apply clustering to gain a first understanding of the structure of large data sets.

*Yet, there exist distressingly little
theoretical **understanding** of clustering*

Understanding the clustering **task**

- Not just analyzing a *particular algorithm* (say EM or K-Means ++)
- Not just optimizing a particular *cost function* (say, k-means).
- Not just estimating particular data *generating model* (say mixture of Gaussians).

Lack of Clustering Theory

Pick any Machine Learning course or textbook.
E.g., the Stanford ML (CS 229) syllabus:

Learning theory. (3 classes)

Bias/variance tradeoff. Union and Chernoff/Hoeffding bounds. VC dimension. Worst case (online) learning.

Practical advice on how to use learning algorithms.

Unsupervised learning. (5 classes)

Clustering. K-means.

EM. Mixture of Gaussians.

Factor analysis.

PCA (Principal components analysis). ICA (Independent components analysis).

Overview of this talk

Two different topics, highlighting aspects that we know too little about.

- 1) **Model (tool) selection issues:** How would you choose the best clustering algorithm for your data? How should you set its parameters (e.g., the number of clusters)?
- 2) **The computational complexity of clustering:** The discrepancy between the theoretical hardness of clustering and practice.

The first question we address:

Given a clustering task,

How should a suitable
clustering paradigm be chosen?

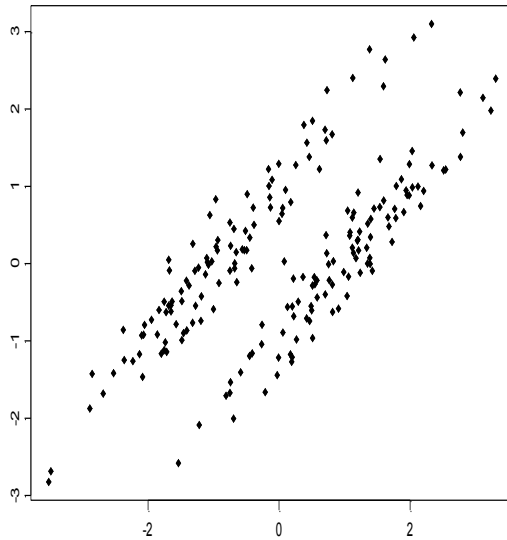
Motivation

Given a concrete clustering task, the user needs to choose a clustering algorithm, as well as its parameter values.

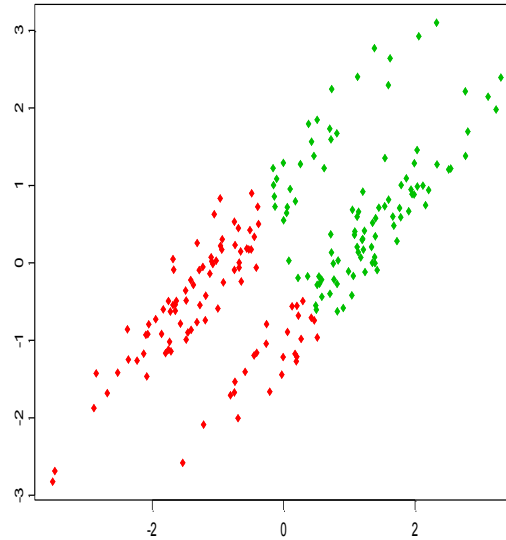
Unlike other common computational tasks, different choices may lead to significantly different clustering outcomes.

An example

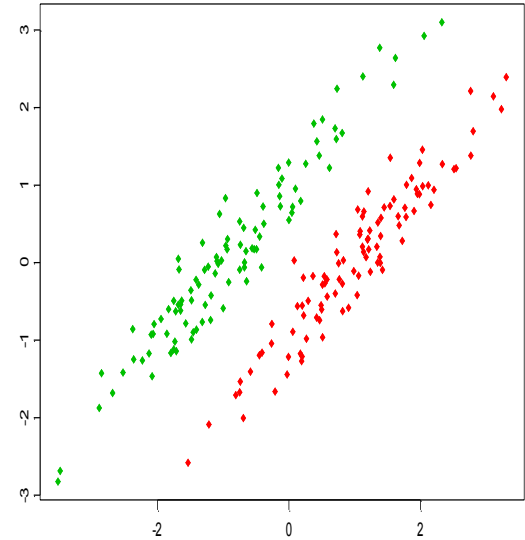
2-d data set



Compact partitioning into two strata



Unsupervised learning



The agreed upon Clustering “Definition”

“Partition the given data set so that

- 1. similar points reside in same cluster*
- 2. non-similar points get separated.”*

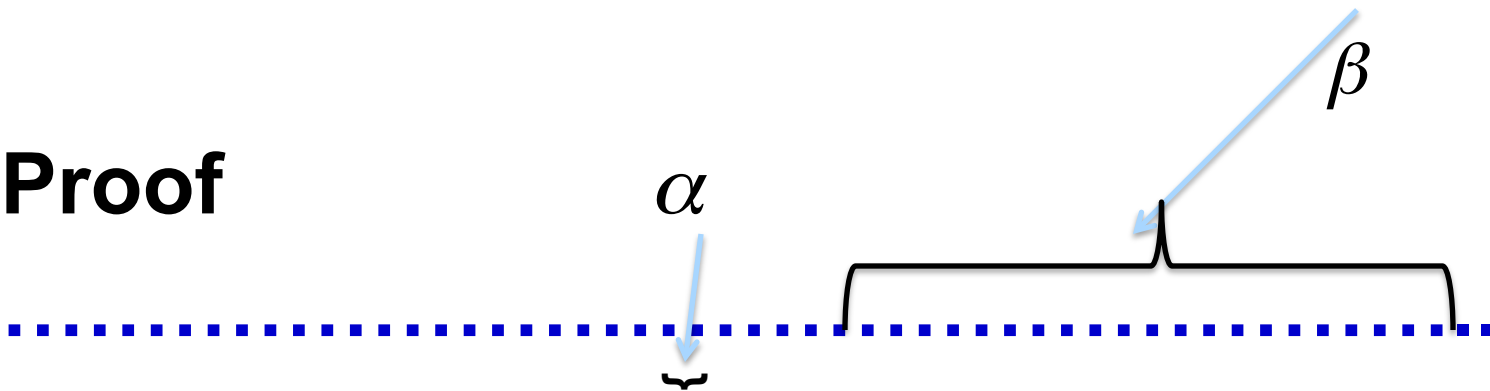
Often, these two requirements cannot be simultaneously met.

The above “definition” does not determine how to handle such conflicts.

A very basic impossibility result

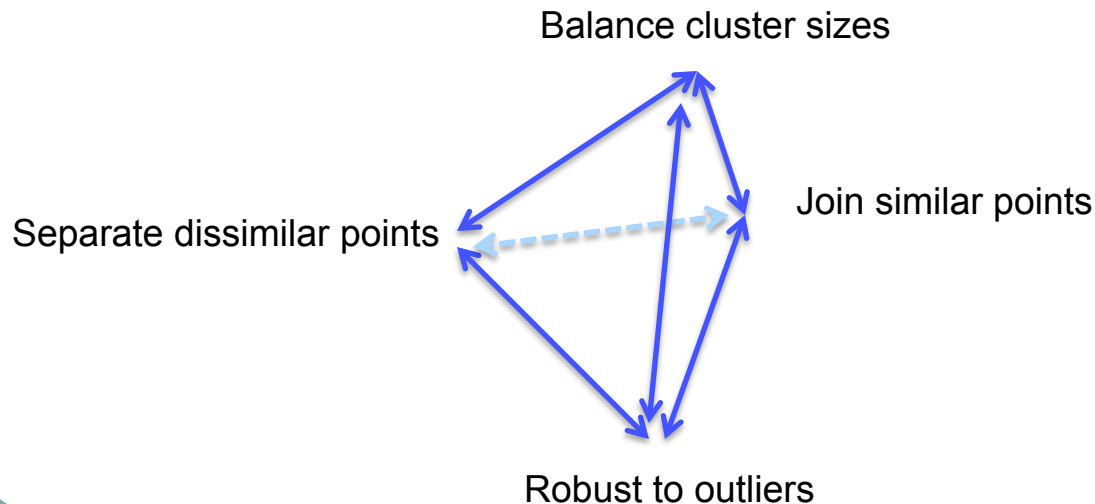
Observation: Pick any values, $\alpha \ll \beta$. *Is there a clustering function that is guaranteed to cluster together every pair of closer than α points and separate every pair with larger than β distance??*
NO!

Proof



Balancing conflicting requirements

One can think of any given clustering algorithm as a point in a simplex whose vertices are the different "desirable" requirements.



Different clustering tools pick different tradeoffs - Examples

- *Single Linkage* – focus on “similar points same cluster”
- *Max Linkage* – focus on “dissimilar points should not share a cluster”

Both are oblivious to balancing cluster population sizes.

- *K-Means* – “balance clusters and avoid having dissimilar points together”.
- *Min-Sum* – Different emphasis on balance.

Different applications call for different values of tradeoffs

- De-duplication of records in a data base -- emphasis on separating dissimilar point.
- Clustering people for predicting viral spread (of disease or rumors) – emphasis on clustering similar points together.
- Clustering neighborhoods to school districts – need for balance between sizes of clusters.

Some more practical examples

“Binning” by Locality-Sensitive Hashing (LSH):

When data sets are very large, clustering is used to save the need to compare all pairs of elements (for De-Duplication or for Nearest Neighbor search or for “Top-k” in query answering).

Here, focus should be on “similar points same cluster”

Choosing a clustering paradigm in practice

How do users **actually** pick a tool for their data?

Currently, in practice, **this is done by most ad-hoc manner.**

Current practices

In practice users pick a clustering method based on:

“Easiness of use – no need to tune parameters”,

“Freely downloadable software”,

“It worked for my friend” (for a different problem ...),

“Runs fast”

etc.

Some common fallacies

“My algorithm outperforms all others”

*“This medication is best,
regardless of what your sickness is!”*

*(There can be no universally-best
clustering algorithm!)*

*“Here’s an algorithm that can be
implemented in linear time”*

“Sick? Take this pill, it’s only \$0.99”

Some more common fallacies

What about “*check against inherent structure in the data*”?

Sure, if your data looks like a collection of well separated dense rounded clouds.

What about **experimental evidence**?

For every two reasonable clustering algorithms there are data sets in which one performs “better” than the other.

From “Science” last fall

RESEARCH | R1

NETWORK SCIENCE

Higher-order organization of complex networks

Austin R. Benson,¹ David F. Gleich,² Jure Leskovec^{3*}

Networks are a fundamental tool for understanding and modeling complex systems in physics, biology, neuroscience, engineering, and social science. Many networks are known to exhibit rich, lower-order connectivity patterns that can be captured at the level of individual nodes and edges. However, higher-order organization of complex networks—at the level of small network subgraphs—remains largely unknown. Here, we develop a generalized framework for clustering networks on the basis of higher-order connectivity patterns. **This framework provides mathematical guarantees on the optimality of obtained clusters and scales to networks with billions of edges.** The framework reveals higher-order organization in a number of networks, including information propagation units in neuronal networks and hub structure in transportation networks. Results show that networks exhibit rich higher-order organizational structures that are exposed by clustering based on higher-order connectivity patterns.

in instances of M that reside in S . Equating this to the conductance metric from spectral graph theory, one of the most useful for graph partitioning scores (11). We refer to $\phi_M(S, l)$ as the motif conductance of S with respect to l .

Finding the exact set of nodes S that maximizes the motif conductance is computationally intractable (12). To approximately minimize Eq. 1 and identify higher-order clusters, we develop an optimization framework that provably finds optimal clusters [supplementary material S1]. We extend the spectral graph clustering algorithm, which is based on the eigenvalues and eigenvectors of matrices associated with the graph, to account for higher-order structures in networks. The resulting method maintains the properties of traditional spectral graph clustering: computational efficiency, ease of implementation, and mathematical guarantees on the near-optimality of clusters. Specifically, the clusters identified by our higher-order clustering framework satisfy the Cheeger inequality (14), which means that our optimization framework finds clusters that

* Networks are a standard representation of complex systems. We minimize the following ratio:

Need for Domain Specific Bias

To turn clustering into a well-defined task, one needs to add some *bias*, expressing some *prior domain knowledge*.


Two approaches to the tool-selection challenge

“Axiomatic” (property-based): formulate properties of clustering functions that allow translating prior knowledge about a clustering task into guidance concerning the choice of suitable clustering functions.

Interactive: Ask the user to provide partial information about the desired outcome on their given data interactively with the algorithm.

The axiomatic approach: *taxonomy of clustering paradigms*

- The goal is to generate a variety of axioms (or properties) over a fixed framework, so that different clustering approaches could be classified by the different subsets of axioms they satisfy.



	Scale Invariance	Antichain Richness	Local Consistency	Full Consistency	Richness	
Single Linkage	+	+	+	+	-	
Center Based	+	+	+	-	+	
Sum of Distances	+	+	+	+	-	
Spectral	+	+	+	+	-	
Silly F	+	+	-	-	+	

Main challenge for the Axiomatic approach

How to come up with properties that make sense to a clustering “customer”.

*A language that bridges between
algorithmic theory and practical
applications.*

The **interactive** approach:

Possible types of user input info

- *Must-link/Can't link* pairs of instances.
(user driven, random, or active learner queries)
- *Merge/Split* clusters on a proposed clustering.
- *Sample clusterings* of small input subsets.

Semi-Supervised-Active-Clustering

Recent work [Ashtiani, BD Kushagra '16]

Consider algorithms that interact with a domain expert by actively querying “**same-cluster/diff-clusters**” over pairs of data points.

- We show that access to few ($O(k \log n)$) such query answers can **turn an NP hard clustering problem to one solvable in linear time.**

Two new considerations

- Computational complexity of clustering tasks.
- Data niceness (“clusterability”) assumptions.

A Clusterability Condition

Given a data set (X, d) and parameter $\gamma > 0$,
a clustering $C = (C_1, \dots, C_k)$ induced by centers
 (μ_1, \dots, μ_k) is **γ -margin separable** if, for any i and
 x in C_i , z not in C_i , $\gamma d(x, \mu_i) < d(z, \mu_i)$.

- A data set (X, d) is **γ -margin nice** for k -means
if it has an optimal k -means clustering C that is
 γ -margin separable.

A Phase-transition phenomena

Theorem [Ashtiani, BD Kushagra '16]:

- There exists a **polynomial time** algorithms that, for every $\gamma > 2$ finds an optimal k-means clustering for every data set that is γ -margin nice.
- For every $\gamma < 2$ the problem of finding an optimal k-means clustering for every data set that is γ -margin nice is **NP hard**.

Our main result

- We design a *probabilistic active semi-supervised* clustering algorithm, $A(\gamma, k)$, such that for every $\gamma > 1$ and every input data set (X, d) and a k -clustering C satisfying the γ -margin condition,

upon making $O(k^2 \log(k) + k \log(n))$ queries of a C -oracle $A(\gamma, k)$ runs in time $O(kn \log(n))$ and w.h.p. outputs the clustering C .

Unusual conclusion

Note that for every $1 < \gamma < 2$ our algorithm demonstrates that access to a small number of *Same-cluster/Diff-cluster* active query answers,

turns an NP-hard clustering problem into one solved in polytime.

Basic algorithm's idea

- **Step1** – sample enough points from X to get “many” in one cluster.
- **Step 2** – estimate the center of that cluster.
- **Step 3** – binary search the “radius” of that cluster.
- **Step 4** – delete all members of X in ball.
- Repeat $k-1$ times.

Part 2: Computational complexity

Is it the case that

“clustering is hard only when
it does not matter”?

A note on Worst-Case Complexity

Worst case complexity is by far the most cited, most researched, best understood, approach to analyzing the difficulty of computational tasks.

However, it's focus on hard, possibly rare, instances, makes it excessively pessimistic

Theoretically hard

Practically feasible

- Propositional Satisfiability (SAT)
- Linear Programming
- Neural Network Training
- *K-means clustering*

Focus on clustering

The most common clustering objectives are NP-hard to optimize (e.g., k-means).

Does this hardness still apply when we restrict our attention to “clusterable” inputs?

Is it the case that “Clustering is Difficult only when it Does Not Matter” (CDNM thesis)?

Outline of this part

- 1) I will start by listing **requirements on notions of clusterability** aiming to sustain the CDNM thesis.
- 2) List various **clusterability notions** that have been recently proposed in this context.
- 3) **Examine** those notions in view of the above requirements.
- 4) *Conclusions, open problems and directions for further research.*

Desiderata for notion of “Clusterable” inputs

1. It should be reasonable to assume that most (or at least a significant proportion) of the inputs one may care about in practice meet the clusterability requirement.
- While there is no way to guarantee that the property will be satisfied by future meaningful inputs, it can serve to eliminate too restrictive notions.
 - May be checked against common generative models.

Desiderata for notion of “Clusterable” inputs

2. There should be *efficient algorithms* that are guaranteed to find a good clustering for any “clusterable” input.

Further requirements

3. *There* should be an efficient algorithm that, given an input, figures out *whether the input is “clusterable” or not*.

Note that in contrast to other computational tasks, checking if a given clustering is indeed optimal is generally not feasible.

Last requirement

4. *Some* commonly used practical algorithm can be guaranteed to perform well (i.e., run in polytime and find close-to-optimal solutions) on all clusterable instances.

This requirement is important when our goal is to **understand** what we witness in practice.

The main open question

Can we come up with a notion of clusterability that meets the above requirements (or even just the first two)?

Recently proposed clusterability notions

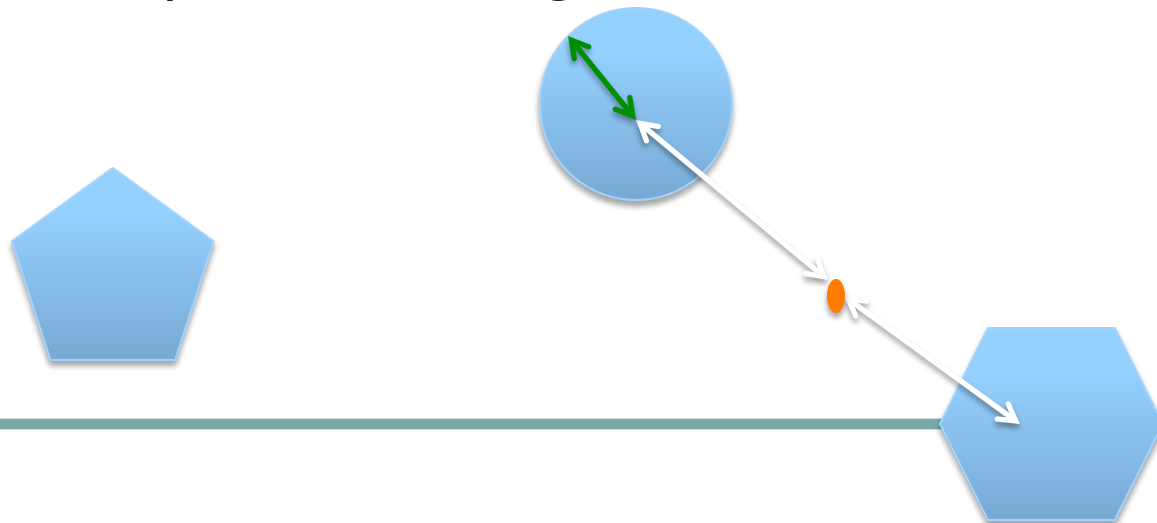
1. Perturbation Robustness(PR) – data set $I=(X,d)$ is robust if *small perturbations of I do not result in changes to its optimal clustering.*

1a. **Additive PR** [Ackerman-BD 2009] - the perturbation may move every point in X by some *bounded distance*.

1b. **Multiplicative PR** [Bilu-Lineal 2010] - the perturbation may change every pairwise point distance by a bounded *multiplicative factor*.

2. Significant loss when reducing the number of clusters

2a. *ϵ -Separatedness* [Ostrovsky et al. 2012]:
an input data set (X, d) to be ϵ -separated for k if the k -means cost of the optimal k -clustering of (X, d) is less than ϵ^2 times the cost of its optimal $(k - 1)$ -clustering.



More notion of “well behaved” clustering inputs

Uniqueness of optimum [Balcan et al. 2013]:

(X, d) is (c, ε) -approximation- *stable* if every clustering C of X whose objective cost over (X, d) is within a factor c of that the optimal clustering, is ε -close to $\text{OPT}(X)$ w.r.t. some natural notion of between-clusterings distance.



More notion of “well behaved” clustering inputs

α -center stability: [Awasthi et al. 2012]:
instance (X, d) is α -center *stable* (with respect to some center based clustering objective) if for any optimal clustering with centers c_1, \dots, c_k , for every $i \leq k$ and every $x \in C_i$, and every $j \neq i$, $\alpha d(x, c_i) < d(x, c_j)$.

Namely, points are closer to their own cluster center by a factor α more than to any other cluster center.

How do these notions fare w.r.t. the list of desirable properties?

- 1) All of these notions imply the **existence** of efficient clustering algorithms (weaker efficiency for APR).
- 2) **None** of them can be efficiently verified.
- 3) Only the ϵ -*Separatedness* gets efficiency for a (semi-) practical algorithms.
- 4) However, **all** (except maybe APR) seem to fail the requirement of being realistic.

What do I mean by “not a realistic clusterability requirement”?

- ϵ -Separatedness [Ostrovsky et al. 2012]

Implies polytime clustering only when the *minimal between-cluster-centers distance* is **> 200 times** the *average distance from a point to its cluster center*.

What do I mean by “not a realistic clusterability requirement”?

For ***Uniqueness of optimum*** [Balcan et al. 2013]: The parameter values sufficient for showing efficiency of clustering imply that the distance of a point to any “foreign” cluster center is larger than its distance to its own cluster center by at least **20 times** the average point-to-its-cluster-center distance.

Provable reason for concern

- The proofs of efficiency for all of the notions (except the APR), rely on showing that they imply α -center stability for some large α .
- However, [Ben-David, Reyzin 2014] show that *for any $\alpha < 2$, solving α -center stable inputs is NP-hard.*
- *2-center stable data sets are still “unrealistically nice”*

The bottom line

The proposed notions provably detects **easy-to-cluster instances**,
but those are **not the “realistic” inputs**.

The current approach to define input niceness that will render **efficiency w.r.t. the number of clusters, k** , seems to be **inherently too restrictive**.

Alternative directions (1)

- All the current approaches that try to tackle the hardness of finding a **minimal cost** (a.k.a. **optimal**) clustering.
- *Is that really required in practice?*

Definitely not!

Alternative directions (2)

Should one really care about an exact number of clusters when that number is high?

Consider clustering for record de-duplication in data repositories.

The number of resulting clusters is huge, but it is not set in advance.

Also, not captured by common regularization

Further big open questions

1. *Can similar approaches be applied to other worst-case hard problems that are being routinely solved in practice?*
2. In particular, can we find a notion of “input niceness” that will explain the practice of Propositional SAT problem?
3. Will the new analyses lead to new useful algorithms?

Thanks for listening

