Theoretical Foundations of Clustering – few results, many challenges

Shai Ben-David
University of Waterloo

MLSS, Beijing, June 2014

High level view of (Statistical) Machine Learning

"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

My focus: Theoretical approach

Why care about theory??

- To provides **performance guarantees**.
- To motivate and direct **algorithm development.**
- To **understand** what we are doing.

Overview of this tutorial

1) What is clustering? Can we formally define it?
2) Model (tool) selection issues: How would you chose the best clustering paradigm for your data? How should you choose the number of clusters?
3) Computational complexity issues: Can good clustering be efficiently computed?

Part 1

What **is** clustering?

The agreed upon “definition”

“Partition the given data set so that

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

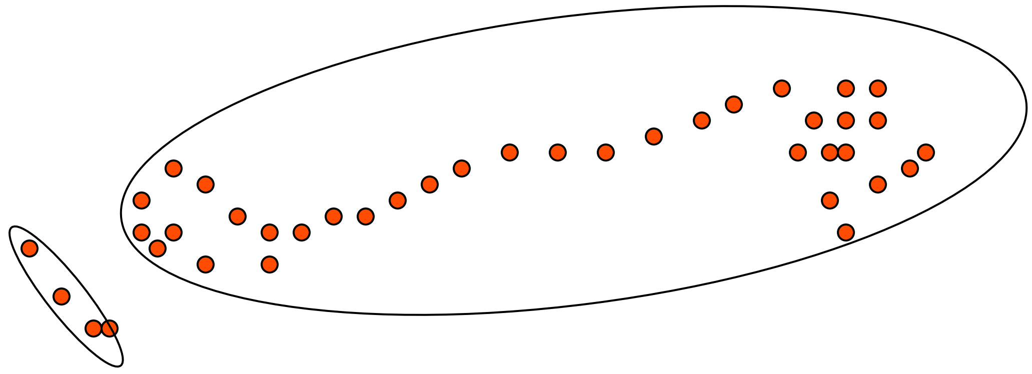
However, usually these two requirements cannot be met together for all points.

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

Consequently, there may be many clustering options

Clustering is not well defined.

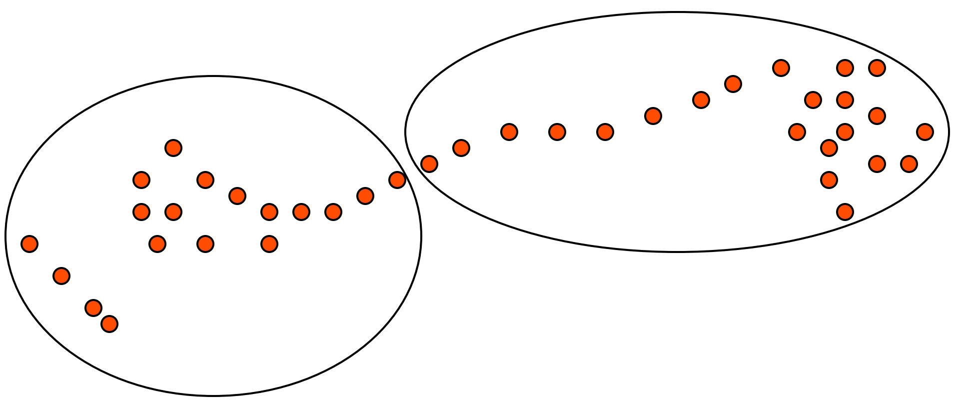
There is a wide variety of different clustering tasks, with different (often implicit) measures of quality.



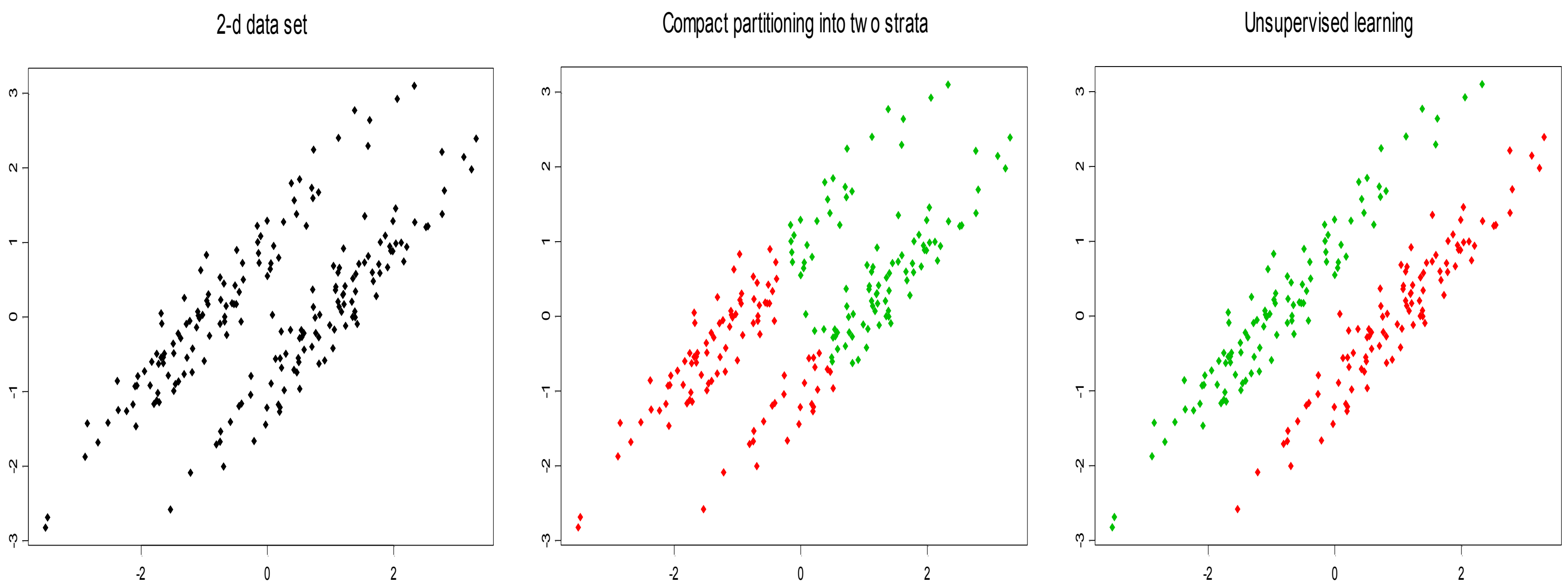
Consequently, there may be many clustering options

Clustering is not well defined.

There is a wide variety of different clustering tasks, with different (often implicit) measures of quality.



Some more examples



Some real examples of clustering ambiguity:

- Cluster paintings

  by painter vs. topic
- Cluster speech recordings

  by speaker vs. content
- Cluster text documents

  by *sentiment vs. topic*
-
-

Inherent obstacles

- Clustering is not well defined.

There is a wide variety of different clustering tasks, with different (often implicit) notions of clustering quality.

- In most practical clustering tasks there is **no clear ground truth** to evaluate your solution by.

(*in contrast with classification tasks, in which you can have a hold-out labeled set to evaluate the classifier against*).

Common Solutions

Postulate some objective (utility) functions – *Sum Of In-Cluster Distances, Average Distances to Center Points, Cut Weight, etc.*

Consider a restricted set of data generating distributions (generative models):

E., g, Mixtures of Gaussians

[Dasgupta '99], [Vempala,, '03], [Kannan et al '04], [Achlitopas, McSherry '05].

Add structure:

Relevant Information –

"Information Bottleneck" approach [Tishby, Pereira, Bialek '99]

Common Solutions (2)

Axiomatic approach:

Postulate ***'clustering axioms'***

that, ideally*,* every clustering approach should satisfy -

So far, usually conclude negative results

(e.g. [Hartigan 1975], [Puzicha, Hofmann, Buhmann '00], [Kleinberg '03]).

Quest for a general Clustering theory

What can we say independently of any

particular *algorithm,*

particular *objective function*

or specific *generative data model*

?

Questions that research of **fundamentals of clustering** should address

- Can clustering be given an *formal* and *general* definition?
- What is a “good” clustering?
- Can we distinguish “clusterable” from “structure-less” data?
- Can we distinguish meaningful clustering from random structure?
- Given a clustering task, how should a user choose a suitable clustering algorithm?

Defining what clustering is

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

We shall address several frameworks for formalizing such bias.

Outline of the (rest of the) talk

Out of the many research directions, I shall focus on the following:

1. Foundations: What **is** clustering? Can we formalize a No-Free-Lunch theorem for it?
2. Developing guidelines for choosing task-appropriate clustering tools.
3. Understanding the practical complexity of clustering – Is clustering easy for any clusterable input data?.

Basic Setting for the Formal Discussion

Definition: A *dissimilarity function* (*DF*) over some domain set ***S***, is a mapping, ***d:SxS*** $\rightarrow$ ***R***$^+$, such that: ***d*** is symmetric, and ***d(x,y)=0*** iff ***x=y***.

- Our Input: A dissimilarity function over some domain ***S*** (or a matrix of pairwise 'distances" between domain points)
- Our Output: A partition of ***S***.
- *We wish to define the properties that distinguish clustering functions from other functions that output domain partitions.*

The clustering-function approach - Kleinberg's Axioms

- *Scale Invariance*

  $F(\lambda d)=F(d)$ for all d and all strictly positive λ.

- *Richness*

  For any finite domain S,

  $\{F(d): d$ *is a DF over* $S\}=\{P: P$ *a partition of* $S\}$

- *Consistency*

  If d' equals d, except for shrinking distances within clusters of $F(d)$ or stretching between-cluster distances , then $F(d)=F(d')$.

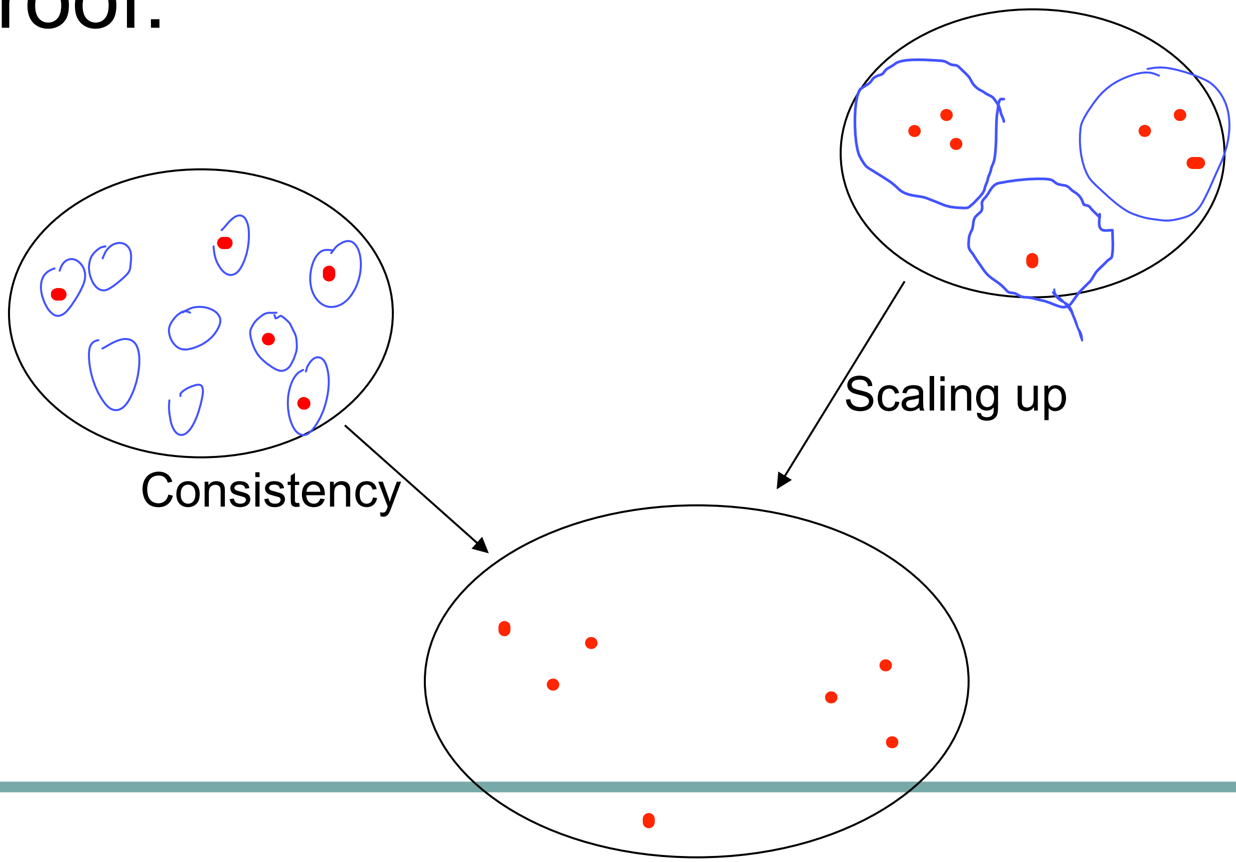
The “Surprising” result

Theorem: There exist no clustering function (that satisfies all of the three Kleinberg axioms simultaneously).

Kleinberg's Impossibility result

There exist no "clustering function"

Proof:



What is the Take-Home Message?

A popular interpretation of Kleinberg’s result is (roughly):

“It’s Impossible to axiomatize clustering”

But, what that paper shows is (only):

These specific three “axioms”, phrased in terms of clustering functions, do not work.

Open questions

We believe that no clustering algorithm can meet all desirable properties.

1. Can we back up this belief by some formal result? Come up with a list of ”really desirable” clustering properties that cannot be simultaneously met.
2. Can we get a Kleinberg style impossibility result for the framework in which the number of clusters k is part of the input?

Ideal Theory

- We would like the ***axioms*** to be such that:

  1. *Any clustering* method satisfies *all* the axioms,

  and

  2. *Any function* that is clearly not a clustering fails to satisfy at least one of the axioms.

  (this is probably too much to hope for).

- We would like to have a list of *simple properties*

  so that major clustering methods are distinguishable from each other using these properties.

High-level Open Questions

- What do we require from a set of clustering axioms? (Meta axiomatization …)
- How can the "completeness" of a set of axioms be defined/argued?
- Are there general, non-trivial, clustering properties that the axioms should prove?.
-

Part 2

Given a clustering task,

How should a suitable clustering paradigm be chosen?

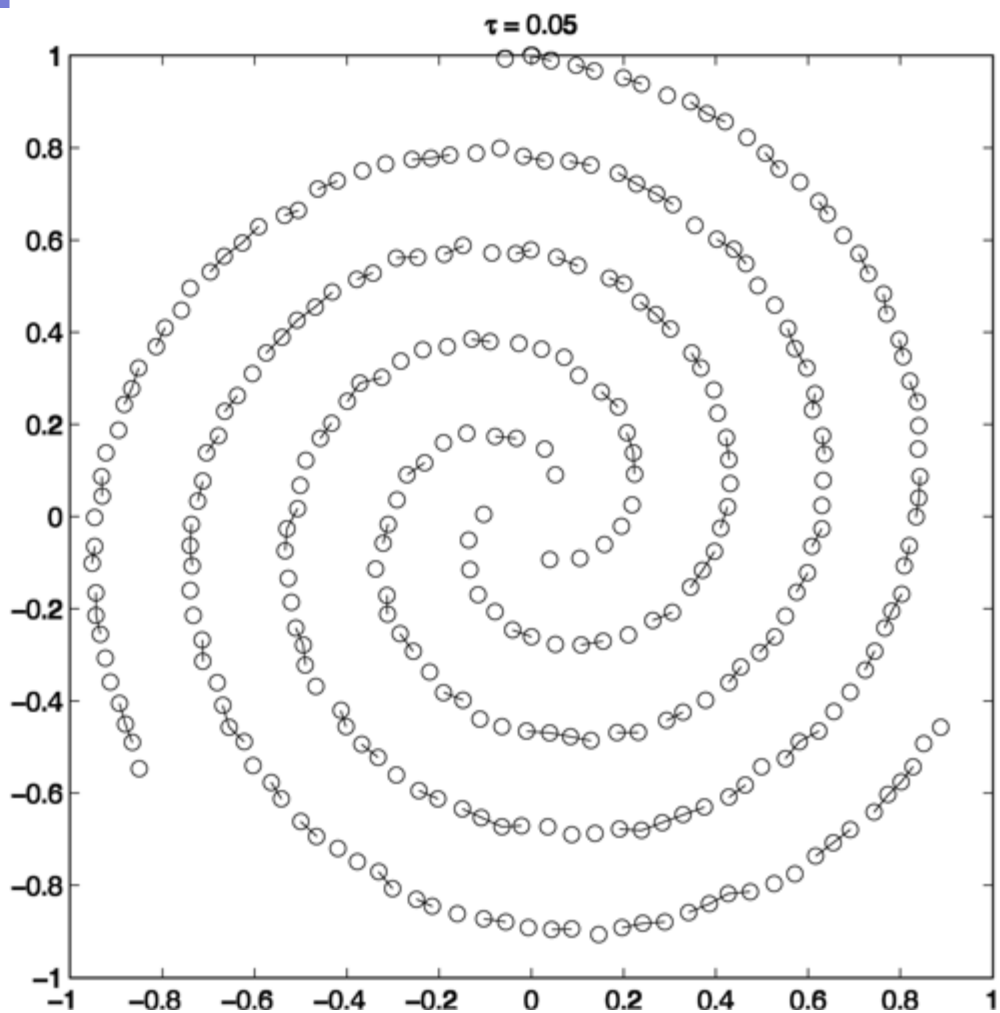
Examples of some popular clustering paradigms – *Linkage Clustering*

- Given a set of points and distances between them, we extend the distance function to apply to any pair of domain subsets. Then the clustering algorithm proceeds in stages.

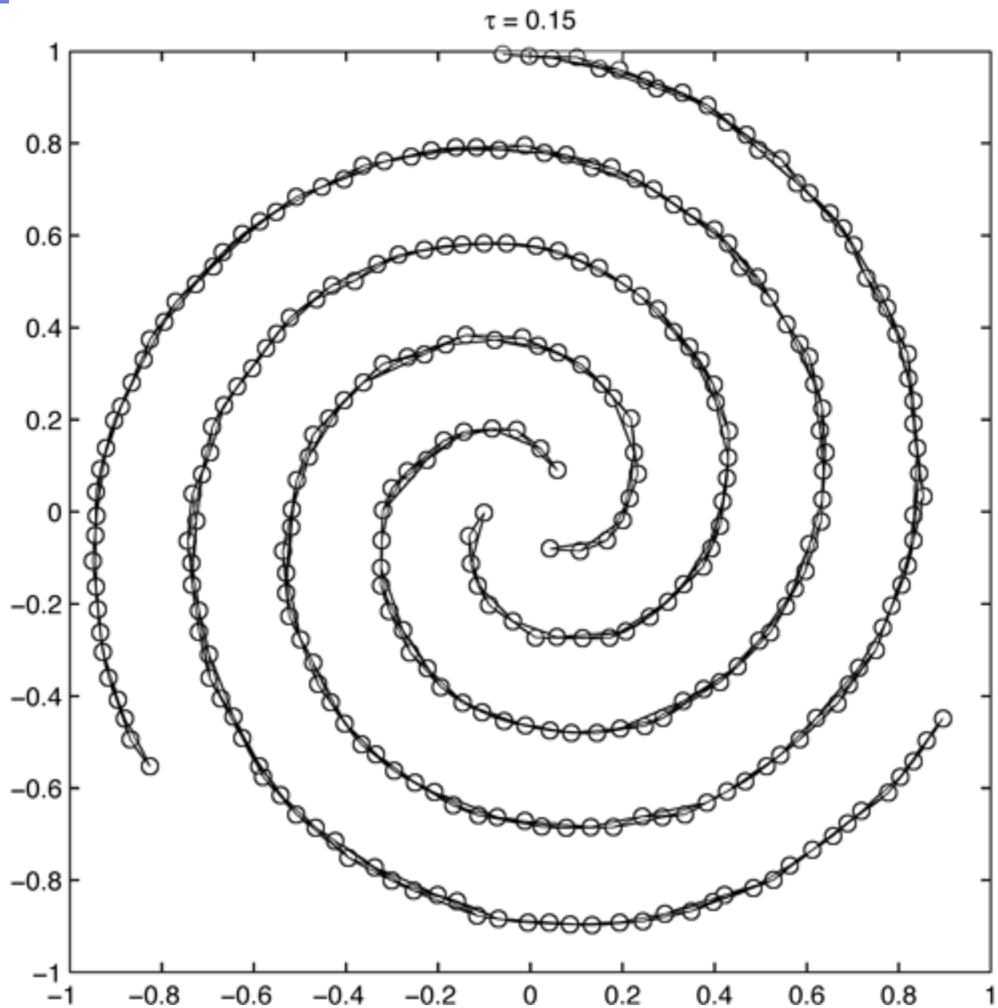
- In each stage the two clusters that have the minimal distance between them are merged.

- The user has to set the stopping criteria – when should the merging stop.

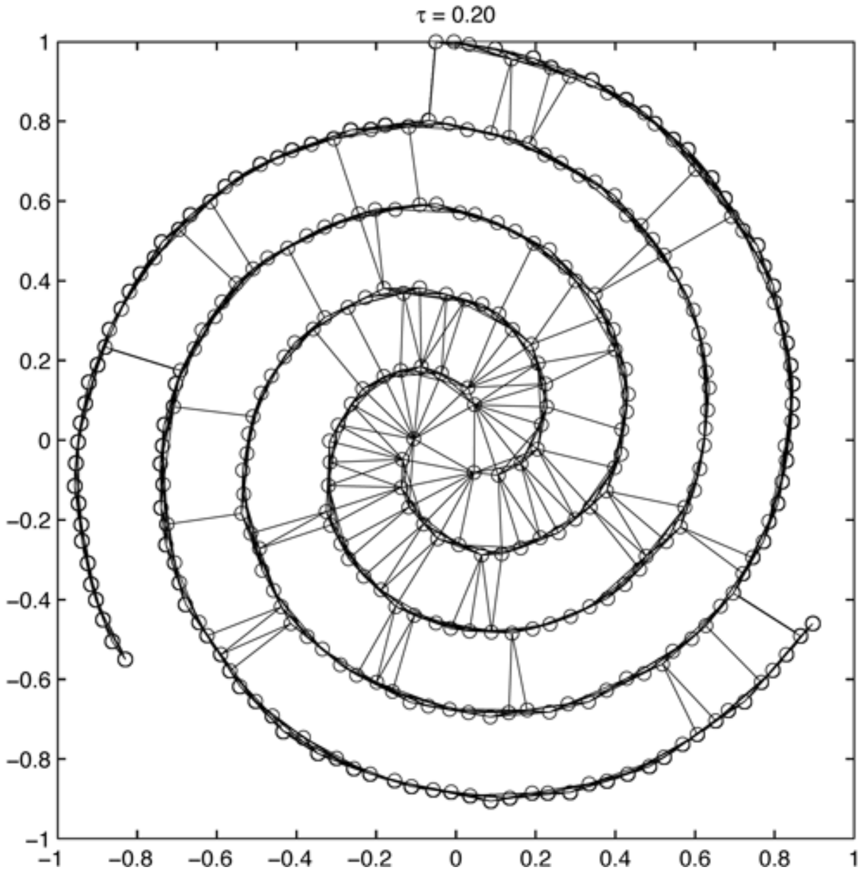
Single Linkage Clustering- early stopping



Single Linkage Clustering – “correct” stopping



Single Linkage Clustering – late stopping



Examples of popular clustering paradigms – *Center-Based Clustering*

The algorithm picks k “center points” and the clusters are defined by assigning each domain point to the center closest to it.

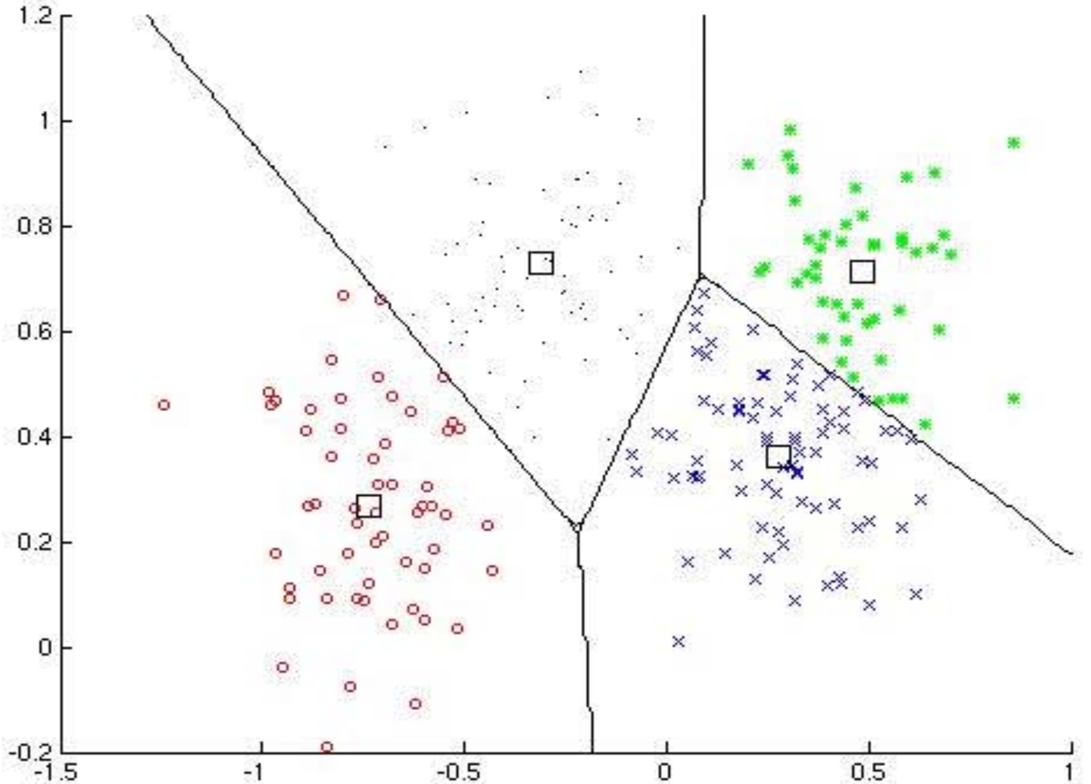
The algorithm aims to minimize some cost function that reflects how “compact” the resulting clusters are.

Center-based algorithm differ by their choice of the cost function (k-means, sum of distances, k-median and more)

The number of clusters, k, is picked by the user.

4-Means clustering example

kmeans_example.gif (GIF Image, 560x421 pixels)   http://cmp.felk.cvut.cz/cmp/software/stprtool/examples/kmeans_example.gif



Some common clustering paradigms

- Cost-driven clustering
- Algorithm-Based clustering
- Generative-Model based clustering

Families of clustering paradigms

2) *Clustering based on Objective Functions (cost driven)*– we define a cost of a clustering and, given a data set, search for a clustering that minimizes the cost for it.

2.1) Center based objectives:

2.1.1 The K-Means objective.

2.1.2 The K-Median objective

2.2) Sum of In-cluster Distances objective.

2.3) Max – Cut objectives.

2.4) Minimize within-cluster-variance/between-cluster-variance.

Families of clustering paradigms

1) *Algorithmically defined:*

1.1) Agglomerative Clustering (Linkage-based): -- iteratively join "closest" clusters.

1.1.1Single Linkage

1.1.2 Average Linkage

1.1.3 Max Linkage

1.2) Model Based algorithms (EM):

1.2.1The K-means algorithm

1.3) Spectral clustering (linear algebra-based algorithms).

Guidelines for choosing a clustering paradigm

With this large verity of different clustering tools (often resulting in very different outcomes), how do users actually pick a tool for their data?

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

By analogy….

Assume I get sick now in Beijing and do not have access to a doctor. I walk into a pharmacy in search for suitable medicine.

However, I can't read Chinese, so what do I do?

I pick a drug based on the colors of its package and its cost….

Quite similarly, 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, though …)", "runs fast" etc.

Guidelines for choosing a clustering paradigm

Challenge: formulate properties of clustering functions that would allow translating prior knowledge about a clustering task into guidance concerning the choice of suitable clustering functions.

Axioms to guide a 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.

"Axioms" *"Properties"*

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

Properties defining clustering paradigms

Next, I will introduce some example high-level properties of clustering functions, and show how they can guide the choice of clustering tools.

Other properties of clust. functions

Order Consistency:

Let d, d' be two dissimilarity measures over the same domain set X.

We say that d and d' are *order compatible* if for every s,t,u,v in X, $d(s,t) < d(u,v)$

If and only if $d'(s,t) < d'(u,v)$.

A clustering function F is *order consistent*, if for any such d, d', $F(X,d) = F(X,d')$.

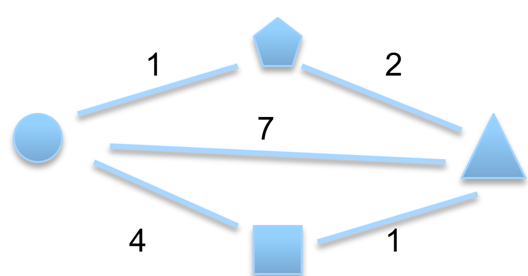
Path-Distance

Given a dissimilarity measure, d over some domain set X, we define the *d-induced path distance*, P_d, by setting, for all $x, y \in X$,

$$P_d(x,y) = \min_{q \in P_{x,y}} \max_{i<|q|} d(q(i), q(i+1))$$

In other words, we find the path from *x* to *y*, which has the smallest longest jump in it.

e.g.



Undrawn edges are large

$P_d($  ,  $) = 2$

Since the path from above has a jump of distance 2

Path Distance Coherence

A clustering function F is Path Distance Coherent

If for any ***X*** and any dissimilarity measures ***d*** and ***d'***,
If ***d*** and ***d'*** induce the same path distance over ***X***, then   ***F(X,d)=F(X,d')***

(in other words, all that the clustering cares about is the path distance induces by d)

Characterization of Single Linkage

Theorem

Single-Linkage is the *only clustering function* satisfying:

k-Richness,

Order-Consistency

and

Path Distance-Coherence

Linkage-Based clustering paradigms

- Single Linkage clustering
- Average Linkage clustering
- Complete Linkage clustering

Linkage Based clustering

- Given (X,d) define an induced dissimilarity over subsets of X, $\hat{d}$

  (it should satisfy some basic requirements)

- Let $F_0(X,d) = \{\{x\} : x \in X\}$

- Construct $F_{n+1}(X,d)$ from $F_n(X,d)$ by merging the two $\hat{d}$ – closest clusters of $SL_n(X,d)$

The requirements on subset-dissimilarity

- Isomorphism invariance
- Coherence with the underlying point-wise dissimilarity.
- “Richness”

Characterizing Linkage-Based clustering methods

- <u>The Refinement property:</u>

  For all $k' < k$, for every $C \in F(X,d, k)$ there exist $C' \in F(X,d, k')$ such that $C \subseteq C'$

- <u>The Locality property:</u>

  For every $S \subseteq F(X,d, k)$,

  $$F(US, d, |S|)=S$$

The “Extended Richness” property

- For every set of domains $\{(X_1, d_1) \ldots\ldots (X_n, d_n)\}$ there is a dissimilarity function d over $\cup_i X_i$ extending each of the d_i's such that $F(\cup_i X_i, d, n) = \{X_1, \ldots\ldots X_n\}$

Characterizing Linkage-Based clustering methods

- Theorem:

  A clustering function can be defined as a *linkage-based clustering* if and only if it satisfies the *Refinement, Extended Richness* and the *Locality* properties.

Some non-linkage paradigms

- K-means (*fails Refinement*)
- Spectral clustering (*fails Locality*)

To summarize

We have come up with characterizations (by high-level input-output properties) of several popular clustering paradigms,

e.g.,

Single Linkage clustering,

general Linkage-Based clusterings.

Other parameters that vary between clustering methods

- Drive towards number of points balance between clusters.
- Sensitivity to point weights.
- Robustness to perturbations and noise.
- Sensitivity to outliers.

Some obvious open challanges

Characterize any of the common center-based clustering paradigms.

Come up with clustering properties that reflect the consideration of users in practical settings.

Part 3: Computational complexity issues

For the last part of the talk, I wish to focus on the next stage – after a clustering paradigm has been selected.

Furthermore, assume that we have decided to apply some cost-based clustering.

An important issue is, how much computation will be needed to find a good clustering?

The computational complexity of clustering tasks:

It is well known that most of the common clustering objectives are NP-hard to optimize.

In practice, however, clustering is being routinely carried out.

Some believe that "clustering is hard only when it does not matter". Can this be formally justified?

The K-Means algorithm

For input set X in R^n , repeat for $i = 0, \ldots,$

Given centers $c^i_1, \ldots c^i_k$, for $l = 0 \ldots,$ do:

For each $l \leq k$

$C^i_j = \{x : d(x, c^i_j) < d(x, c^i_l)$ for all $l \neq j\}$

C^{i+1}_l = the center of C^i_j

More about the K-Means Alg

- Choice of initial centers $c^0_1, \ldots c^0_k$

  Makes a difference – often chosen uniformly at random over X.

- *Poor performance guarantees:*

1. May terminate in local optimum.
2. May require exponential number of rounds before terminating.

Better guarantees for clusterable inputs

- Define 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 then ε^2 times the cost of the optimal

  (k − 1)-clustering of (X, d).
- Ostrovski et al (2007) show that

  for small ε this implies that K-means reaches optimal solution fast (when initial centers are carefully picked)

How realistic is that condition?

- For the Ostrovski et al condition to imply fast optimal clustering, at least two of the k clusters should be at least 60 times their diameter away from each other ….

Other notions of Clusterable Data

Perturbation Robustness: An input data set is *perturbation robust* if small perturbations its points do not result in a change of the optimal clustering for that set.

An input set (X, d) is ε-Additive PR

if for some optimal k-clustering C,

for every d',

if $|d(x, y) - d'(x, y)| \leq \varepsilon$ for every $x, y \in X$,

then C is also optimal for (X, d').

Additive PR makes clustering easier

- Ackerman and BD (2009) show that for every center-based clustering objective and every μ > 0 there exists an algorithm that runs in time $O(m^{k/\mu^2})$ and finds the optimal clustering for every instance that is μ-APR. Using the results of BD (2007) the parameter m in the runtime can be replaced by $(dk/\mu^2 \varepsilon^2)$ if one

  settles for a solution whose cost is at most

  $\varepsilon|X|D(X)$ above that of an optimal clustering,

Some concerns

While this run time is polynomial in the size of the input for any fixed k and μ

Its gets formidably high for large number of clusters, k.

Multiplicative Perturbation robustness

An input set (X, d) is c-Multiplicative PR

if

for some optimal k-clustering C,

for every d′, if

$1/c \le d(x, y)/d'(x, y) \le c$

for every $x, y \in X$,

then C is also optimal for (X, d').

Other investigated Notions of clusterability

Several other notions of “clusterability” have been suggested and shown to make clustering computationally easier.

- α-center stability: Awasthi et al. (2012) define an instance (X,d) to be α-center *stable* if for any optimal clustering C, points are closer to their own cluster center by a factor α more than to any other cluster center.

More clusterability conditions

- *Uniqueness of optimum:* Balcan et al. (2008)
- *(1 + α) Weak Deletion Stability:* Awasthi et al. (2010)

“Conditional” feasibility of clustering

Under each of these notions, there exist clustering algorithms that, when the data

Is sufficiently clusterable, find optimal clusterings in polynomial time (in both the input size and the number of clusters, k).

The key technical component

All of those results go through a notion of "α center robustness". Namely, in an optimal clustering of the given input data, every point is closer to its own center by factor of α more than to any other center.

However, [Reyzin Ben-David] show that for $\alpha < 2$ center-based clustering is still NP-hard.

In conclusion

Although many believe that “clustering is hard only when it does not matter”,

we do have convincing theoretical support to this claim.

All the current results suffer from either requiring unrealistically high running time, or assuming inputs are unrealistically nice.

Another issue with existing results

The currently proposed notions of clusterability refer to the optimal solution, and cannot be computed efficiently from the input data.

Open questions

Do there exist notions of clusterability that are:

- Reasonable to assume for naturally arising data.
- Imply efficiency of clustering.
- Can be tested efficiently from the input data.

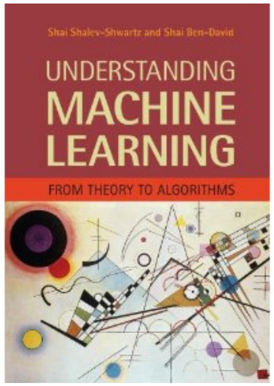
??

Summary

Clustering raises many challenges that are both practically important and theoretically approachable.

I addressed three directions: Defining clustering, Devising guidance for algorithm selection, and Understanding the computational complexity of clustering in practice.

Just out:



See all 1 image(s)

Publisher: learn how customers can search inside this book.

Tell the Publisher!
I'd like to read this book on Kindle

Don't have a Kindle? Get your Kindle here, or download a **FREE** Kindle Reading App.

Understanding Machine Learning: From Theory to Algorithms [Hardcover]
Shai Shalev-Shwartz (Author), Shai Ben-David (Author)

List Price: ~~CDN$ 62.95~~
Price: CDN$ 50.36 Prime
You Save: CDN$ 12.59 (20%)
Pre-order Price Guarantee. Learn more.

This title has not yet been released.
You may pre-order it now and we will deliver it to you when it arrives.
Ships from and sold by **Amazon.ca**. Gift-wrap available.

Save Up to 90% on Textbooks
Hit the books in Amazon.ca's Textbook Store and save up to 90% on used textbooks and 35% on new textbooks. Learn more.

Vous voulez voir cette page en français ? Cliquez ici.

Quantity: 1

or
Sign in to turn on 1-Click ordering.

Share

Book Description

Publication Date: **May 31 2014** | ISBN-10: **1107057132** | ISBN-13: **978-1107057135**

Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this textbook is to introduce machine learning, and the algorithmic paradigms it offers, in a principled way. The book provides an extensive theoretical account of the fundamental ideas underlying machine learning and the mathematical derivations that transform these principles into practical algorithms. Following a presentation of the basics of the field, the book covers a wide array of central topics that have not been addressed by previous textbooks. These include a discussion of the computational complexity of learning and the concepts of convexity and stability; important algorithmic paradigms including stochastic gradient descent, neural networks, and structured output learning; and emerging theoretical concepts such as the PAC-Bayes approach and compression-based bounds. Designed for an advanced undergraduate or beginning graduate course, the text makes the fundamentals and algorithms of machine learning accessible to students and non-expert readers in statistics, computer science, mathematics, and engineering.