Clustering is easy when ....

What?

Shai Ben-David

School of Computer Science

University of Waterloo, Canada

NIPS 2015 Learning Faster Easy Data Workshop

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*

Our basic rationale

For many practical computation tasks, *"naturally arising" inputs* are considerably easier than the worst-case instances.

Example: *SAT solvers run efficiently in practice*

Major challenge: How can such inputs be defined?

Basic questions

1) *Are there properties that distinguish naturally occurring inputs?*
2) *May such inputs be easy to solve?*

3) *Can such inputs be formally distinguished for some specific tasks?*

4) *Can there be a generic definition that holds across many computational tasks?*

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 the talk

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 is reasonable to assume that most (or at least a significant proportion) of the inputs one may care about in practice are "clusterable".

- 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.
- Maybe checked against common generative models.

Desiderata for notion of “Clusterable” inputs

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

(we will have to be more specific about the meaning of “efficient”. In particular, the dependence on number of clusters)

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)?

How does our "Additive Perturbation Robustness" fare?

1. I believe that the APR meets the "realistic inputs" requirement – small feature-measurements inaccuracies should not have dramatic effect on solutions.
2. The resulting algorithm is polytime but not practically efficient.
3. We have no way of testing inputs.
4. The algorithm is not practically common

Recently proposed clusterability notions

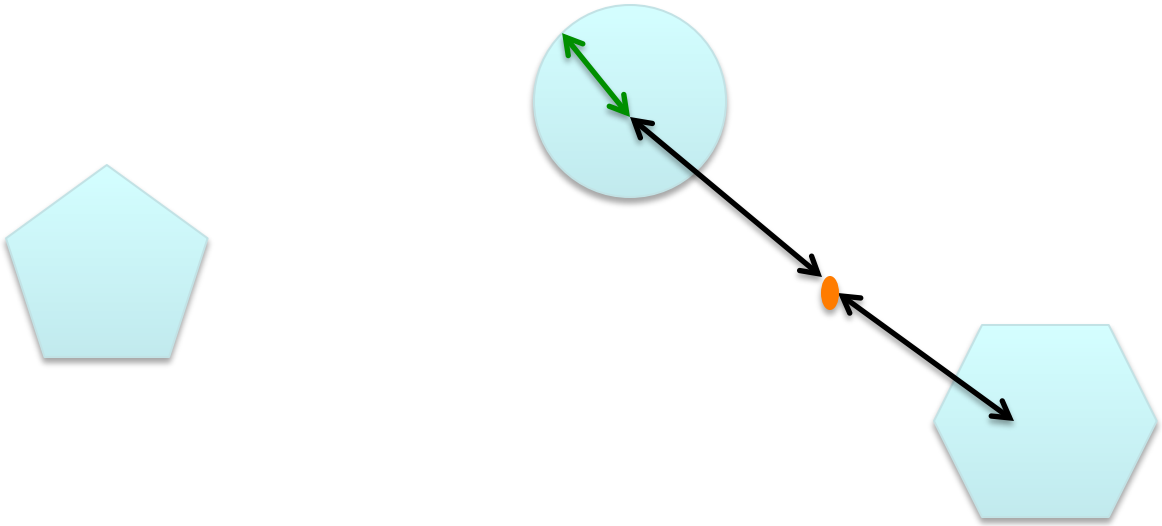
1. ***Perturbation Robustness(PR)*** – data set I is robust if small perturbations of I do not result in changes to its optimal clustering.

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

1b. ***Multiplicative PR*** *[Bilu-Lineal 2010]* - the perturbation may change every pairwise point distance my 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 then 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 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 $c_1, \ldots c_k$, 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 that 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 α>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 really the “realistic” inputs.

The current approach to define input niceness that will render efficiency as a function of the number of clusters, k, seems to be inherently too restrictive.

Alternative directions (1)

- All the current approaches that try to tackle the exponential dependence on the number of clusters are based on formalizing large between-clusters separation.

- It seems that substantiating the CDNM thesis will require a different approach.

A different type of “clusterability”

There is a lower bound D*(X), on the k-means cost that can be efficiently computed by a spectral method.

M. Meila (2006) showed that if the cost of an optimal clustering for an input X is close enough to D*(X), then one can efficiently check whether a given clustering is close to optimal.

Easy clustering under the Meila condition.

- **[BD, unpublished]**: Under that condition, k-means can be efficiently approximated.

Intuitively, the condition states that:

The volume spanned by the optimal cluster centers is much larger than the k-means cost.

Alternative directions (2)

- All the current approaches that try to tackle the hardness of finding a minimal cost clustering.

- Is that what is really required in practice?

Alternative directions (3)

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

Consider the common task of clustering for record de-duplication in data repositories. Clustering is a common tool.

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

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?

Our basic definition – ρ-robust inputs

We say that an input, I, is ρ- robust for a problem P, if, for every input

I’ such that $d(I, I') \leq \rho$,

$P(I)=P(I')$.

(where d is some metric over the space of inputs for the problem P).

Our basic results

We will show that, for many NP-hard problems in machine learning and computational geometry, there exists a polynomial time algorithm that finds the optimal solution for any robust input.

(However, the complexity of these algorithms has exponential dependency on the robustness parameter, ρ).

First set of results
([BD-Simon 2000], [BD-2004])

For each of the above classes, for every $\rho>0$, there exists a polynomial time algorithm that finds the optimal solution for all ρ-robust inputs.

(However, the complexity of these algorithms has exponential dependence on the robustness parameter, ρ).

Algorithm's quality of approximation

Background: Combinatorial Optimization problems are defined via an "cost *function*", assigning a real number $\pi(I,T)$, for every input I and a solution T for it.

An algorithm, A, is an ε-approximation if, for all input, I,

$$\pi(I, A(I)) \leq \pi(I, Opt(I))(1+ \varepsilon)$$

Towards a new notion of approximation

For an input instance, I, and a cost function π,

define,

$\lambda_\rho(I) =$

$\mathrm{Sup}_{\{I' : d(I, I') < \rho\}} \{|\pi(I, \mathrm{Opt}(I)) - \pi(I', \mathrm{Opt}(I))| \,/\, \pi(I, \mathrm{Opt}(I))\}$

This parameter models how sensitive is the input I to small perturbations (w.r.t. the given optimization problem)

New measure of approximation quality

An algorithm is a ρ approximation if for every input I

$$\pi(I, A(I)) \leq \pi(I, Opt(I))(1+ \lambda_\rho (I))$$

Note that if the instance I is ρ-robust, the algorithm A is required to output a (fully) optimal solution.

Recall that the common notion of approximation requires

$$\pi(I, A(I)) \leq \pi(I, Opt(I))(1+ \varepsilon)$$

Basic Property of the new measure

- For inputs that have highly irregular behavior of the cost, *the new measure is less demanding*.

- For tamed/robust/regular inputs,
  (I.e. inputs, *I*, for which utility function is smooth in the neighborhood of optimal solutions for an input *I*),
  success under the new measure implies success under the usual measure

Hopefully real-life inputs are like that.

The main approximation results

Theorem 1: For each of the problems mentioned above (*BSH, BSHH, DOB, k-means*),

For every ρ, there exist an efficient (poly-time) ρ-approximation algorithm.

More precisely, the algorithm is polynomial in the input size, n and the Euclidean dimension, but exponential in $1/\rho^2$

Theorem 2: Unless P=NP, there exist no

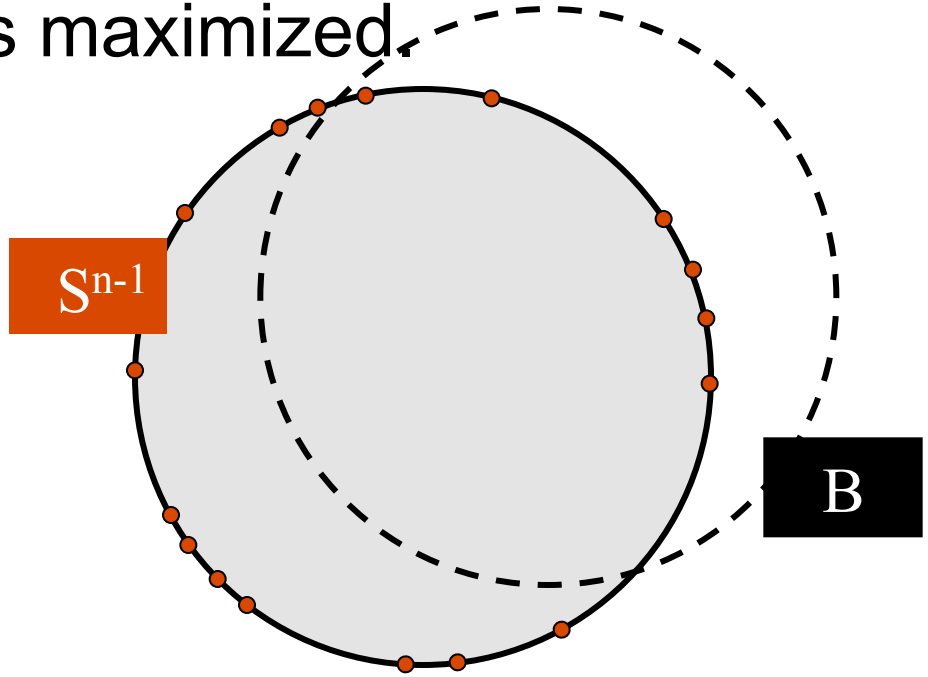
ρ- approximation scheme that runs in time

polynomial in $1/\rho$

The Densest Open Ball Problem

Input: A finite set P of points on the unit sphere S^{d-1}.

Output: An open Ball B of radius 1 so that $|B \cap P|$ is maximized.



Algorithms for the Densest Open Ball Problem

Alg. 1. ■ For every $x_1, \ldots x_{d+1} \in P$,

- find the center of their minimal enclosing Ball, $Z(x_1, \ldots, x_{d+1})$
- Check $|B[Z(x_1, \ldots, x_{d+1}), 1] \cap P|$

■ Output the ball with maximum intersection with P

Running time: $\sim|P|^{d+1}$ exponential in d.

Another Algorithm (for the Densest Open Ball Problem)

Fix a parameter $k << d$,

Alg. 2. Apply Alg. 1 only for subsets of size $\leq k$, i.e.,

- **For every $x_1, \ldots x_k \in P$,**
  - **find the center of their minimal enclosing Ball, $Z(x_1, \ldots, x_k)$**
  - **Check $|B[Z(x_1, \ldots, x_k), 1] \cap P|$**
- **Output the ball with maximum intersection with P**

Running time: $\sim |P|^k$

But, does it output a good hypothesis?

Our Core Geometric Result

The following result shows that computations from local data (k-size subsets) can approximate global computations, with precision guarantee depending only on the local parameter, k.

Theorem (folklore?): For every $k < n$, d, subset S of the d-dimensional unit ball, and every y in the convex hull of S, there exist a subset, $x_{1,\ldots}x_k$

So that

$$\|y - Zx_1 \ldots x_k\| \leq \frac{1}{\sqrt{k}}$$

The resulting approximation complexity

Theorem: For each of the problems mentioned above (BSH, BSHH, DOB, k-means),
for every $\rho>0$, there exist a ρ-approximation algorithm that runs in time $O(n^{\rho^{-2}})$

(As opposed to the NP-hardness of ε_0 – approximation, for some $\varepsilon_0 >0$, for the common definition of approximation complexity).

The resulting complexity bound for learning

Theorem: For each of the problems mentioned above (BSH, BSHH, DOB, k-means),

for every $\rho>0$, there exist a learning algorithm that

achieves error at most ε above that of the ρ – margin error of the best classifier in the class in time

$$O((\varepsilon\rho)^{-2\rho^{-2}})$$

(Since $(\varepsilon\rho)^{-2}$ examples suffice for getting the required generalization error).

Proof Idea

I. Use the above argument to replace the exponential dependence on d by (exponential) dependence on ρ.

II. Apply the definition of – approximation,

to replace approximation in the space of solutions by approximation in terms of risk (the objective function).

Comparison to Smoothed Analysis

Smoothed Analysis (Spielman and Teng) addresses the same type of problem.

- They analyze *specific algorithms*, while we consider the complexity of the *problem*.

- They "smoothen" the inputs by taking averages over random *perturbations of input* points, and then take worst case over all points.

  Here we consider the worst-case bounded-average perturbation.