Clustering with Advice: A Statistical View

by

Hassan Ashtiani

A research proposal
submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
Computer Science
at
University of Waterloo

Waterloo, Ontario, Canada, 2015

Abstract

The outcome of clustering is immensely affected by the decisions made by the user of the clustering. These design choices range from the way features are extracted (or more generally, preprocessing) to the selection of the clustering algorithm itself. Regrettably, these decisions are usually made in ad hoc ways. In this research, we are looking for methods to incorporate domain knowledge into clustering systematically.

We propose a protocol in which the domain expert provides a clustering of a relatively small random subset of a data set. The learning algorithm then uses this "advice" to automatically devise a clustering that is consistent with the domain knowledge. In our research, we call this framework *Clustering with Advice* (or CLAD).

There are important questions about CLAD. First of all, we need a model that is flexible enough to capture domain knowledge. In addition, there should be a mechanism to reduce the risk of over-fitting. Therefore, the statistical and computational properties of the framework should be analyzed.

As the first step, we formulate CLAD as a representation learning problem. In ReCLAD[1], a learning algorithm searches for a data representation under which the output of a fixed clustering algorithm, say k-means, is aligned with the intended clustering. We provide a formal statistical model for analyzing the sample complexity ReCLAD.

We then introduce a notion of capacity of a class of possible representations, in the spirit of VC-dimension, showing that classes of representations that have finite such dimension can be learned with finite sample error bounds. As a special case, we provide such dimension for the class of representations induced by linear embeddings.

Despite our promising progress, a number of important issues remain to be addressed. A critical shortcoming of the current algorithm that we use for ReCLAD is its high computational complexity. This is not surprising, because even the simple k-means clustering problem is NP-hard. Therefore, it is important to propose approaches that would make the current framework more applicable for large data sets.

[1] ReCLAD stands for REpresentation learning for CLustering with ADvice

Table of Contents

Chapter 1

Introduction

Clustering can be thought as the task of automatically dividing a set of objects into "coherent" subsets. This definition is not concrete, but its vagueness allows it to serve as an umbrella term for a wide diversity of algorithmic paradigms. Clustering algorithms are being routinely applied in a huge variety of fields.

Given a dataset that needs to be clustered for some application, one can choose among a variety of different clustering algorithms, along with different preprocessing techniques, that are likely to result in dramatically different answers. It is therefore critical to incorporate prior knowledge about the data and the intended semantics of the clustering into the process of picking a clustering algorithm (or, clustering model selection). Regretfully, there seems to be no systematic tool for incorporation of domain expertise for clustering model selection, and such decisions are usually being made in embarrassingly *ad hoc* ways.

Therefore, it is important to provide such a framework for clustering. Our aim is to address this problem and the related issues from a formal statistical viewpoint.

1.1 Clustering with Advice (CLAD)

We approach the challenge by introducing a scenario in which the domain expert (i.e., the intended user of the clustering) conveys her domain knowledge by providing a clustering of a small random subset of her data set. For example, consider a big customer service center that wishes to cluster incoming requests into groups to streamline their handling. Since the data base of requests is too large to be organized manually, the service center wishes to employ a clustering program. As the clustering designer, we would then ask the service center to pick a random

sample of requests, manually cluster them, and show us the resulting grouping of that sample. The clustering tool then should use that sample clustering to pick a clustering method that, when applied to the full data set, will result in a clustering that follows the patterns demonstrated by that sample clustering.

We call this framework *Clustering with Advice* (CLAD). It can be thought as a form of "semi-supervised clustering". Although the scenario is intuitive, it has not been studied before[1]. Hence, it is useful and fruitful to investigate it thoroughly.

There are a handful of important questions that should be answered about our framework (i.e., clustering with advice). In particular:

- What kind of **models** can we use to capture domain expert's knowledge?
- What kind of **algorithmic/computational** approaches can we provide to infer/train the parameters of the proposed models?
- What sort of **theoretical guarantees** should we expect from these algorithms?
- What are the **assumptions** that we need to make to be able to show such performance guarantees?
- What are the real-world **applications** of these algorithms?
- What are **connections** between this framework and other related frameworks?

These are the main questions that we are interested to answer in our research. Other questions may certainly arise along the way. We have already made some progress showing that this line of research is fruitful.

1.2 Proposal Structure

In Chapter 2 we review the related literature and point out their connection to our research.

In Chapter 3, we will introduce the formal framework to study the problem of clustering with advice. In particular, we formulate the problem as a *representation learning* problem. In addition, we specify the PAC-type guarantee that we expect from a supervised clustering algorithm.

[1]There are other frameworks for clustering that have an element of supervision. For example, some methods use a form of *side-information*. We will point out their differences later.

In Chapter 4, we analyze the sample complexity of the proposed model. We investigate empirical risk minimizers and provide sufficient conditions to guarantee they are PAC-learners. In particular, we show that a generalized notion of *pseudo-dimension* (analogous to VC-dimension for binary classification) characterizes PAC learnability of a class of representations.

In Chapter 5, we conclude and investigate future research directions.

Chapter 2

Related Work

In this chapter, we review the relevant literature on the problem of clustering with supervision. This area of research can be conceptually categorized under the "semi-supervised learning" field. However, we need to be more specific and distinguish between the clustering and the classification tasks. This differentiation has made several authors to name the clustering task as "semi-supervised clustering" ([8, 10, 22]).

In the next section, we categorize semi-supervised clustering models in terms of the protocol used to convey supervision. Then, we will review different approaches to semi-supervised clustering.

2.1 Supervision Protocol

The most common method to convey supervision is through a set of pairwise *must-/cannot-link* constraints over the instances ([32]). These constraints are sometimes called "side-information" ([33]). In this setting it is usually assumed that the given data points lie in some metric space and the learner has access to the pairwise distances; however, this rough distance information is not enough for clustering and the supervised constraints should also be taken into account.

It should be noted that the our CLAD protocol introduced in the Introduction is related to the above model. However, in CLAD, the learner has access to the clustering of a small set of instances (rather than the pairwise information).

In some other scenarios, the supervised feedback is in the form of pairwise similarities ([21, 19]). In this case, the goal is to learn a good clustering without seeing all the pairwise similarities.

In order to reduce the amount of required supervision, usually an active setting is used where the pairwise constraints are asked by the learner gradually ([21, 19]). In a related setting, [30] considered an active framework where the learner, instead of asking about a pairwise similarity, makes a one-vs-all query (which means that the similarity of the instance with all of the other instances is requested).

Inspired by the query models in concept learning ([3]), Balcan et al. ([6]) proposed an interactive setup where in each step the learner outputs a clustering, and the teacher corrects him. This correction is either in the form of a *split* advice, or a *merge* advice. This type of supervision has the advantage of being more intuitive for the domain expert (i.e., the teacher). However, for large data sets with large number of clusters it is hard for the teacher to check the output of the learner in each step.

2.2 Semi-Supervised Clustering Methods

2.2.1 Constrained Clustering

Semi-supervised clustering with pairwise constraints is probably the oldest method to inject supervision into clustering. The common way of using such supervision is by changing the objective of clustering so that violation of these constraints is penalized ([17, 23, 11]). These methods are sometimes called "constrained clustering".

There have been several attempts to benefit from supervision for k-means clustering. Wagstaff et al. ([32]) modified the well known Lloyd's algorithm ([24]) to avoid assigning conflicting instances to the same cluster. Also, Basu et al. ([8]) used labeled data to initialize the centres for the Lloyd's algorithm.

Hierarchical (i.e., agglomerative) clustering methods have also been extended to the supervised setting. In [26], pairwise constraints were used to prune the clustering tree. Davidson and Ravi ([16]) also studied this setting, and showed some computational hardness results about the satisfiability of these constraints.

The problem with the constrained clustering is that most of the proposed methods are *ad hoc* in two ways. First, the objective of clustering is selected in an ad hoc way without a clear justification. Second, the optimization problem is usually NP-hard, and only heuristics are used to solve the problem.

2.2.2 Metric Learning

Another approach—which is more relevant to our research—keeps the clustering optimization objective fixed and instead searches for a metric that roughly fits the given constraints. In particular, the metric is learned based on some objective function over metrics ([33, 2, 28]), so that pairs of instances marked *must-link* will be close in the new metric space (and *cannot-link* pairs be considered as far apart).

Note that however, these objective functions are usually rather *ad hoc*. In particular, it is not clear in what sense they are compatible with the adopted clustering algorithm (such as k-means). This means that performing clustering in the new space does not necessarily result in a clustering consistent with the given side-information.

A systematic way to define the objective of metric learning is to use the clustering loss directly. This is actually our approach in ReCLAD, as was briefly discussed in Introduction.

Another way to address this deficiency is to combine the two optimization problems: the metric learning, and the constrained clustering. Bilenko et al. ([14]) proposed an objective function to optimize the metric and the clustering in the same time. It then uses an iterative EM-type algorithm for optimization. Also, Basu et al. ([9]) proposed a similar framework with a different objective. The drawbacks of these integrated models are similar to those of constrained clustering: unjustifiability of the used objective functions, and the high computational complexity.

2.2.3 The Merge-Split Model

In Section 2.1, we introduced the interactive clustering framework of Balcan and Blum ([6]). In this active setting, the learner outputs a clustering in each step, and the teacher corrects him by advising him to either merge two clusters or split a cluster. In the beginning, the only thing that the learner knows is that the true clustering belongs to a given set of possible clusterings (i.e., a hypothesis class). In [6, 5], the computational and information complexity of this problem was investigated, showing some upper and lower bounds (e.g., for the case of finite hypothesis classes). In [5], the scenario was extended to noisy teachers or a teacher with incomplete response.

This model can be useful for certain applications. The interactive nature of the query model is particularly interesting. Also, the framework is theoretically solid and the algorithms are accompanied with nice theoretical results.

However, there are some limitations. Most importantly, the positive results about this framework are weak. For instance, the nice upper bounds for the query complexity of this model are

only proved for special cases—mainly for finite hypothesis classes. In addition, in order to get those bounds, it is expected from the teacher to respond to "exponentially large-sized" queries, a task that is certainly exhausting for teachers.

It is however conceivable that making some modifications to this model – through changing the queries/assumptions – would make it more applicable.

2.2.4 Generative Models

Generative models are being used in different learning tasks, including semi-supervised clustering. In these models, it is assumed that the instances (together with their true assigned partitions) are generated from a distribution. The task is then to find the true distribution based on some observations. In order to make this possible, one needs to make strong assumptions about the distribution. The common approach is to consider a parametric distribution (e.g., mixture of Gaussian) and try to estimate its parameters.

Basu et al. ([10]) considered a generative model based on Hidden Markov Random Fields (HMRFs). They showed that this model can be regarded as a probabilistic interpretation of [9], where the Euclidean distortion is generalized to Bregman's divergence. It was then showed ([22]) that this in turn is a special case of weighted kernel k-means ([18]).

In a recent paper, Gopal and Yang ([20]) proposed an approach in which it is assumed that the data is generated by a mixture model (Gaussian or Von-Mises Fisher). The parameters of this model is then found such that the probability of generating the supervised labels is maximized.

These models are useful when we have solid information about the data-generating distribution. However, in practice, the data is almost never truly generated from a mixture model. Therefore, at least an agnostic guarantee is needed to make sure that the outcome of the algorithm is the best possible model within the considered class. Second, the computational complexity is high and usually just a local minimum is found. Third, the maximum likelihood estimate is not a good objective, because it doesn't necessarily capture the true objective (i.e., the clustering loss). These drawbacks make the use of these methods limited.

2.2.5 Property-based Clustering

A totally different approach to the problem of communicating user expertise for the purpose of choosing a clustering tool is discussed in [1]. They considered a set of *properties* (or *requirements*) for clustering algorithms, and investigated which of those properties hold for various

algorithms. The user can then pick the right algorithm based on the requirements that she wants the algorithm to meet.

However, to turn such an approach into a practically useful tool, one will need to come up with properties that are relevant to the end user of clustering –a goal that is still far from being reached. Also, these properties are more useful for picking the general clustering paradigm (e.g., agglomerative or center-based), rather than picking the specific parameters (e.g, the target metric).

2.3 Conclusions

As it was discussed in the previous section, most of the research done on semi-supervised clustering is devoted to the algorithmic side of the problem[1]: each proposed method exploits the supervision in an optimization problem. However, the choice of objective function is rarely justified. In addition, the optimization problem is usually still hard, and the proposed algorithms find only a local optimum. In the following, we review the main drawbacks of different approaches.

- **Why does optimizing the proposed objective capture our expectations from the task?** In practice, for constrained clustering and metric learning methods (or their combination) the objective function is selected in an ad hoc way. In addition, the maximum likelihood approach of generative models is not well justified.
- **Are the proposed methods computationally efficient?** Most of the constrained clustering objectives are *hard* to optimize (or approximate). This is usually the case for generative models as well. For the merge-split model also clustering can be hard.
- **Are the sssumptions realistic?** In generative models, it is assumed that the data is generated by a specific parametric distribution. Unfortunately, there is no guarantee about the outcome of these methods when the true distribution fails to match the expectations.
- **Are the proposed methods information-efficient?** or does the proposed method use the supervised data efficiently? Aside form some exceptions (including the merge-split model) no upper or lower bounds for the information/statistical complexity of the task is obtained. Even for the merge-split model, these results are not generally promising. In addition, contrary to supervised learning, proving generalization bounds for semi-supervised clustering is very rare.

[1] This is actually the case for the whole clustering literature

Therefore, there is still much room for research in semi-supervised clustering. In particular, there is a need for a theoretically solid work in this area, beyond just another heuristic algorithm.

Chapter 3

Representation Learning for CLAD

In this chapter, we propose a model to capture domain knowledge in the Clustering with Advice (CLAD) framework. The idea is to "train" a model based on the given supervised data in CLAD framework (i.e, the small random subset of data set that is clustered by the domain expert).

The model used for this purpose should be flexible enough to be able to capture the domain expert's knowledge. In addition, aiming to achieve generalization guarantees for such an approach, it is essential to introduce some *inductive bias* to avoid over-flexibility (and consequently over-fitting). We can do this by restricting the clustering model to be from a predetermined hypothesis class (or a set of concrete clustering algorithms). In the next section, we approach the problem from a representation learning perspective.

3.1 Representation Learning for CLAD (ReCLAD)

In this section, we propose a representation learning approach to the CLAD framework. We call this approach *ReCLAD* which stands for *REpresentation learning for CLustering with ADvice*.

Assume that the expert-desirable clustering can be approximated by first embedding the sample into some Euclidean (or Hilbert) space, and then performing a fixed clustering algorithm (e.g., k-means clustering) in the new space. In this case, by approximating the implicit embedding, one can recover the clustering that is consistent with domain knowledge.

To be concrete, we first formulate clustering with advice as a *representation learning* problem. We do that by fixing a clustering algorithm, say k-means clustering, and searching for an embedding of the data under which k-means respects the given sample clustering. Note that

any embedding corresponds to a "kernel"; therefore, we can alternatively view this as a kernel learning problem.

The choice of k-means in the above is however, rather arbitrary. In the next section, we formulate the ReCLAD problem for a general unsupervised clustering algorithm. However, as we will see, the case of k-means is especially interesting and will be studied separately.

Our contribution in this chapter is to present ReCLAD model for the problem of clustering with advice. Furthermore, a precise formulation of ReCLAD is provided where an appropriate loss function is introduced to quantify the success of the learning algorithm. Then, we define the analogous notion of PAC-learnability[1] for the problem of learning representation for clustering.

In the following, we provide the notations needed to introduce the PAC-ReCLAD framework.

3.2 Preliminaries and Notations

Let X be a finite domain set. A *k-clustering* of X is a partition of X into k subsets. If C is a k-clustering, we denote the subsets of the partition by $C_1, ..., C_k$, therefore we have $C = \{C_1, .., C_k\}$. Let π^k denote the set of all permutations over $[k]$ where $[k]$ denotes $\{1, 2, ..., k\}$. The clustering difference between two clusterings, C^1 and C^2, with respect to X is defined by

$$\Delta_X(C^1, C^2) = \min_{\sigma \in \pi^k} \frac{1}{|X|} \sum_{i=1}^{k} |C_i^1 \Delta C_{\sigma(i)}^2| \tag{3.1}$$

where $|.|$ and Δ denote the cardinality and the symmetric difference of sets respectively. For a sample $S \subset X$, and C^1 (a partition of X), we define $C^1\Big|_S$ to be a partition of S induced by C^1, namely $C^1\Big|_S = \{C_1^1 \cap S, \ldots, C_k^1 \cap S\}$. Accordingly, the sample-based difference between two partitions is defined by

$$\Delta_S(C^1, C^2) = \Delta_S(C^1\Big|_S, C^2\Big|_S) \tag{3.2}$$

Fix an unsupervised clustering algorithm, e.g., k-means clustering, that given a data set, outputs a k-partition of the data. We denote C_X as the outcome of clustering X (i.e., it is a k-clustering of X). Note that the unsupervised clustering algorithm is fixed and should be clear from the context.

[1] PAC stands for the well known notion of "probably approximately correct", popularized by [29].

Let f be a mapping from X to $\mathbb{R}^d$. We define C_X^f the result of clustering X after mapping it to a new space using f. In other words, $C_X^f = C_{f(X)}$.

The difference between two mappings f_1 and f_2 with respect to X is defined by the difference between the result of clustering using these mappings. Formally,

$$\Delta_X(f_1, f_2) = \Delta_X(C_X^{f_1}, C_X^{f_2}) \tag{3.3}$$

3.3 Formal Problem Statement (PAC-ReCLAD)

Let C^* be the target k-clustering of X. A *representation learning algorithm* $A(.,.)$ takes as input a sample set $S \subset X$ and its clustering, $C^*\Big|_S$, and outputs a mapping f from a set of mappings $\mathcal{F}$.

We call this learning problem ReCLAD which stands for *REpresentation learner for CLustering with ADvice*.

Definition 1. *Probably Approximately Correct Representation Learning for Clustering with Advice (PAC-ReCLAD)*

Let $\mathcal{F}$ be a set of mappings from X to $\mathbb{R}^d$. A representation learning algorithm A is a PAC-ReCLAD learner with sample complexity $m_{\mathcal{F}} : (0,1)^2 \mapsto \mathbb{N}$ with respect to $\mathcal{F}$, if for every $(\epsilon, \delta) \in (0,1)^2$, every domain set X and every clustering of X, C^*, the following holds:

if S is a randomly (uniformly) selected subset of X of size at least $m_{\mathcal{F}}(\epsilon, \delta)$, then with probability at least $1 - \delta$

$$\Delta_X(C^*, C_X^{f_A}) \leq \inf_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f) + \epsilon \tag{3.4}$$

where $f_A = A(S, C^*\Big|_S)$, is the output of the algorithm.

Note 1. *In this definition, f_A is the mapping that the algorithm outputs. Using this mapping, X is mapped to a new space. The result of clustering in this new space is $C_X^{f_A}$. Therefore, it is assumed that a fixed unsupervised clustering is used to cluster the data in the new space. In the next section, we will fix the k-means clustering for this purpose.*

Note 2. *This can be regarded as a formal PAC framework to analyze the problem of clustering with advice. The learner is compared to the best mapping in the class $\mathcal{F}$. This means that this is an* agnostic *framework.*

Note 3. *In this proposal, we investigate the* transductive *setup, where there is a given data set, known to the learner, that needs to be clustered. Clustering often occurs as a task over some data generating distribution (e.g., [31]). The current work can be readily extended to that setting. However, in that case, we assume that the clustering algorithm gets, on top of the clustered sample, a large unclustered sample drawn form that data generating distribution.*

A natural question is providing bounds on the sample complexity of PAC-ReCLAD with respect to $\mathcal{F}$. Intuitively, for richer classes of mappings, we need larger clustered samples. Therefore, we need to introduce an appropriate notion of "capacity" for $\mathcal{F}$ and bound the sample complexity based on it. This is addressed in the next chapter.

In the next section, we specialize the general framework of ReCLAD for the case of k-means clustering.

3.4 The Case of K-means Clustering (Re-KLAD)

In the previous section it was stated that the ReCLAD method relies on an unsupervised clustering method. In this section, we fix the k-means clustering algorithm in the ReCLAD framework. It means that we are looking for a representation of data under which the result of k-means clustering is consistent with the domain knowledge. We call this approach ReKLAD (which stands for REpresentation Learning for K-means clustering with ADvice).

k-means belongs to the class of center-based clustering methods. In these algorithms, the goal is to find a set of "centers" (or prototypes), and the clusters are the Voronoi cells induced by this set of centers. The objective of such a clustering is to minimize the expected value of some monotonically increasing function of the distances of points to their cluster centers. The k-means clustering objective is arguably the most popular paradigm in this class. Currently, center-based clustering tools lack a vehicle for incorporating domain expertise. Domain knowledge is usually taken into account only through an ad hoc choice of input data representation. Regretfully, it might not be realistic to require the domain expert to translate sufficiently elaborate task-relevant knowledge into hand-crafted features. This makes the study of ReKLAD interesting and important.

Also, k-means is especially interesting because it is flexible: for *any* target clustering in any domain, there exists a corresponding embedding to a new space such that the solution of k-means in the new space is the same as target clustering[2].

[2]This property is sometimes called k-Richness

As a result, we formulate the ReCLAD problem for the case of k-means clustering. In the following, we introduce the formal definitions.

3.4.1 Definitions and Notations

Let f be a mapping from X to $\mathbb{R}^d$, and $\mu = (\mu_1, \dots \mu_k)$ be a vector of k centers in $\mathbb{R}^d$. The clustering defined by (f, μ) is the partition over X induced by the μ-Voronoi partition in $\mathbb{R}^d$. Namely,

$$C_f(\mu) = (C_1, \dots C_k), \text{ where for all } i,$$
$$C_i = \{x \in X : \|f(x) - \mu_i\|_2 \leq \|f(x) - \mu_j\|_2 \text{ for all } j \neq i\}$$

The k-means cost of clustering X with a set of centers $\mu = \{\mu_1, \dots, \mu_k\}$ and with respect to a mapping f is defined by

$$COST_X(f, \mu) = \frac{1}{|X|} \sum_{x \in X} \min_{\mu_i \in \mu} \|f(x) - \mu_i\|_2^2 \tag{3.5}$$

The k-means clustering algorithm finds the set of centers μ_X^f that minimize this cost[3]. In other words,

$$\mu_X^f = \arg\min_{\mu} COST_X(f, \mu) \tag{3.6}$$

Also, for a partition C and mapping f, we can define the cost of clustering as follows.

$$COST_X(f, C) = \frac{1}{|X|} \sum_{i \in [k]} \min_{\mu_j} \sum_{x \in C_i} \|f(x) - \mu_j\|_2^2 \tag{3.7}$$

The following proposition shows the "k-richness" property of k-means objective.

Proposition 1. *Let X be a domain set. For every k-clustering of X, C, and every $d \in \mathbb{N}^+$, there exist a mapping $g : X \mapsto \mathbb{R}^d$ such that $C_X^g = C$.*

[3]We assume that the solution to k-means clustering is unique. We will elaborate about this issue in the next sections.

Proof. The mapping g can be picked such that it collapses each cluster C_i into a single point in $\mathbb{R}^n$ (and so the image of X under mapping g will be just k single points in $\mathbb{R}^n$). The result of k-means clustering under such mapping will be C. □

For a mapping f as above, let C_X^f denote the k-means clustering of X induced by f, namely

$$C_X^f = C_f(\mu_X^f) \tag{3.8}$$

3.4.2 PAC-ReKLAD

Now that we have the needed notations, we can formally define the PAC-ReKLAD problem. However, the definition is exactly the same as that of PAC-ReCLAD (Definition 1). We only need to make the use of k-means clustering as the unsupervised tool explicit.

We avoid repeating the definition. We just note for PAC-ReKLAD is the same as PAC-ReCLAD, except that the meaning of C_X^f is more explicit: C_X^f is k-clustering induced by first mapping X to a new space using f, and then performing *k-means* clustering in the new space.

Proving a bound for the sample complexity of PAC-ReKLAD is a concrete and crucial problem. In the next chapter, we address this issue.

Chapter 4

Statistical Analysis of ReCLAD

The important question that was raised in the previous chapter was that of the sample complexity: what is the size of a sample, to be clustered by the domain expert, that suffices for finding a close-to-optimal embedding (i.e., a mapping that generalizes well on the test data)?

Intuitively, this sample complexity depends on the richness of the class of potential embeddings that the algorithm is choosing from. In standard supervised learning, there are well established notions of capacity of hypothesis classes (e.g., VC-dimension) that characterize the sample complexity of learning. In this chapter we will introduce relevant notions of capacity for ReCLAD.

Particularly, we introduce a combinatorial parameter, a specific notion of the capacity of the class of mappings, that determines the sample complexity of ReCLAD for the case of k-means clustering (i.e., ReKLAD framework) . This combinatorial notion is a multivariate version of *pseudo-dimension* of a class of real-valued mappings. We show that there is *uniform convergence* of empirical losses to the true loss, over any class of mappings, $\mathcal{F}$, at a rate that is determined by the proposed dimension.

This implies that any empirical risk minimization algorithm (ERM) will successfully learn such a class from sample sizes upper bounded by those rates.

Finally, we analyze a particular natural class –the class of linear mappings from $\mathbb{R}^{d_2}$ to $\mathbb{R}^{d_1}$– and show that roughly speaking, sample size of $O(\frac{d_1 d_2}{\epsilon^2})$ is sufficient to guarantee an ϵ-optimal answer.

4.1 Technical Background

Statistical convergence rates of sample clustering to the optimal clustering, with respect to some data generating probability distribution, play a central role in our analysis. From that perspective, most relevant to our work in this chapter are results that provide generalization bounds for k-means clustering. Ben-David [12] proposed the first dimension-independent generalization bound for k-means clustering based on compression techniques. This result was tightened in [13] through an analysis of Rademacher complexity. Also, [25] investigated a more general framework, in which generalization bounds for k-means as well as other algorithms can be obtained.

It should be noted that these results are about the standard clustering setup (without any supervised feedback), where the data representation is fixed and known to the clustering algorithm. However, analysis of the semi-supervised clustering problem –particularly PAC-ReCLAD– is still open.

4.2 ERM as a Representation Learner

In order to prove an upper bound for the sample complexity of ReCLAD, we need to consider an algorithm, and prove a sample complexity bound for it. Here, we show that any ERM-type algorithm[1] can be used for the ReKLAD framework. Therefore, we will be able to prove an upper bound for the sample complexity of PAC-ReKLAD.

Let $\mathcal{F}$ be a class of mappings and X be the domain set. A TERM[2] learner for $\mathcal{F}$ takes as input a sample $S \subset X$ and its clustering Y and outputs:

$$A^{TERM}(S, Y) = \arg\min_{f \in \mathcal{F}} \Delta_S(C^f_X\Big|_S, Y) \tag{4.1}$$

Note that we call it transductive, because it is implicitly assumed that it has access to the unlabeled dataset (i.e., X). A TERM algorithm goes over all mappings in $\mathcal{F}$ and selects the mapping which is the most consistent mapping with the given clustering: the mapping under which if we perform k-means clustering of X, the sample-based Δ-difference between the result and Y is minimized.

Intuitively, this algorithm will work well when the empirical Δ-difference and the true Δ-difference of the mappings in the class are close to each other. In this case, by minimizing the

[1] ERM stands for Empirical Risk Minimization

[2] TERM stands for Transductive Empirical Risk Minimizer

empirical difference, the algorithm will automatically minimize the true difference as well. In order to formalize this idea, we define the notion of "representativeness" of a sample.

Definition 2. *(ϵ-Representative Sample) Let $\mathcal{F}$ be a class of mappings from X to $\mathbb{R}^d$. A sample S is ϵ-representative with respect to $\mathcal{F}$, X and the clustering C^*, if for every $f \in \mathcal{F}$ the following holds*

$$|\Delta_X(C^*, C_X^f) - \Delta_S(C^*, C_X^f))| \leq \epsilon \tag{4.2}$$

The following theorem shows that for the TERM algorithm to work, it is sufficient to supply it with a representative sample.

Theorem 1. *(Sufficiency of Uniform Convergence) Let $\mathcal{F}$ be a set of mappings from X to $\mathbb{R}^d$. If S is an $\frac{\epsilon}{2}$-representative sample with respect to X, $\mathcal{F}$ and C^* then*

$$\Delta_X(C^*, C_X^{\hat{f}}) \leq \Delta_X(C^*, C_X^{f^*}) + \epsilon \tag{4.3}$$

where $f^* = \arg\min_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f)$ and $\hat{f} = A^{TERM}(S, C^*\big|_S)$.

Proof. Using $\frac{\epsilon}{2}$-representativeness of S and the fact that $\hat{f}$ is the empirical minimizer of the loss function, we have

$$\Delta_X(C^*, C_X^{\hat{f}}) \leq \Delta_S(C^*, C_X^{\hat{f}}) + \frac{\epsilon}{2} \tag{4.4}$$

$$\leq \Delta_S(C^*, C_X^{f^*}) + \frac{\epsilon}{2} \tag{4.5}$$

$$\leq \Delta_X(C^*, C_X^{f^*}) + \frac{\epsilon}{2} + \frac{\epsilon}{2} \tag{4.6}$$

$$\leq \Delta_X(C^*, C_X^{f^*}) + \epsilon \tag{4.7}$$

□

Therefore, we just need to provide an upper bound for the sample complexity of uniform convergence: "how many instances do we need to make sure that with high probability our sample is ϵ-representative?"

4.3 Classes of Mappings with a Uniqueness Property

In general, the solution to k-means clustering may not be unique. Therefore, the learner may end up with finding a mapping that corresponds to multiple different clusterings. This is not desirable, because in this case, the output of the learner will not be interpretable. Therefore, it is reasonable to choose the class of potential mappings in a way that it includes only the mappings under which the solution is unique.

In order to make this idea concrete, we need to define an appropriate notion of uniqueness. We use a notion similar to the one introduced by [7] with a slight modification[3].

Definition 3. *((η, ϵ)-Uniqueness) We say that k-means clustering for domain X under mapping $f : \mathcal{X} \mapsto \mathbb{R}^d$ has a (η, ϵ)-unique solution, if every η-optimal solution of the k-means cost is ϵ-close to the optimal solution. Formally, the solution is (η, ϵ)-unique if for every partition P that satisfies*

$$COST_X(f, P) < COST_X(f, C_X^f) + \eta \tag{4.8}$$

would also satisfy

$$\Delta_X(C_X^f, P) < \epsilon \tag{4.9}$$

In the degenerate case where the optimal solution to k-means is not unique itself (and so C_X^f is not well-defined), we say that the solution is not (η, ϵ)-unique.

It can be noted that the definition of (η, ϵ)-uniqueness not only requires the optimal solution to k-means clustering to be unique, but also all the "near-optimal" minimizers of the k-means clustering cost should be "similar". This is a natural strengthening of the uniqueness condition, to guard against cases where there are η_0-optimizers of the cost function (for arbitrarily small η_0) with totally different solutions.

Now that we have a definition for uniqueness, we can define the set of mappings for X under which the solution is unique. We say that a class of mappings F has (η, ϵ)-*uniqueness property* with respect to X, if every mapping in F has (η, ϵ)-uniqueness property over X.

Note that given an arbitrary class of mappings F, we can find a subset of it that satisfies (η, ϵ)-uniqueness property over X. Also, as argued above, this subset is the useful subset to work with. Therefore, in the rest of this chapter, we investigate learning for classes with (η, ϵ)-uniqueness property. In the next section, we prove uniform convergence results for such classes.

[3] Our notion is additive in both parameters rather than multiplicative

4.4 Uniform Convergence Results

In Section 4.2, we defined the notion of ϵ-representative samples. Also, we proved that if a TERM algorithm is fed with such a representative sample, it will work satisfactorily. The most technical part of the proof is then about the question "how large should be the sample in order to make sure that with high probability it is actually a representative sample?"

In order to formalize this notion, let $\mathcal{F}$ be a set of mappings from a domain X to $(0,1)^n$[4]. Define the sample complexity of uniform convergence, $m_{\mathcal{F}}^{UC}(\epsilon, \delta)$, as the minimum number m such that for every fixed partition C^*, if S is a randomly (uniformly) selected subset of X with size m, then with probability at least $1-\delta$, for all $f \in \mathcal{F}$ we have

$$|\Delta_X(C^*, C_X^f) - \Delta_S(C^*, C_X^f)| \leq \epsilon \tag{4.10}$$

The technical part of this chapter is devoted to provide an upper bound for this sample complexity.

4.4.1 Preliminaries

Definition 4. *(ϵ-cover and covering number) Let $\mathcal{F}$ be a set of mappings from X to $(0,1)^n$. A subset $\hat{F} \subset \mathcal{F}$ is called an ϵ-cover for $\mathcal{F}$ with respect to the metric $d(.,.)$ if for every $f \in \mathcal{F}$ there exists $\hat{f} \in \hat{F}$ such that $d(f, \hat{f}) \leq \epsilon$. The covering number, $\mathcal{N}(\mathcal{F}, d, \epsilon)$ is the size of the smallest ϵ-cover of $\mathcal{F}$ with respect to d.*

In the above definition, we did not specify the metric d. In our analysis, we are interested in the L_1 distance with respect to X, namely:

$$d_{L_1}^X(f_1, f_2) = \frac{1}{|X|} \sum_{x \in X} \|f_1(x) - f_2(x)\|_2 \tag{4.11}$$

Note that the mappings we consider are not real-valued functions, but their output is an n-dimensional vector. This is in contrast to the usual analysis used for learning real-valued functions. If f_1 and f_2 are real-valued, then L_1 distance is defined by

[4] In the analysis, for simplicity, we will assume that the set of mappings is a function to the bounded space $(0,1)^n$ wherever needed

$$d_{L_1}^X(f_1, f_2) = \frac{1}{|X|} \sum_{x \in X} |f_1(x) - f_2(x)| \tag{4.12}$$

We will prove sample complexity bounds for our problem based on the L_1-covering number of the set of mappings. However, it will be beneficial to have a bound based on some notion of capacity, similar to VC-dimension, as well. This will help in better understanding and easier analysis of sample complexity of different classes. While VC-dimension is defined for binary valued functions, we need a similar notion for functions with outputs in $\mathbb{R}^n$. For real-valued functions, we have such notion, called pseudo-dimension ([27]).

Definition 5. *(Pseudo-Dimension) Let $\mathcal{F}$ be a set of functions from X to $\mathbb{R}$. Let $S = \{x_1, x_2, \ldots, x_m\}$ be a subset of X. Then S is pseudo-shattered by $\mathcal{F}$ if there are real numbers $r_1, r_2, \ldots, r_m$ such that for every $b \in \{0,1\}^m$, there is a function $f_b \in \mathcal{F}$ with $sgn(f_b(x_i) - r_i) = b_i$ for $i \in [m]$. Pseudo dimension of $\mathcal{F}$, called $Pdim(\mathcal{F})$, is the size of the largest shattered set.*

It can be shown (e.g., Theorem 18.4. in [4]) that for a real-valued class F, if $Pdim(F) \leq q$ then $\log \mathcal{N}(F, d_{L_1}^X, \epsilon) = \mathcal{O}(q)$ where $\mathcal{O}()$ hides logarithmic factors of $\frac{1}{\epsilon}$. In the next sections, we will generalize this notion to $\mathbb{R}^n$-valued functions.

4.4.2 Reduction to Binary Hypothesis Classes

Let $f_1, f_2 \in \mathcal{F}$ be two mappings and σ be a permutation over $[k]$. Define the binary-valued function $h_\sigma^{f_1,f_2}(.)$ as follows

$$h_\sigma^{f_1,f_2}(x) = \begin{cases} 1 & x \in \cup_{i=1}^k (C_i^{f_1} \Delta C_{\sigma(i)}^{f_2}) \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

Let $H_\sigma^{\mathcal{F}}$ be the set of all such functions with respect to $\mathcal{F}$ and σ:

$$H_\sigma^{\mathcal{F}} = \{h_\sigma^{f_1,f_2}(.) : f_1, f_2 \in \mathcal{F}\} \tag{4.14}$$

Finally, let $H^{\mathcal{F}}$ be the union of all $H_\sigma^{\mathcal{F}}$ over all choices of σ. Formally, if π is the set of all permutations over $[k]$, then

$$H^{\mathcal{F}} = \cup_{\sigma \in \pi} H_\sigma^{\mathcal{F}} \tag{4.15}$$

For a set S, and a binary function $h(.)$, let $h(S) = \frac{1}{|S|}\sum_{x\in S} h(x)$. We now show that a uniform convergence result with respect to $H^{\mathcal{F}}$ is sufficient to have uniform convergence for the Δ-difference function. Therefore, we will be able to investigate conditions for uniform convergence of $H^{\mathcal{F}}$ rather than the Δ-difference function.

Theorem 2. *Let X be a domain set, $\mathcal{F}$ be a set of mappings, and $H^{\mathcal{F}}$ be defined as above. If $S \subset X$ is such that*

$$\forall h \in H^{\mathcal{F}}, |h(S) - h(X)| \leq \epsilon \tag{4.16}$$

then S will be ϵ-representative with respect to $\mathcal{F}$, i.e., for all $f_1, f_2 \in \mathcal{F}$ we will have

$$|\Delta_X(C_X^{f_1}, C_X^{f_2}) - \Delta_S(C_X^{f_1}, C_X^{f_2})| \leq \epsilon \tag{4.17}$$

Proof.

$$|\Delta_S(C_X^{f_1}, C_X^{f_2}) - \Delta_X(C_X^{f_1}, C_X^{f_2})| \tag{4.18}$$

$$= \left| \left(\min_{\sigma} \frac{1}{|S|} \sum_{x\in S} h_{\sigma}^{f_1,f_2} \right) - \left(\min_{\sigma} \frac{1}{|X|} \sum_{x\in X} h_{\sigma}^{f_1,f_2} \right) \right| \tag{4.19}$$

$$\leq \left| \max_{\sigma} \left(\frac{1}{|S|} \sum_{x\in S} h_{\sigma}^{f_1,f_2} - \frac{1}{|X|} \sum_{x\in X} h_{\sigma}^{f_1,f_2} \right) \right| \tag{4.20}$$

$$\leq \left| \max_{\sigma} \left(h_{\sigma}^{f_1,f_2}(S) - h_{\sigma}^{f_1,f_2}(X) \right) \right| \leq \epsilon \tag{4.21}$$

□

The fact that $H^{\mathcal{F}}$ is a class of binary-valued functions enables us to provide sample complexity bounds based on VC-dimension of this class. However, providing bounds based on VC-Dim$(H^{\mathcal{F}})$ is not sufficient, in the sense that it is not convenient to work with the class $H^{\mathcal{F}}$. Instead, it will be nice if we can prove bounds directly based on the capacity of the class of mappings, $\mathcal{F}$. In the next section, we address this issue.

4.4.3 L_1-Covering Number and Uniform Convergence

The classes introduced in the previous section, $H^{\mathcal{F}}$ and $H^{\mathcal{F}}_{\sigma}$, are binary hypothesis classes. Also, we have shown that proving a uniform convergence result for $H^{\mathcal{F}}$ is sufficient for our purpose. In this section, we show that a bound on the L_1 covering number of $\mathcal{F}$ is sufficient to prove uniform convergence for $H^{\mathcal{F}}$.

In Section 4.3, we argued that we only care about the classes that have (η, ϵ)-uniqueness property. In the rest of this section, assume that $\mathcal{F}$ is a class of mappings from X to $(0,1)^n$ that satisfies (η, ϵ)-uniqueness property.

Lemma 1. *Let $f_1, f_2 \in \mathcal{F}$. If $d_{L_1}(f_1, f_2) < \frac{\eta}{12}$ then $\Delta_X(f_1, f_2) < 2\epsilon$*

We leave the proof of this lemma for the appendix, and present the next lemma.

Lemma 2. *Let $H^{\mathcal{F}}$ be defined as in the previous section. Then,*

$$\mathcal{N}(H^{\mathcal{F}}, d^X_{L_1}, 2\epsilon) \leq k! \mathcal{N}(\mathcal{F}, d^X_{L_1}, \frac{\eta}{12}) \tag{4.22}$$

Proof. Let $\hat{\mathcal{F}}$ be the $\frac{\eta}{12}$-cover corresponding to the covering number $\mathcal{N}(\mathcal{F}, d^X_{L_1}, \frac{\eta}{12})$. Based on the previous lemma, $H^{\hat{\mathcal{F}}}_{\sigma}$ is a 2ϵ-cover for $H^{\mathcal{F}}_{\sigma}$. But we have only $k!$ permutations of $[k]$, therefore, the covering number for $H^{\hat{\mathcal{F}}}$ is at most $k!$ times larger than $H^{\hat{\mathcal{F}}}_{\sigma}$. This proves the result. □

Basically, this means that if we have a small L_1 covering number for the mappings, we will have the uniform convergence result we were looking for. The following theorem proves this result.

Theorem 3. *Let $\mathcal{F}$ be a set of mappings with (η, ϵ)-uniqueness property. Then there for some constant α we have*

$$m^{UC}_{\mathcal{F}}(\epsilon, \delta) \leq O(\frac{\log k! + \log \mathcal{N}(\mathcal{F}, d^X_{L_1}, \frac{\eta}{\alpha}) + \log(\frac{1}{\delta})}{\epsilon^2}) \tag{4.23}$$

Proof. Following the previous lemma, if we have a small L_1-covering number for $\mathcal{F}$, we will also have a small covering number for $H^{\mathcal{F}}$ as well. But based on standard uniform convergence theory, if a hypothesis class has small covering number, then it has uniform convergence property. More precisely, (e.g., Theorem 17.1 in [4]) we have:

$$m^{UC}_{H^{\mathcal{F}}}(\epsilon_0, \delta) \leq O(\frac{\log \mathcal{N}(H^{\mathcal{F}}, d^X_{L_1}, \frac{\epsilon_0}{16}) + \log(\frac{1}{\delta})}{\epsilon_0^2}) \tag{4.24}$$

Applying Lemma 2 to the above proves the result. □

4.4.4 Bounding L_1-Covering Number

In the previous section, we proved if the L_1-covering number of the class of mappings is bounded, then we will have uniform convergence. However, it is desirable to have a bound with respect to a combinatorial dimension of the class (rather than the covering number). Therefore, we will generalize the notion of pseudo-dimension for the class of mappings that take value in $\mathbb{R}^n$.

Let $\mathcal{F}$ be a set of mappings form X to $\mathbb{R}^n$. For every mapping $f \in \mathcal{F}$, define real-valued functions $f_1, \ldots, f_n$ such that $f(x) = (f_1(x), \ldots, f_n(x))$. Now let $F_i = \{f_i : f \in F\}$. This means that $F_1, F_2, \ldots, F_n$ are classes of real-valued functions. Now we define pseudo-dimension of $\mathcal{F}$ as follow.

$$Pdim(\mathcal{F}) = n \max_{i \in [n]} Pdim(F_i) \tag{4.25}$$

Proposition 2. *Let $\mathcal{F}$ be a set of mappings form X to $\mathbb{R}^n$. If $Pdim(F) \leq q$ then*

$$\log \mathcal{N}(F, d_{L_1}^X, \epsilon) = \mathcal{O}(q)$$

where $\mathcal{O}()$ hides logarithmic factors.

Proof. The result follows from the corresponding result for bounding covering number of real-valued functions based on pseudo-dimension mentioned in the preliminaries section. The reason is that we can create a cover by composition of the $\frac{\epsilon}{n}$-covers of all F_i. However, this will at most introduce a factor of n in the logarithm of the covering number. □

Therefore, we can rewrite the result of the previous section in terms of pseudo-dimension.

Theorem 4. *Let $\mathcal{F}$ be a class of mappings with (η, ϵ)-uniqueness property. Then*

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq \mathcal{O}(\frac{k + Pdim(\mathcal{F}) + \log(\frac{1}{\delta})}{\epsilon^2}) \tag{4.26}$$

where $\mathcal{O}()$ hides logarithmic factors of k and $\frac{1}{\eta}$.

4.5 Sample Complexity of PAC-ReKLAD

In this section, we provide the main result of this chapter. In Section 4.2 we had showed that uniform convergence is sufficient for a TERM algorithm to work. Also, in the previous section, we proved a bound for the sample complexity of uniform convergence. The following theorem, which is the main technical result of this chapter, combines these two and provides a sample complexity upper bound for PAC-ReKLAD framework.

Theorem 5. (Sample Complexity of ReKLAD)

Let $\mathcal{F}$ be a class of (η, ϵ)-unique mappings. Then the sample complexity of representation learning for k-means clustering (ReKLAD) with respect to $\mathcal{F}$ is upper bounded by

$$m_{\mathcal{F}}(\epsilon, \delta) \leq \mathcal{O}(\frac{k + Pdim(\mathcal{F}) + \log(\frac{1}{\delta})}{\epsilon^2}) \tag{4.27}$$

where $\mathcal{O}$ hides logarithmic factors of k and $\frac{1}{\eta}$.

The proof is done by combining Theorems 1 and 4.

The following result shows an upper bound for the sample complexity of learning linear mappings (or equivalently, Mahalanobis metrics).

Corollary 1. *Let $\mathcal{F}$ be a set of (η, ϵ)-unique* linear *mappings from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$. Then we have*

$$m_{\mathcal{F}}(\epsilon, \delta) \leq \mathcal{O}(\frac{k + d_1 d_2 + \log(\frac{1}{\delta})}{\epsilon^2}) \tag{4.28}$$

Proof. It is a standard result that the pseudo-dimension of a vector space of real-valued functions is just the dimensionality of the space (in our case d_1) (e.g., Theorem 11.4 in [4]). Also, based on our definition of $Pdim$ for $\mathbb{R}^{d_2}$-valued functions, it should scale by a factor of d_2. □

Chapter 5

Conclusions and Future Plans

In this proposal we introduced the problem of clustering with advice (CLAD) and provided a formal statistical framework for analyzing such framework. In particular, we modeled CLAD as a representation learning problem, called ReCLAD (i.e., representation learning for CLAD).

In ReCLAD, the learner—unaware of the target clustering of the domain—is given a clustering of a small sample set. The learner's task is then finding a mapping (among a class of mappings) under which the result of clustering of the domain is as close as possible to the true clustering. For the special case of k-means clustering, this framework was called ReKLAD.

In particular, the notion of PAC-ReCLAD was introduced in Chapter 3, specifying formally our expectations from a semi-supervised clustering algorithm. Then, an important question was raised: what is the sample complexity of PAC-ReCLAD?

In Chapter 4, we provided the results on the sample complexity of PAC-ReKLAD. More specifically, a notion of *vector-valued pseudo-dimension* for the class of mappings was defined, and the sample complexity was upper bounded based on it. This means that for the classes with higher such dimension, more clustered samples are required. Furthermore, it was proved that any ERM-type algorithm that has access to such a sample will work satisfactorily

In order to prove this result, a notion of uniform convergence was defined, and it was shown that the rate of convergence depends on the pseudo-dimension of the class of mappings. This was in turn proved using a bound on the covering number of the set of mappings.

5.1 Future Research Directions

Although we have had promising results, lots of open questions remain to be answered. We plan to address these issues in our future research.

- In our framework, we assumed that the number of clusters is given and fixed for both the main task (i.e., clustering of the whole domain set) and the clustering of the given sample. However, it is conceivable that the domain expert would partition the small sample into fewer number of clusters. Therefore, it is important to "learn" how to pick the right number of clusters as well.

- In our model, we *indexed* clustering methods by a set of mappings. For this reason, we fixed a *rich* clustering algorithm (i.e., k-means clustering) and searched for the right mapping. However, a more direct way is to index clustering algorithms by a parametrized family of objective functions. The supervised task is then to find those parameters.

- The choice of k-means clustering was rather arbitrary, except that it is *rich*. Therefore, it will be useful to extend the results of PAC-ReKLAD to other clustering algorithms (i.e., considering the general PAC-ReCLAD framework).

- It can be noted that we did not analyze the computational complexity of the proposed algorithms for PAC-ReKLAD. In fact, the problem is NP-hard, as the standard k-means clustering is hard even without learning the representation. However, it is important to provide computationally efficient algorithms. This can be done either by picking other clustering algorithms or by exploiting the "niceness" of data-generating distribution (e.g., a similar notion of uniqueness proposed by [7] makes the complexity of k-means clustering algorithm polynomial.)

- There are other supervision protocols that were discussed in Chapter 2. In particular, in many cases the supervised feedback is in the form of pairwise constraints. This is in contrast to CLAD framework where the domain expert gives the clustering of a random sample. Therefore, it is important to study the connection between these two scenarios, and possibly extend our results to the other case.

- Another supervision protocol which has not been studied yet is the *comparison-based* clustering where the domain expert is asked to compare two given clusterings and should select one that is better. This can be more intuitive for the expert in many cases.

- In CLAD framework, we assumed that the clustered sample is picked randomly. However, we may also consider an *active* setting, where the learner chooses this sample set gradually.

- One other observation is that representation learning can be regarded as a special case of metric learning; because for every mapping, we can define a distance function that computes the distance in the mapped space. In this light, we can make the problem more general by requiring the learner to choose a distance function rather than a mapping. This is a more challenging problem and still open.

- In this proposal, we used supervision as a tool to capture domain knowledge. However, in addition to information-theoretic benefits of supervised feedback, there can be computational gains as well. For example, k-means clustering is NP-hard. However, if we have access to an oracle (i.e., domain expert), we may be able to find the solution using a few queries. This line of research—which is parallel to what we studied in this proposal—is also a totally new and potentially fruitful research direction.

Appendix A

Proof of Lemma 1

Let $\mathcal{F} : X \mapsto (0,1)^n$ be a set of mappings that have (η, ϵ)-uniqueness property. Let $f_1, f_2 \in \mathcal{F}$ and $d_{L_1}(f_1, f_2) < \frac{\eta}{12}$. We need to prove that $\Delta_X(f_1, f_2) < 2\epsilon$. In order to prove this, note that due to triangular inequality, we have

$$\begin{aligned}\Delta_X(f_1, f_2) &= \Delta_X(C^{f_1}(\mu^{f_1}), C^{f_2}(\mu^{f_2})) \\ &\leq \Delta_X(C^{f_1}(\mu^{f_1}), C^{f_1}(\mu^{f_2})) + \Delta_X(C^{f_1}(\mu^{f_2}), C^{f_2}(\mu^{f_2})) \quad \text{(A.1)}\end{aligned}$$

Therefore, it will be sufficient to show that each of the Δ-terms above is smaller than ϵ. We start by proving a useful lemma.

Lemma 3. *Let $f_1, f_2 \in \mathcal{F}$ and $d_{L_1}(f_1, f_2) < \frac{\eta}{6}$. Let μ be an arbitrary set of k centers in $(0,1)^n$. Then*

$$|COST_X(f_1, \mu) - COST_X(f_2, \mu)| < \frac{\eta}{2}$$

Proof.

$$\begin{aligned}&|COST_X(f_1, \mu) - COST_X(f_2, \mu)| \\ &\qquad = \left| \left(\frac{1}{|X|} \sum_{x \in X} \min_{\mu_j \in \mu} \|f_1(x) - \mu_j\|^2 \right) - \left(\frac{1}{|X|} \sum_{x \in X} \min_{\mu_j \in \mu} \|f_2(x) - \mu_j\|^2 \right) \right| \quad \text{(A.2)}\end{aligned}$$

$$\leq \frac{1}{|X|}\sum_{x\in X}\max_{\mu_j\in\mu}\Big|\|f_1(x)-\mu_j\|^2-\|f_2(x)-\mu_j\|^2\Big| \tag{A.3}$$

$$=\frac{1}{|X|}\sum_{x\in X}\max_{\mu_j\in\mu}\Big|\|f_1(x)\|^2-\|f_2(x)\|^2-2<\mu_j,f_1-f_2>\Big| \tag{A.4}$$

$$=\frac{1}{|X|}\sum_{x\in X}\max_{\mu_j\in\mu}\Big|<f_1-f_2,f_1+f_2-2\mu_j>\Big| \tag{A.5}$$

$$\leq \frac{3}{|X|}\sum_{x\in X}\|f_1-f_2\|\leq\frac{3\eta}{6}\leq\frac{\eta}{2} \tag{A.6}$$

□

Now we are ready to prove that the first Δ-term is smaller than ϵ, i.e., $\Delta_X(C^{f_1}(\mu^{f_1}),C^{f_1}(\mu^{f_2}))<\epsilon$. But to do so, we only need to show that $COST_X(f_1,\mu^{f_2})-COST_X(f_1,\mu^{f_1})<\eta$; because in that case, due to (η,ϵ)-uniqueness property of f_1, the result will follow. Now, using Lemma 3, we have

$$COST_X(f_1,\mu^{f_2})-COST_X(f_1,\mu^{f_1}) \tag{A.7}$$

$$\leq\Big(COST_X(f_2,\mu^{f_2})+\frac{\eta}{2}\Big)-COST_X(f_1,\mu^{f_1}) \tag{A.8}$$

$$=\min_{\mu}(COST_X(f_2,\mu))-\min_{\mu}(COST_X(f_1,\mu))+\frac{\eta}{2} \tag{A.9}$$

$$\leq\max_{\mu}\left(COST_X(f_2,\mu)-COST_X(f_1,\mu)\right)+\frac{\eta}{2} \tag{A.10}$$

$$\leq\frac{\eta}{2}+\frac{\eta}{2}\leq\eta \tag{A.11}$$

where in the first and the last line we used Lemma 3.

Finally, we need to prove the second Δ-inequality, i.e., $\Delta_X(C^{f_1}(\mu^{f_2}),C^{f_2}(\mu^{f_2}))\leq\epsilon$. Assume contrary. But based on (η,ϵ)-uniqueness property of f_2, we conclude that $COST_X(f_2,C^{f_1}(\mu^{f_2}))-$

$COST_X(f_2, C^{f_2}(\mu^{f_2})) \geq \eta$. In the following, we prove that this cannot be true, and hence a contradiction.

Let $m_x = \arg\min_{\mu_0 \in \mu^{f_2}} \|f_1(x) - \mu_0\|^2$. Then, based on the boundedness of $f_1(x)$,$f_2(x)$ and we have:

$$COST_X(f_2, C^{f_1}(\mu^{f_2})) - COST_X(f_2, C^{f_2}(\mu^{f_2})) \tag{A.12}$$

$$= \left(\frac{1}{|X|} \sum_{x \in X} \|f_2(x) - m_x\|^2 \right) - COST_X(f_2, \mu_2) \tag{A.13}$$

$$= \left(\frac{1}{|X|} \sum_{x \in X} \|f_2(x) - f_1(x) + f_1(x) - m_x\|^2 \right) - COST_X(f_2, \mu_2) \tag{A.14}$$

$$\begin{aligned} = & \frac{1}{|X|} \sum_{x \in X} \|f_2(x) - f_1(x)\|^2 + \frac{1}{|X|} \sum_{x \in X} \|f_1(x) - m_x\|^2 \\ & + \frac{1}{|X|} \sum_{x \in X} 2 < f_2(x) - f_1(x), f_1(x) - m_x > - COST_X(f_2, \mu_2) \end{aligned} \tag{A.15}$$

$$\begin{aligned} \leq & \frac{2}{|X|} \sum_{x \in X} \|f_2(x) - f_1(x)\| + COST_X(f_1, \mu_1) \\ & + \frac{4}{|X|} \sum_{x \in X} \|f_2(x) - f_1(x)\| - COST_X(f_2, \mu_2) \end{aligned} \tag{A.16}$$

$$\leq \frac{6}{|X|} \sum_{x \in X} \|f_2(x) - f_1(x)\| + (COST_X(f_1, \mu_1) - COST_X(f_2, \mu_2)) \tag{A.17}$$

$$\leq \frac{6\eta}{12} + \frac{\eta}{2} \leq \eta \tag{A.18}$$

References

[1] Margareta Ackerman, Shai Ben-David, and David Loker. Towards property-based classification of clustering paradigms. In *Advances in Neural Information Processing Systems*, pages 10–18, 2010.

[2] Babak Alipanahi, Michael Biggs, Ali Ghodsi, et al. Distance metric learning vs. fisher discriminant analysis. In *Proceedings of the 23rd national conference on Artificial intelligence*, pages 598–603, 2008.

[3] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[4] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[5] Pranjal Awasthi and Reza B Zadeh. Supervised clustering. In *Advances in Neural Information Processing Systems*, pages 91–99, 2010.

[6] Maria-Florina Balcan and Avrim Blum. Clustering with interactive feedback. In *Algorithmic Learning Theory*, pages 316–328. Springer, 2008.

[7] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077. Society for Industrial and Applied Mathematics, 2009.

[8] Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, 2002.

[9] Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *Proceedings of*

the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, pages 42–49. Citeseer, 2003.

[10] Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68. ACM, 2004.

[11] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.

[12] Shai Ben-David. A framework for statistical clustering with constant time approximation algorithms for k-median and k-means clustering. *Machine Learning*, 66(2-3):243–257, 2007.

[13] Gérard Biau, Luc Devroye, and Gábor Lugosi. On the performance of clustering in hilbert spaces. *Information Theory, IEEE Transactions on*, 54(2):781–790, 2008.

[14] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM, 2004.

[15] Avrim Blum. Approximation-stability and perturbation-stability. In *DAGSTUHL Workshop on Analysis of Algorithms Beyond the Worst Case*, 2014.

[16] Ian Davidson and SS Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *Knowledge Discovery in Databases: PKDD 2005*, pages 59–70. Springer, 2005.

[17] Ayhan Demiriz, Kristin P Bennett, and Mark J Embrechts. Semi-supervised clustering using genetic algorithms. *Artificial neural networks in engineering (ANNIE-99)*, pages 809–814, 1999.

[18] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[19] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. *arXiv preprint arXiv:1102.3887*, 2011.

[20] Siddharth Gopal and Yiming Yang. Transformation-based probabilistic clustering with supervision.

[21] Akshay Krishnamurthy, Sivaraman Balakrishnan, Min Xu, and Aarti Singh. Efficient active algorithms for hierarchical clustering. *arXiv preprint arXiv:1206.4672*, 2012.

[22] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: a kernel approach. *Machine learning*, 74(1):1–22, 2009.

[23] Martin HC Law, Alexander P Topchy, and Anil K Jain. Model-based clustering with probabilistic constraints. In *SDM*. SIAM, 2005.

[24] Stuart P Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[25] Andreas Maurer and Massimiliano Pontil. k-dimensional coding schemes in hilbert spaces. *Information Theory, IEEE Transactions on*, 56(11):5839–5846, 2010.

[26] Vincent Michel, Alexandre Gramfort, Gaël Varoquaux, Evelyn Eger, Christine Keribin, and Bertrand Thirion. A supervised clustering approach for fmri-based inference of brain states. *Pattern Recognition*, 45(6):2041–2049, 2012.

[27] David Pollard. *Convergence of stochastic processes*. David Pollard, 1984.

[28] Wei Tang, Hui Xiong, Shi Zhong, and Jie Wu. Enhancing semi-supervised clustering: a feature projection perspective. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 707–716. ACM, 2007.

[29] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[30] Konstantin Voevodski, Maria-Florina Balcan, Heiko Röglin, Shang-Hua Teng, and Yu Xia. Active clustering of biological sequences. *The Journal of Machine Learning Research*, 13(1):203–225, 2012.

[31] Ulrike Von Luxburg and Shai Ben-David. Towards a statistical theory of clustering. In *Pascal workshop on statistics and optimization of clustering*, pages 20–26, 2005.

[32] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[33] Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.