Thesis for The Degree of Licentiate of Engineering

Query-Efficient Correlation Clustering via Active Learning of Pairwise Similarities

Linus Aronsson

Division of Data Science and AI
Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2024

Query-Efficient Correlation Clustering via Active Learning of Pairwise Similarities

Linus Aronsson

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Data Science and AI
Some Research Group
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2024.

Query-Efficient Correlation Clustering via Active Learning of Pairwise Similarities

Linus Aronsson

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Clustering is an important unsupervised learning problem used to group objects based on their characteristics. Correlation clustering is arguably the most natural form of clustering because it only assumes access to a pairwise similarity measure between objects, where the similarities can be expressed as any positive or negative real number. This makes correlation clustering widely applicable to many problems, even when high quality feature vectors are not available. However, obtaining pairwise similarities between all objects may be expensive and impractical in many contexts. Motivated by this, we study the problem of finding high quality correlation clustering solutions with a constrained budget of queries for pairwise similarities. Acquiring the most informative data within a constrained budget is generally studied in the field of *active learning*. Therefore, we develop a generic pool-based batch active learning procedure with the purpose of performing query-efficient correlation clustering, which is highly robust to noisy oracle feedback.

Keywords

Active learning, active clustering, correlation clustering, acquisition function, batch active learning, noisy active learning.

List of Papers

Appended papers

This thesis is based on the following two papers:

[**Paper I**] **Linus Aronsson**, Morteza Haghir Chehreghani. *Correlation Clustering with Active Learning of Pairwise Similarities.* Transactions on Machine Learning Research, 2024.

[**Paper II**] **Linus Aronsson**, Morteza Haghir Chehreghani. *Information-Theoretic Active Correlation Clustering.* arXiv preprint arXiv:2402.03587, 2024. Under submission.

Other papers

The following papers were part of my PhD studies. However, they are not appended to this thesis, due to contents not directly related to the thesis.

[**a**] Sanna Jarl, **Linus Aronsson**, Sadegh Rahrovani, Morteza Haghir Chehreghani. *Active learning of driving scenario trajectories.* Engineering Applications of Artificial Intelligence 113, 104972, 2022.

[**b**] Peter Samoaa, **Linus Aronsson**, Antonio Longa, Philipp Leitner, Morteza Haghir Chehreghani. *A unified active learning framework for annotating graph data with application to software source code performance prediction.* arXiv preprint arXiv:2304.13032, 2023. Under submission.

[**c**] Peter Samoaa, **Linus Aronsson**, Philipp Leitner, Morteza Haghir Chehreghani. *Batch Mode Deep Active Learning for Regression on Graph Data.* IEEE International Conference on Big Data (BigData), 5904-5913, 2023.

Acknowledgment

I would like to start by thanking my main supervisor, Morteza Haghir Chehreghani. Your enthusiasm, positivity, and extensive knowledge have been invaluable to my research so far. I also want to thank my co-supervisor Dag Wedelin and examiner Graham Kemp. I am grateful to all colleagues and fellow PhD students for creating a great environment to work in. I want to thank all administrative and technical staff for all the great help. Finally, I want to thank my family and friends for their continued support throughout my PhD studies.

The work in this thesis was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations and data handling were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) and the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE), High Performance Computing Center North (HPC2N) and Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) partially funded by the Swedish Research Council through grant agreements no. 2022-06725 and no. 2018-05973.

Contents

Part I

Summary

Chapter 1

Introduction

Clustering is an important unsupervised learning problem used to group objects based on their characteristics. The primary objective is to ensure that objects within the same cluster are highly similar, while objects in different clusters are distinctly dissimilar. Various clustering methods have been developed to address this problem in different settings. When this abstract description of a clustering task is directly formulated as an optimization problem, we obtain the well-known problem of *correlation clustering* (CC) (Bansal et al., 2004). Given a set of N objects and a pairwise similarity measure between them, the objective of CC is to partition the objects into clusters that maximize within-cluster similarity and minimize between-cluster similarity. Unlike many other clustering methods, CC does not require the number of clusters to be predetermined; instead, it determines the optimal number of clusters automatically based on the available pairwise similarities.

CC is arguably the most natural and simple form of clustering, as it only requires a pairwise similarity measure between objects. This makes CC highly general and is consequently applicable to a wide range of applications, including image segmentation (Kim et al., 2011), bioinformatics (Bonchi et al., 2013), social network analysis (Bonchi et al., 2012), and clustering aggregation (Chehreghani & Chehreghani, 2020). CC was initially explored using binary pairwise similarities in $\{-1, +1\}$ (Bansal et al., 2004), and was later extended to support arbitrary positive and negative pairwise similarities (Charikar et al., 2005; Demaine et al., 2006). Finding the optimal solution for CC is known to be NP-hard and APX-hard (Bansal et al., 2004; Demaine et al., 2006), presenting significant challenges. As a result, various approximation algorithms have been developed to address this problem. Among these, methods based on local search are noted for their superior performance in terms of clustering quality and computational efficiency (Thiel et al., 2019; Chehreghani, 2023).

Existing CC methods typically presume the availability of all $\binom{N}{2}$ pairwise similarities in advance (where N is the number of objects). However, as highlighted in (Bressan et al., 2019; García-Soriano et al., 2020), generating similarities can be computationally demanding and may require resource-intensive queries, such as those provided by human experts. For example,

identifying interactions between biological entities often necessitates the skills of highly trained professionals, consuming both time and valuable resources (García-Soriano et al., 2020). In tasks like entity resolution, obtaining similarity queries through crowdsourcing can also incur monetary costs. Consequently, we are interested in obtaining good CC solutions with a limited number of queries for pairwise similarities between objects.

In machine learning, such a question is generally addressed by *active learning*. Its objective is to acquire the most informative data within a constrained budget. Active learning has proven effective in various tasks, including software performance prediction (Samoaa et al., 2023), recommender systems (Rubens et al., 2015), sound event detection (Zhao et al., 2020), analysis of driving time series (Jarl et al., 2022), drug discovery (Viet Johansson et al., 2022), and analysis of logged data (Yan et al., 2018). In the context of active learning, the selection of which data to query is guided by an *acquisition function*.

Active learning is most commonly studied for classification and regression problems (Settles, 2009). However, it has also been studied for clustering and is sometimes referred to as *supervised clustering* (Awasthi & Zadeh, 2010). The motivation for supervised clustering is that by including some supervision in the clustering process, one can find clusters that are more relevant to the application and better reflect the expectations of the users. The objective is to discover the ground-truth clustering with a minimal number of queries to an *oracle* (e.g., a human expert). In this scenario, queries are typically executed in one of two ways: (i) By asking whether two clusters should merge or if one cluster should be divided into multiple clusters (Balcan & Blum, 2008; Awasthi & Zadeh, 2010); (ii) By querying the pairwise relations between objects (Basu et al., 2004; García-Soriano et al., 2020), which we focus on in this thesis.

In this thesis, we present a generic framework for active learning of pairwise similarities in the context of CC, where the queries for similarities are assumed to be noisy[1]. The thesis is structured in the following way. Chapter 2 provides the necessary background knowledge to help understand the material in the appended papers. Chapter 3 summarizes the contributions of each paper in detail. Chapter 4 contains concluding remarks on the work done so far and includes a brief discussion on possible future directions for this research project.

The following research questions are considered in this thesis.

- RQ1. *How can active learning be used to deliver satisfactory CC results with a limited number of queries for pairwise similarities between objects assuming a noisy oracle?*
- RQ2. *Can information-theoretic acquisition functions be extended to active learning of pairwise relations, as well as to non-parametric models such as CC?*[2]

Two papers are appended in the second part of the thesis, both of which

[1] Noisy queries means the oracle may provide incorrect feedback with some probability (see Section 2.2.4).

[2] Information-theoretic acquisition functions are commonly studied for active learning in the context of supervised learning (see Section 2.2.5 for details).

are listed and briefly summarized below, where additional details are provided in Chapter 3.

- Paper 1 (Aronsson & Chehreghani, 2024a) focuses on RQ1. The contributions are: (i) A generic framework for active learning for CC which offers several advantages over previous work on query-efficient CC; (ii) An efficient and noise-robust local search algorithm for CC, which automatically determines the number of clusters, to be used within the framework; (iii) Two acquisition functions called *maxmin* and *maxexp* to be used within the framework when selecting which similarities to query; (iv) Several experimental studies demonstrating the effectiveness of our framework compared to previous work.
- Paper 2 (Aronsson & Chehreghani, 2024b) builds on the framework from Paper 1 with a focus on the development of more effective acquisition functions. This means RQ1 is also considered in Paper 2, but RQ2 is the main focus. The contributions of Paper 2 are: (i) Four information-theoretic acquisition functions based on *entropy* and *information gain*; (ii) Extensive experimental studies demonstrating the benefit of all proposed acquisition functions.

Chapter 2

Background

In this chapter, we provide a background of the concepts discussed in the appended papers.

2.1 Correlation clustering

The following section provides a formal definition of correlation clustering (CC) and describes common approximation algorithms for CC.

2.1.1 Definition

We are given a set of N objects (data points) indexed by $\mathcal{V} = \{1, \dots, N\}$. The set of pairs of objects in $\mathcal{V}$ is denoted by $\mathcal{E} = \{(u, v) \mid u, v \in \mathcal{V}\}$. Let $\sigma : \mathcal{E} \to \mathbb{R}$ be a function representing the pairwise similarity measure between all objects in $\mathcal{V}$[1]. We assume $\sigma(u, u) = 0$ and $\sigma(u, v) = \sigma(v, u)$, resulting in $\binom{N}{2} = (N \times (N-1))/2$ unique similarities. A clustering is a partition of $\mathcal{V}$. In this section, we encode a clustering with K clusters as a clustering solution $\boldsymbol{c} \in \mathbb{K}^N$ where $\mathbb{K} = \{1, \dots, K\}$ and $c_u \in \mathbb{K}$ denotes the cluster label of object $u \in \mathcal{V}$. We denote by $\mathcal{C}$ the set of clustering solutions for all possible partitions (clusterings) of $\mathcal{V}$. We say a pair $(u, v) \in \mathcal{E}$ *violates* a clustering $\boldsymbol{c}$ if they are in the same cluster with $\sigma(u, v) < 0$ or in different clusters with $\sigma(u, v) \geq 0$. Given similarities σ and a clustering solution $\boldsymbol{c} \in \mathcal{C}$, let $V : \mathcal{E} \to \mathbb{R}^+$ be a function representing cluster violations and is defined as follows.

$$V(u, v \mid \boldsymbol{c}) \triangleq \begin{cases} |\sigma(u, v)| & \text{if } (u, v) \text{ violates } \boldsymbol{c} \\ 0 & \text{otherwise} \end{cases}$$

Given this, the CC cost function $R : \mathcal{C} \to \mathbb{R}^+$ aims to penalize the sum of violations and is defined as

[1] In Paper 2, we describe the pairwise similarities using a matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$, where $S_{uv} = \sigma(u, v)$, as this notation was better suited for the content of this paper.

$$R(\boldsymbol{c}) \triangleq \sum_{(u,v)\in\mathcal{E}} V(u,v \mid \boldsymbol{c}). \tag{2.1}$$

The goal of CC is to find the optimal clustering $\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}\in\mathcal{C}} R(\boldsymbol{c})$. However, this problem is known to be NP-hard and APX-hard (Bansal et al., 2004; Demaine et al., 2006). As a consequence, a number of approximation algorithms have been derived in the CC literature. In the next section, we describe some of these methods.

2.1.2 Approximation algorithms

Many approximation algorithms have been proposed in the CC literature (Bansal et al., 2004; Charikar et al., 2005; Demaine et al., 2006; Giotis & Guruswami, 2006; Ailon et al., 2008; Elsner & Schudy, 2009). In this section, we describe the methods most relevant to the appended papers of this thesis: pivot and local search.

Pivot

Pivot-based CC algorithms are often noted for their simplicity and computational efficiency. In this section, we describe the classic pivot-based CC algorithm *QwickCluster* (Ailon et al., 2008). The procedure is outlined in Alg. 1. It begins by selecting a pivot $u \in \mathcal{V}$ uniformly at random. Then, it forms a cluster with u and all objects in $\mathcal{V} \setminus u$ which have a positive similarity with u. This process is then iterated on all remaining objects until no objects are left.

Alg. 1 is the basis for all existing works on query-efficient correlation clustering (Mazumdar & Saha, 2017; Bressan et al., 2019; García-Soriano et al., 2020; Silwal et al., 2023). However, it suffers from a number of limitations in practice: (i) It is sensitive to noise. This is because after forming a cluster on line 5 with pivot u, all objects with a similarity to u which is incorrectly positive (due to noise) are permanently stuck in the wrong cluster; (ii) It cannot utilize the magnitude of the similarities. In other words, if $\sigma(u,v) \triangleq \text{sign}(\sigma(u,v))$ for all pairs, such that all similarities are binary in $\{-1,+1\}$, Alg. 1 would produce the same clustering.

Algorithm 1 QwickCluster (Ailon et al., 2008)

1: **Input:** Objects $\mathcal{V}$, similarity function σ.
2: $\mathcal{O} \leftarrow \mathcal{V}$
3: **while** $\mathcal{O} \neq \emptyset$ **do**
4: Select pivot $u \in \mathcal{O}$ uniformly at random
5: $\mathcal{O}_u \leftarrow \{u\} \cup \{v \in \mathcal{O} \mid \sigma(u,v) \geq 0\}$
6: Output cluster $\mathcal{O}_u$
7: $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{O}_u$
8: **end while**

Local search

Local search methods are often noted for their superior performance in practice, compared to other approximation algorithms for CC (Thiel et al., 2019; Chehreghani, 2023). In Alg. 2, we outline a generic description of local search for CC (Elsner & Schudy, 2009). The procedure begins by initializing a random clustering $\boldsymbol{c} \in \mathcal{C}$ with a fixed number of clusters K. Then, it iterates two steps until convergence: (i) Select an object $u \in \mathcal{V}$ uniformly at random; (ii) Move the object u to the cluster which maximally reduces the cost $R(\boldsymbol{c})$ in Eq. 2.1. The procedure has converged to a local minimum when there is no object remaining which can decrease the cost if moved to a different cluster.

There are a number of ways to improve the local search method in Alg. 2: (i) A slight modification of the algorithm allows it to automatically determine the number of clusters. In Paper 1, appended in the second part of this thesis, we describe the details of this; (ii) One can run Alg. 2 T times, and return the clustering which converges to the best local minima (i.e., lowest cost); (iii) The calculation of the cost $R(\boldsymbol{c})$ on line 5 is $O(KN^2)$. This can be improved to $O(KN)$ by noting that the object u only impacts the cost $R(\boldsymbol{c})$ for the $O(N)$ pairs to which it belongs.

Algorithm 2 Local Search (Elsner & Schudy, 2009)

1: **Input:** Objects $\mathcal{V}$, similarity function σ, number of clusters K.
2: $\boldsymbol{c} \leftarrow$ random clustering in $\mathcal{C}$ with K clusters
3: **while** not converged **do**
4: Select $u \in \mathcal{V}$ uniformly at random
5: Assign object u to cluster $k \in \{1, \ldots, K\}$ that maximally reduces cost $R(\boldsymbol{c})$
6: **end while**
7: **return** $\boldsymbol{c}$

2.2 Active learning

In this section, we describe active learning (AL). We begin by motivating the use of AL compared to traditional passive learning. After this, we provide a formal definition.

2.2.1 Motivation

Many machine learning paradigms such as supervised learning and unsupervised learning are traditionally performed *passively*. This means the learning algorithm is given all input information beforehand (e.g., a fully labelled training set for supervised learning), and is expected to return a satisfactory solution for the given problem. However, in many cases this approach is problematic for the following reasons: (i) The input information may be expensive to obtain, motivating the need for cost-efficient querying of information; (ii) Part of the input information may be noisy and misleading, hindering accurate and efficient

learning; (iii) Many machine learning problems are inherently difficult. For example, clustering is known to be an *underspecified* problem (Basu et al., 2004), leading to arbitrary solutions (clusterings) that do not reflect the expectations of the user.

AL is a principled approach to solve one or more of the above issues, depending on the context. More concretely, AL allows the learning algorithm to iteratively query an oracle (e.g., a human expert) for information that maximally improves the current solution. What constitutes improvement depends on the context, and the selection of what information to query is guided by an *acquisition function*. For example, in a clustering context, which is the main focus of the papers appended in the second part of this thesis, the information provided by the oracle can be used to guide the clustering algorithm towards a better solution. The next section provides a formal definition of AL.

2.2.2 Pool-based batch active learning

Generally, three forms of AL are considered: pool-based, stream-based and membership query synthesis (Settles, 2009). In this thesis, we consider pool-based AL, which is the most common form of AL. In pool-based AL, one assumes access to a pool of data items $\mathcal{P}$, for which we initially have partial information about, represented by $\mathcal{I}$. The kind of data items in the pool and the type of information available for each data item depends on the context. See concrete example in next section.

Alg. 3 describes a generic procedure for pool-based batch AL. Typically, AL is iterated several times until a predefined querying budget is reached. Each iteration of the procedure consists of three steps: (i) *Update* learning algorithm f given pool $\mathcal{P}$ and available information $\mathcal{I}$ to obtain current solution; (ii) *Select* a batch $\mathcal{B} \subseteq \mathcal{P}$ of B data items using query strategy $\mathcal{Q}$. The goal of the query strategy is to select the data items that provide the most information to the learning algorithm in the current iteration; (iii) *Query* the oracle for information about elements in $\mathcal{B}$ and add it to $\mathcal{I}$.

Algorithm 3 Generic Pool-Based Batch Active Learning

1: **Input:** Pool $\mathcal{P}$, initial information $\mathcal{I}$, query strategy $\mathcal{Q}$, learning algorithm f, batch size B.
2: **while** querying budget not reached **do**
3: *Update* learning algorithm f given pool $\mathcal{P}$ and available information $\mathcal{I}$ to obtain current solution.
4: *Select* batch $\mathcal{B} \subseteq \mathcal{P}$ of size $|\mathcal{B}| = B$ using query strategy $\mathcal{Q}$.
5: *Query* the oracle for information about elements in $\mathcal{B}$ and add it to $\mathcal{I}$.
6: **end while**

In the next section, we describe an instantiation of the generic procedure in Alg. 3 for supervised learning scenarios. In the appended papers of this thesis, we propose an instantiation of Alg. 3 for active learning of pairwise similarities in the context of CC.

2.2.3 Active learning of labels for supervised learning

In this section, we describe active learning of labels of data points for supervised learning scenarios (i.e., regression and classification problems), which is the most commonly studied form of active learning (Settles, 2009; Ren et al., 2022). The procedure is shown in Alg. 4, and is an instantiation of the generic procedure in Alg. 3. In this case, the set of unlabeled data points $\mathcal{U}$ correspond to the pool of data items $\mathcal{P}$. In addition, the labels of data points in $\mathcal{L}$ correspond to the information $\mathcal{I}$ queried so far. In some cases, the data points in $\mathcal{L}$ are also included in the pool. This is the case when multiple queries for the same data item are allowed, which is common when the oracle is assumed noisy (see next section). The goal of the procedure is to get the model f to perform well with as few labels queried as possible, as each query incurs some cost.

Algorithm 4 Pool-Based Batch Active Learning for Supervised Learning

Input: Unlabeled data points $\mathcal{U}$, initially labelled data points $\mathcal{L}$, query strategy $\mathcal{Q}$, model f, batch size B.
while querying budget not reached **do**
    Update model f by training on labeled data points $\mathcal{L}$.
    Select batch $\mathcal{B} \subseteq \mathcal{U}$ of size $|\mathcal{B}| = B$ using query strategy $\mathcal{Q}$.
    Query the oracle for labels of data points in $\mathcal{B}$.
    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{B}$
end while

In the following sections, we describe details of Alg. 4, including common query strategies and assumptions about the oracle.

2.2.4 Noise model of oracle

There are different ways to model the oracle when evaluating an AL procedure. The assumptions made about the oracle during evaluation are meant to reflect the practical scenario of interest. For example, the oracle may correspond to a human expert or a (possibly) less expensive automated process. Naturally, a less expensive oracle may provide more noisy feedback.

In general, the oracle responds to a query with correct information with probability $1 - \gamma$ and a noisy response with probability γ, for some noise level $\gamma \in [0, 1]$. In addition, the noise may either be *persistent* or *non-persistent*. Persistent noise means each query for the same data item returns the same (possibly noisy) feedback each time. Non-persistent noise means each query for the same data item may return different feedback each time a query is made. Consequently, multiple queries of the same data item are only useful for non-persistent noise. Most work on AL consider perfect oracles (i.e., noise level $\gamma = 0$), in which case the persistence of the noise becomes irrelevant. In studies that consider noisy oracles with $\gamma > 0$, a non-persistent noise model is a common assumption to deal with the noise (Sheng et al., 2008; Settles, 2009). This assumption is natural in practice, because it models an oracle which is

allowed to correct for previous mistakes. For example, if the query strategy suspects that the feedback provided by the oracle of some data item was incorrect, it may ask the oracle to take a closer inspection in a future iteration and change its response if needed. The papers appended in the second part of this thesis consider a strictly noisy oracle, where the noise is non-persistent.

2.2.5 Query strategies

In this section, we explain the most common query strategies proposed in the AL literature (Settles, 2009). We begin discussing single-sample selection, when the batch size is $B = 1$. Using a batch size larger than one leads to a number of difficulties, all of which are discussed in Section 2.2.7.

It is common to define a query strategy in terms of an acquisition function $a : \mathcal{P} \to \mathbb{R}$. The value $a(i)$ indicates the informativeness of data item $i \in \mathcal{P}$, where a larger value indicates higher informativeness. A selection is then made according to $i^* = \arg\max_{i \in \mathcal{P}} a(i)$.

Uncertainty sampling

Uncertainty sampling is the most commonly used query strategy in AL (Lewis & Gale, 1994). It corresponds to selecting the data item for which the current model f is the most uncertain about. This is most straightforward for probabilistic models that output conditional class probabilities $f(x \mid \mathcal{L}) = P_{\mathcal{L}}(y \mid x)$, such as neural networks with a *softmax* output layer. The conditioning on $\mathcal{L}$ means the model f is trained using labeled dataset $\mathcal{L}$. The following acquisition function is one way to quantify the uncertainty of the model given class probabilities.

$$a^{\text{LC}}(x) = 1 - P_{\mathcal{L}}(\hat{y} \mid x),$$

where $\hat{y} = \arg\max_y P(y \mid x)$. It is referred to as *least confident* (LC) in the literature (Lewis & Gale, 1994). a^{LC} only considers the most probable class. There are other alternatives that consider more or all information of the distribution $P_{\mathcal{L}}(y \mid x)$. For example, let y_1 and y_2 be the first and second most probable classes according to $P_{\mathcal{L}}(y \mid x)$. Then, *margin sampling* (MS) is defined as follows (Scheffer et al., 2001).

$$a^{\text{MS}}(x) = P_{\mathcal{L}}(y_1 \mid x) - P_{\mathcal{L}}(y_2 \mid x).$$

Finally, the most popular approach to quantify the model uncertainty is to calculate the Shannon *entropy* from information-theory (Shannon, 1948).

$$a^{\text{Entropy}}(x) = -\sum_i P_{\mathcal{L}}(y_i \mid x) \log P_{\mathcal{L}}(y_i \mid x). \tag{2.2}$$

For regression problems, it becomes less straightforward. A common solution is to consider Bayesian regression models such as Gaussian Process Regressors, which provide uncertainty estimates (Holzmüller et al., 2023).

Expected error reduction

An ideal query strategy for AL selects the data point $x \in \mathcal{U}$ which minimizes the expected future error (risk) (Roy & McCallum, 2001). Let

$$\mathrm{Err}(f, \mathcal{D}) = \int_x P(x) L(P(y \mid x), P_{\mathcal{D}}(y \mid x)) dx$$

be the risk of classifier f trained on a labeled dataset $\mathcal{D}$, where $P(x)$ and $P(y \mid x)$ are the true marginal and conditional input distributions, respectively, and $L(\cdot, \cdot)$ is some loss function. We can then define an acquisition function based on the error reduction (ER), that selects the data point that maximally reduces the risk if included in the training set.

$$a^{\mathrm{ER}}(x) = \mathrm{Err}(f, \mathcal{L}) - \mathrm{Err}(f, \mathcal{L} \cup \{(x, y')\}).$$

However, there are three problems with this: (i) The true marginal input distribution $P(x)$ is unknown; (ii) The true conditional input distribution $P(y \mid x)$ is unknown; (iii) The true label y' of the candidate data point $x \in \mathcal{U}$ is unknown before we query the oracle. The first problem is solved by Monte-Carlo estimation over the pool of unlabeled data points $\mathcal{U}$. The second problem is solved by estimating $P(y \mid x)$ with current model $P_{\mathcal{L}}(y \mid x)$. The third problem is solved by considering pseudo-labels in expectation w.r.t. the current model $P_{\mathcal{L}}(y \mid x)$.

Consider the cross-entropy loss $L(P(y \mid x), P_{\mathcal{D}}(y \mid x)) = -\sum_i P(y_i \mid x) \log P_{\mathcal{D}}(y_i \mid x))$. Under the assumption that $P(y \mid x) = P_{\mathcal{D}}(y \mid x)$, this loss reduces to the entropy $H(y \mid x, \mathcal{D}) = -\sum_i P_{\mathcal{D}}(y_i \mid x) \log P_{\mathcal{D}}(y_i \mid x)$. Given this, we obtain

$$\begin{aligned} a^{\mathrm{ER}}(x) &\approx \frac{1}{|\mathcal{U}|} \sum_{x' \in \mathcal{U}} H(y \mid x', \mathcal{L}) - \sum_i P_{\mathcal{L}}(y_i \mid x) H(y \mid x', \mathcal{L} \cup \{(x, y_i)\}) \\ &= H(Y_{\mathcal{U}} \mid \mathcal{L}) - H(Y_{\mathcal{U}} \mid x, y_x, \mathcal{L}) \\ &= I(Y_{\mathcal{U}}; y_x \mid x, \mathcal{L}), \end{aligned} \tag{2.3}$$

where $Y_{\mathcal{U}}$ is a random vector of the labels of data points in $\mathcal{U}$ and y_x is a random variable for the label of the candidate data point $x \in \mathcal{U}$. Under these assumptions, we observe that maximizing the expected error reduction is equivalent to selecting the data point $x \in \mathcal{U}$ that maximizes the expected *information gain* between labels $Y_{\mathcal{U}}$ and label y_x, where the expectation is w.r.t. the current model $P_{\mathcal{L}}(y \mid x)$. In other words, the expected reduction in entropy over labels in $Y_{\mathcal{U}}$ if x is included in the training data. Expected error reduction, unlike uncertainty sampling, may select data points which are *representative* of the input distribution (approximated by the pool $\mathcal{U}$). Because of this, a^{ER} may be less sensitive to selecting outliers. Concretely, an outlier (which could be uncertain according to the model) will provide little to no information about labels $Y_{\mathcal{U}}$, and is therefore unlikely to be selected by a^{ER}.

However, computing Eq. 2.3 for all $x \in \mathcal{U}$ can still be computationally demanding in practice because one needs to re-train the model $K|\mathcal{U}|$ times,

where K is the number of classes. This problem can be mitigated in different ways (Roy & McCallum, 2001): (i) One can evaluate Eq. 2.3 for a subset of data points in the pool $\mathcal{U}$, by first filtering out data points we suspect will not be informative; (ii) Some models f allow efficient incremental re-training when only a single data point is added to the training set (such as Naive Bayes classifiers); (iii) In some cases, one can approximate full training of the model f, for example through estimation of output variance based on the Fisher information matrix (MacKay, 1992; Cohn et al., 1994).

Much work on AL, including state-of-the-art deep AL query strategies, correspond to approximating the information gain (i.e., expected error reduction) in different ways (Ash et al., 2020; Kirsch & Gal, 2022; Ren et al., 2022). Motivated by this, in Paper 2 appended in the second part of this thesis, we propose information-theoretic acquisition functions based on entropy and information gain, to be used in the context of active learning of pairwise similarities for CC.

2.2.6 Selection bias

When the query strategy utilizes model uncertainty, such as for uncertainty sampling and expected error reduction explained in the previous section, the selections may be biased. This occurs because in early iterations of the AL procedure, the model is trained on limited (biased) information $\mathcal{L}$. As a consequence, selections that utilize uncertainty quantified by the model will also be biased. This means the model will continue to be biased in following iterations and make further suboptimal selections, and this may be hard to recover from. This is a common problem in AL (Settles, 2009).

A common approach to address this issue is to consider a *warm start* of the AL procedure. This means one assumes that $\mathcal{L}$ contains a sufficient amount of labeled data points initially, to avoid the bias as much as possible. This can be argued to be unrealistic, as in practice we may have very little or no information initially, which is referred to as a *cold start*. Because of this, some studies try to make AL work in a cold start setting (Yuan et al., 2020). Interestingly, it has been found that selection bias may be helpful in certain contexts (Farquhar et al., 2021).

2.2.7 Batch selection

With the rise of deep learning, batch selection has become increasingly popular in the context of AL. The reason for this is that if the model is expensive to train (e.g., neural networks) and/or if the pool $\mathcal{P}$ is huge, one can not afford to re-train the model for each selected data point (Ren et al., 2022). Single-sample acquisition functions, such as those explained in Section 2.2.5, can be used for batch selection by top-B selection according to $\mathcal{B}^* = \arg\max_{\mathcal{B}\subseteq\mathcal{P},|\mathcal{B}|=B} \sum_{i\in\mathcal{B}} a(i)$. However, this approach will not ensure *diversity* among elements in $\mathcal{B}$, leading to redundant information being queried. For example, assume we have a dense region in feature space where the model is currently uncertain. Acquisition functions based on uncertainty sampling may naively select all B samples from this region, when only a single sample in this region might have been sufficient

to generalize.

Consequently, many works on batch AL propose query strategies that explicitly consider interactions between samples in a batch, leading to better diversity (Kirsch et al., 2019; Ash et al., 2020; Ren et al., 2022). However, the number of possible batches grows exponentially with batch size B and pool size $|\mathcal{P}|$, leading to significant challenges. As a consequence, such methods are often computationally inefficient in practice.

The work in (Kirsch et al., 2023) proposes a simple method for improving the batch diversity for single-sample acquisition functions using noise. They propose three different approaches: *softmax*, *power* and *soft-rank*. For single-sample acquisition functions that produce non-negative scores and where samples with a score close to zero should be avoided, power acquisition is generally the best choice.[2] Power acquisition works as follows. Given any single-sample acquisition function a^{X}, let $a^{\mathrm{PowerX}}(i) = \log(a^{\mathrm{X}}(i)) + \epsilon_i$ where $\epsilon_i \sim \mathrm{Gumbel}(0; \beta^{-1})$ for all $i \in \mathcal{P}$, where $\beta \in \mathbb{R}$ is a hyperparameter. This approach leads to diversity in the selected batch, while being efficient in practice due to top-B selection. Kirsch et al., 2023 show that this approach is competitive with more complex and computationally inefficient batch selection methods.

2.2.8 Evaluation of active learning procedures

The key step in active learning is the construction of query strategies. In this section, we present two common ways to evaluate the performance of different query strategies. Both approaches are illustrated in Figure 2.1. The most common way is to use active learning curves, as shown in the left plot of Figure 2.1. This corresponds to measuring the model performance w.r.t. some metric (e.g., accuracy) at each iteration of the AL procedure (Alg. 3) for all query strategies. In this case, we see that *Strategy 1* is the best, as it reaches a good model performance with the least number of queries. In some cases, it is convenient to summarize the performance of a query strategy in a single number. A common approach is to compute the area under the corresponding active learning curve, as illustrated in Figure 2.1.

[2] In Paper 2 appended in the second part of this thesis, we consider such acquisition functions.

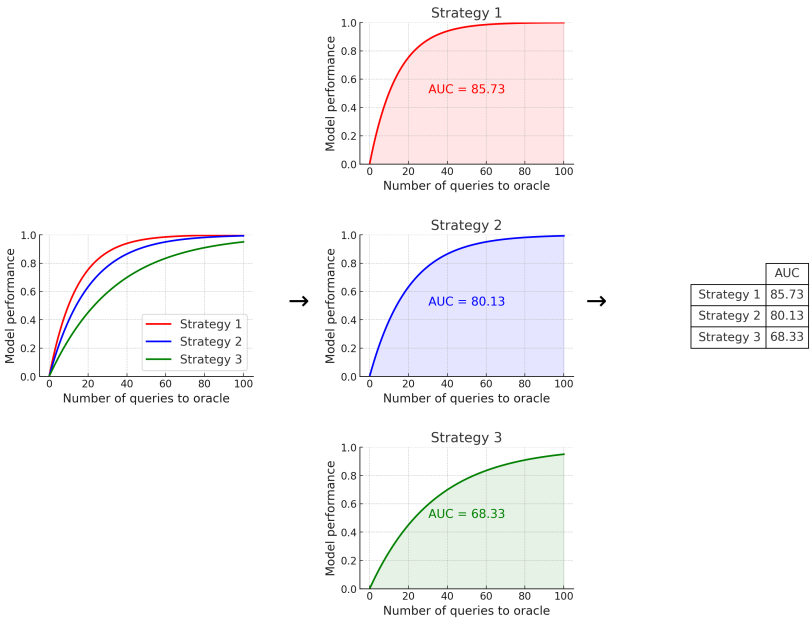

Figure 2.1: Illustration of evaluation of AL procedures. The left plot shows typical AL curves comparing the performance of different query strategies. The y-axis shows the model performance w.r.t. some performance metric (e.g., accuracy). The x-axis shows the number of queries made so far. In some cases, it is convenient to summarize the performance of a query strategy with a single number. A common approach is to compute the area under the corresponding active learning curve, as illustrated in the figure.

Chapter 3

Summary of Included Papers

In this chapter, we provide a summary of the two papers appended in the second part of this thesis.

3.1 Paper 1

In Paper 1, we study correlation clustering where the pairwise similarities are not assumed to be known beforehand. Instead, we aim to perform cost-efficient queries for the most informative pairwise similarities to a noisy oracle. Unlike previous work, we aim to address this problem using active learning. In particular, we describe a framework following the generic pool-based active learning procedure in Alg. 3. In this context, the pool $\mathcal{P}$ of data items to query information about correspond to all pairs of objects. This implies the size of the pool is $|\mathcal{P}| = \binom{N}{2} = O(N^2)$. In addition, the currently available information $\mathcal{I}$ corresponds to the queried pairwise similarities between objects.

For generality, our framework makes no assumptions about the similarities provided by the oracle. This means the pairwise similarities can be any positive or negative real number, and might even be inconsistent (i.e., violate transitivity). Without loss of generality, we restrict the similarities to the range $[-1, +1]$. In this case, $+1$ and -1 respectively indicate definite similarity and dissimilarity. Thus, a similarity close to 0 indicates a lack of knowledge about the relation between the two objects. This yields full flexibility in the type of feedback that the oracle can provide, in particular in uncertain settings. For example, assume the true similarity between objects u and v is $+1$. With binary feedback, the oracle may simply return -1 in the case of uncertainty/mistake. However, with feedback in $[-1, 1]$ their mistake/uncertainty may yield for example -0.1. A faulty feedback of -0.1 is much less severe than -1, and our framework can take advantage of this.

In this context, the learning algorithm (responsible for the *update* step of Alg. 3) corresponds to some approximation algorithm for correlation clustering

(see Section 2.1.2). The goal of the clustering algorithm is to find a clustering that (approximately) minimizes the correlation clustering cost in Eq. 2.1, given the pairwise similarities available in the current iteration. Ideally, we want a correlation clustering algorithm that (i) is robust to noisy/inconsistent similarities, (ii) automatically determines the number of clusters given the similarities and (iii) is computationally efficient. Motivated by this, we propose a correlation clustering algorithm based on local search, which fulfills the mentioned criteria.

We propose two ways to select informative pairs to query, called *maxmin* and *maxexp*. In short, both of these methods aim to query pairs with small absolute similarity that belong to triples (u, v, w) that violate the transitive property of pairwise similarities. In other words, the goal is to reduce the inconsistency of the similarities by resolving violations of the transitive property in triples. We assume a non-persistent noise model for the oracle, which both of these query strategies take advantage of.

Finally, we evaluate our framework on both synthetic and real-world datasets. We compare our framework to a query-efficient pivot-based correlation clustering algorithm called QECC (García-Soriano et al., 2020) and two adapted state-of-the-art active constraint clustering methods, called COBRAS (van Craenendonck et al., 2017) and nCOBRAS (Soenen et al., 2020). Our framework outperforms the baseline methods in a noisy setting.

3.2 Paper 2

In Paper 2, we follow the framework developed in Paper 1, with a focus on the *selection* step. In other words, we propose a number of effective acquisition functions to be used within the active correlation clustering framework. In particular, we consider information-theoretic acquisitions function based on *entropy* and *information gain*. As explained in Section 2.2.5, information-theoretic acquisition functions are commonly studied in traditional active learning for supervised learning scenarios, where labels of data points are queried. Especially, the information gain is theoretically well-motivated due to its connection to expected error reduction. We extend these methods to active learning of pairwise similarities in the context of correlation clustering. However, the proposed methods are more general than this and may be used for active learning of pairwise relations for any non-parametric learning algorithm.

In standard active learning, information-theoretic acquisition functions depend on conditional class probabilities $P_{\mathcal{L}}(y \mid x)$ (see Eq. 2.2 and Eq. 2.3). In our setting, we replace class probabilities with cluster assignment probabilities, which allow us to compute both the entropy and information gain using the same general ideas from Section 2.2.5. In short, the cluster assignment probabilities are obtained through mean-field approximation of the Gibbs distribution over clustering solutions, given the current similarities. The computational difficulties of information gain (explained in Section 2.2.5) also apply in this setting. We mitigate this issue in the following ways: (i) We filter the pool to only contain the pairs with large entropy; (ii) We perform efficient

incremental updates of the cluster assignment probabilities; (iii) We use the symmetry of information gain (or equivalently the mutual information) to obtain an alternative formulation, which allows for a more efficient approximation in practice.

Finally, we extensively evaluate all proposed acquisition functions on synthetic and real-world datasets. We observe that all information-theoretic acquisition functions outperform all baseline methods, including maxmin and maxexp. Expectedly, information gain is an overall better option compared to entropy.

Chapter 4

Discussion and Future Work

In this thesis, we study correlation clustering where the pairwise similarities are not assumed to be known beforehand. This is an important problem because, in many practical scenarios, obtaining similarities between objects can be expensive. Therefore, the goal is to find the ground-truth clustering, with as few queries for similarities between objects as possible.

In machine learning, the problem of finding the most informative data within a constrained budget is usually studied by active learning. Consequently, we address this problem by developing a generic pool-based active learning procedure for query-efficient correlation clustering, following the structure of Alg. 3. The procedure is generic because it can be used with any approximation algorithm for correlation clustering and any acquisition function. In contrast, in previous work on query-efficient correlation clustering, the selection of which similarities to query is tightly integrated into the clustering algorithm (which tends to be based on QwickCluster, see Alg. 1), leading to restrictions in practice.

We find that our proposed framework is highly robust to noisy feedback, and significantly outperforms baseline methods in the presence of noise. In addition, we find that all proposed acquisition functions (maxmin, maxexp and information-theoretic methods) outperform random selection of similarities. Overall, we have observed that acquisition functions based on information gain perform the best. However, this may depend on the setting (e.g., how the pairwise similarities are initialized). We refer to the appended papers in the second part of the thesis for more details on this.

An important aspect of query-efficient clustering is *inferring* the pairwise relation between pairs of objects which have not been queried, based on the pairwise relations between other (correlated) objects which have been queried. For example, given three objects u, v, w, assume the following similarities have been queried: $\sigma(u, v) = +1$ and $\sigma(u, w) = +1$. Given this, one can infer that $\sigma(v, w) = +1$, using the transitive property of similarities. If the oracle is assumed to provide perfect feedback (i.e., zero noise), exploiting this can lead to

a significant improvement in performance. However, with even a small amount of noise, we have found that inferring leads to worse performance compared to not doing so. Studying ways to perform noise-robust inferring of similarities is an important area of future research, as it may drastically decrease the number of queries required.

The proposed framework scales quadratically with the number of objects N in terms of both runtime and memory. This limits the applicability of the framework to fairly small problems (e.g., $N \leq 10000$). Therefore, a future direction of research is to study how the framework can be modified to support large-scale problems.

As explained in Section 2.2.6, selection bias is a common problem in active learning. This also applies to our setting, in particular for the information-theoretic acquisition functions that utilize model uncertainty. In future work, we may investigate ways to mitigate this problem without the need for a warm start of the active learning procedure (see Section 2.2.6 for details).

Finally, we consider batch selection $B > 0$, meaning that many similarities are queried in each iteration of the active learning procedure. As explained in Section 2.2.7, diversity among the elements in a batch is important for query-efficient learning. In our framework, we ensure batch diversity using acquisition noise (see Section 2.2.7 for details). However, there may be better ways to achieve batch diversity by explicitly considering interactions between elements in a batch, which may be investigated in future work.

Bibliography

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423 (cit. on p. 12).

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Comput.*, *4*(4), 590–604. https://doi.org/10.1162/NECO.1992.4.4.590 (cit. on p. 14).

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1994). Active learning with statistical models. In G. Tesauro, D. S. Touretzky & T. K. Leen (Eds.), *Advances in neural information processing systems 7* (pp. 705–712). MIT Press. (Cit. on p. 14).

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In W. B. Croft & C. J. van Rijsbergen (Eds.), *Proceedings of the 17th annual international ACM-SIGIR conference on research and development in information retrieval.* (pp. 3–12). ACM/Springer. https://doi.org/10.1007/978-1-4471-2099-5_1 (cit. on p. 12).

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In C. E. Brodley & A. P. Danyluk (Eds.), *Proceedings of the eighteenth international conference on machine learning ICML* (pp. 441–448). Morgan Kaufmann. (Cit. on pp. 13, 14).

Scheffer, T., Decomain, C., & Wrobel, S. (2001). Active Hidden Markov Models for Information Extraction. In F. Hoffmann, D. J. Hand, N. M. Adams, D. H. Fisher & G. Guimarães (Eds.), *Advances in intelligent data analysis, 4th international conference, IDA* (pp. 309–318, Vol. 2189). Springer. https://doi.org/10.1007/3-540-44816-0_31 (cit. on p. 12).

Bansal, N., Blum, A., & Chawla, S. (2004). Correlation clustering. *Machine Learning*, *56*(1-3), 89–113. https://doi.org/10.1023/B:MACH.0000033116.57574.95 (cit. on pp. 3, 8).

Basu, S., Banerjee, A., & Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In M. W. Berry, U. Dayal, C. Kamath & D. B. Skillicorn (Eds.), *Proceedings of the fourth SIAM international conference on data mining* (pp. 333–344). SIAM. https://doi.org/10.1137/1.9781611972740.31 (cit. on pp. 4, 10).

Charikar, M., Guruswami, V., & Wirth, A. (2005). Clustering with qualitative information. *J. Comput. Syst. Sci.*, *71*(3), 360–383. https://doi.org/10.1016/J.JCSS.2004.10.012 (cit. on pp. 3, 8).

Demaine, E. D., Emanuel, D., Fiat, A., & Immorlica, N. (2006). Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, *361*(2-3), 172–187. https://doi.org/10.1016/j.tcs.2006.05.008 (cit. on pp. 3, 8).

Giotis, I., & Guruswami, V. (2006). Correlation clustering with a fixed number of clusters. *Theory Comput.*, *2*(13), 249–266. https://doi.org/10.4086/TOC.2006.V002A013 (cit. on p. 8).

Ailon, N., Charikar, M., & Newman, A. (2008). Aggregating inconsistent information: Ranking and clustering. *J. ACM*, *55*(5), 23:1–23:27. https://doi.org/10.1145/1411509.1411513 (cit. on p. 8).

Balcan, M., & Blum, A. (2008). Clustering with interactive feedback. In Y. Freund, L. Györfi, G. Turán & T. Zeugmann (Eds.), *Algorithmic learning theory, 19th international conference, ALT* (pp. 316–328, Vol. 5254). Springer. https://doi.org/10.1007/978-3-540-87987-9_27 (cit. on p. 4).

Sheng, V. S., Provost, F. J., & Ipeirotis, P. G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In Y. Li, B. Liu & S. Sarawagi (Eds.), *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 614–622). ACM. https://doi.org/10.1145/1401890.1401965 (cit. on p. 11).

Elsner, M., & Schudy, W. (2009). Bounding and comparing methods for correlation clustering beyond ILP. *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, 19–27 (cit. on pp. 8, 9).

Settles, B. (2009). *Active learning literature survey* (Computer Sciences Technical Report No. 1648). University of Wisconsin–Madison. (Cit. on pp. 4, 10–12, 14).

Awasthi, P., & Zadeh, R. B. (2010). Supervised clustering. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel & A. Culotta (Eds.), *Advances in neural information processing systems 23* (pp. 91–99). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2010/hash/18997733ec258a9fcaf239cc55d53363-Abstract.html (cit. on p. 4).

Kim, S., Nowozin, S., Kohli, P., & Yoo, C. D. (2011). Higher-order correlation clustering for image segmentation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 24* (pp. 1530–1538). (Cit. on p. 3).

Bonchi, F., Gionis, A., Gullo, F., & Ukkonen, A. (2012). Chromatic correlation clustering. In Q. Yang, D. Agarwal & J. Pei (Eds.), *The 18th ACM SIGKDD international conference on knowledge discovery and data mining, KDD* (pp. 1321–1329). ACM. https://doi.org/10.1145/2339530.2339735 (cit. on p. 3).

Bonchi, F., Gionis, A., & Ukkonen, A. (2013). Overlapping correlation clustering. *Knowl. Inf. Syst.*, *35*(1), 1–32. https://doi.org/10.1007/S10115-012-0522-9 (cit. on p. 3).

Rubens, N., Elahi, M., Sugiyama, M., & Kaplan, D. (2015). Active learning in recommender systems. In F. Ricci, L. Rokach & B. Shapira (Eds.),

Recommender systems handbook (pp. 809–846). Springer. https://doi.org/10.1007/978-1-4899-7637-6_24 (cit. on p. 4).

Mazumdar, A., & Saha, B. (2017). Clustering with noisy queries. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 5788–5799). https://proceedings.neurips.cc/paper/2017/hash/db5cea26ca37aa09e5365f3e7f5dd9eb-Abstract.html (cit. on p. 8).

van Craenendonck, T., Dumancic, S., & Blockeel, H. (2017). COBRA: A fast and simple method for active clustering with pairwise constraints. In C. Sierra (Ed.), *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI* (pp. 2871–2877). ijcai.org. https://doi.org/10.24963/IJCAI.2017/400 (cit. on p. 18).

Yan, S., Chaudhuri, K., & Javidi, T. (2018). Active learning with logged data. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning (icml)* (pp. 5517–5526, Vol. 80). PMLR. (Cit. on p. 4).

Bressan, M., Cesa-Bianchi, N., Paudice, A., & Vitale, F. (2019). Correlation clustering with adaptive similarity queries. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 12510–12519). https://proceedings.neurips.cc/paper/2019/hash/b0ba5c44aaf65f6ca34cf116e6d82ebf-Abstract.html (cit. on pp. 3, 8).

Kirsch, A., van Amersfoort, J., & Gal, Y. (2019). BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 7024–7035). https://proceedings.neurips.cc/paper/2019/hash/95323660ed2124450caaac2c46b5ed90-Abstract.html (cit. on p. 15).

Thiel, E., Chehreghani, M. H., & Dubhashi, D. P. (2019). A non-convex optimization approach to correlation clustering. *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, 5159–5166. https://doi.org/10.1609/aaai.v33i01.33015159 (cit. on pp. 3, 9).

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., & Agarwal, A. (2020). Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. *8th International Conference on Learning Representations, ICLR*. https://openreview.net/forum?id=ryghZJBKPS (cit. on pp. 14, 15).

Chehreghani, M. H., & Chehreghani, M. H. (2020). Learning representations from dendrograms. *Mach. Learn.*, *109*(9-10), 1779–1802. https://doi.org/10.1007/S10994-020-05895-3 (cit. on p. 3).

García-Soriano, D., Kutzkov, K., Bonchi, F., & Tsourakakis, C. E. (2020). Query-efficient correlation clustering. In Y. Huang, I. King, T. Liu & M. van Steen (Eds.), *WWW '20: The web conference* (pp. 1468–1478). ACM / IW3C2. https://doi.org/10.1145/3366423.3380220 (cit. on pp. 3, 4, 8, 18).

Soenen, J., Dumancic, S., van Craenendonck, T., & Blockeel, H. (2020). Tackling noise in active semi-supervised clustering. In F. Hutter, K. Kersting, J. Lijffijt & I. Valera (Eds.), *Machine learning and knowledge discovery in databases ECML PKDD* (pp. 121–136, Vol. 12458). Springer. https://doi.org/10.1007/978-3-030-67661-2_8 (cit. on p. 18).

Yuan, M., Lin, H.-T., & Boyd-Graber, J. (2020, November). Cold-start active learning through self-supervised language modeling. In B. Webber, T. Cohn, Y. He & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 7935–7948). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.637 (cit. on p. 14).

Zhao, S., Heittola, T., & Virtanen, T. (2020). Active learning for sound event detection. *IEEE ACM Trans. Audio Speech Lang. Process.*, *28*, 2895–2905. https://doi.org/10.1109/TASLP.2020.3029652 (cit. on p. 4).

Farquhar, S., Gal, Y., & Rainforth, T. (2021). On statistical bias in active learning: How and when to fix it. *9th International Conference on Learning Representations, ICLR*. https://openreview.net/forum?id=JiYq3eqTKY (cit. on p. 14).

Jarl, S., Aronsson, L., Rahrovani, S., & Chehreghani, M. H. (2022). Active learning of driving scenario trajectories. *Eng. Appl. Artif. Intell.*, *113*, 104972. https://doi.org/10.1016/J.ENGAPPAI.2022.104972 (cit. on p. 4).

Kirsch, A., & Gal, Y. (2022). Unifying Approaches in Active Learning and Active Sampling via Fisher Information and Information-Theoretic Quantities. *Trans. Mach. Learn. Res.*, *2022*. https://openreview.net/forum?id=UVDAKQANOW (cit. on p. 14).

Ren, P., Xiao, Y., Chang, X., Huang, P., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2022). A survey of deep active learning. *ACM Comput. Surv.*, *54*(9), 180:1–180:40. https://doi.org/10.1145/3472291 (cit. on pp. 11, 14, 15).

Viet Johansson, S., Gummesson Svensson, H., Bjerrum, E., Schliep, A., Haghir Chehreghani, M., Tyrchan, C., & Engkvist, O. (2022). Using active learning to develop machine learning models for reaction yield prediction. *Molecular Informatics*, *41*(12), 2200043. https://doi.org/https://doi.org/10.1002/minf.202200043 (cit. on p. 4).

Chehreghani, M. H. (2023). Shift of pairwise similarities for data clustering. *Mach. Learn.*, *112*(6), 2025–2051. https://doi.org/10.1007/S10994-022-06189-6 (cit. on pp. 3, 9).

Holzmüller, D., Zaverkin, V., Kästner, J., & Steinwart, I. (2023). A framework and benchmark for deep batch active learning for regression. *J. Mach. Learn. Res.*, *24*, 164:1–164:81. http://jmlr.org/papers/v24/22-0937.html (cit. on p. 12).

Kirsch, A., Farquhar, S., Atighehchian, P., Jesson, A., Branchaud-Charron, F., & Gal, Y. (2023). Stochastic batch acquisition: A simple baseline for deep active learning. *Trans. Mach. Learn. Res.*, *2023*. https://openreview.net/forum?id=vcHwQyNBjW (cit. on p. 15).

Samoaa, P., Aronsson, L., Longa, A., Leitner, P., & Chehreghani, M. H. (2023). A unified active learning framework for annotating graph data with application to software source code performance prediction. *CoRR*, *abs/2304.13032*. https://doi.org/10.48550/ARXIV.2304.13032 (cit. on p. 4).

Silwal, S., Ahmadian, S., Nystrom, A., McCallum, A., Ramachandran, D., & Kazemi, S. M. (2023). KwikBucks: Correlation clustering with cheap-weak and expensive-strong signals. *The Eleventh International Conference on Learning Representations, ICLR*. https://openreview.net/forum?id=p0JSSa1AuV (cit. on p. 8).

Aronsson, L., & Chehreghani, M. H. (2024a). Correlation clustering with active learning of pairwise similarities. *Transactions on Machine Learning Research* (cit. on p. 5).

Aronsson, L., & Chehreghani, M. H. (2024b). Information-theoretic active correlation clustering. https://arxiv.org/abs/2402.03587 (cit. on p. 5).