Applied Intelligence
https://doi.org/10.1007/s10489-023-04516-5

Active pairwise distance learning for efficient labeling of large datasets by human experts

Joris Pries[1] · **Sandjai Bhulai**[2] · **Rob van der Mei**[1]

Accepted: 7 February 2023

Abstract

In many machine learning applications, the labeling of datasets is done by human experts, which is usually time-consuming in cases of large data sets. This raises the need for methods to make optimal use of the human expert by selecting model instances for which the expert opinion is of most added value. This paper introduces the problem of *active pairwise distance learning* (APDL), where the goal is to *actively* learn the pairwise distances between *all* instances. Any distance function can be used, which means that APDL techniques can e.g., be used to determine likeness between faces or similarities between users for recommender systems. Starting with an unlabeled dataset, each round an expert determines the distance between one pair of instances. Thus, there is an important choice to make each round: 'Which combination of instances is presented to the expert?' The objective is to accurately predict all pairwise distances, while minimizing the usage of the expert. In this research, we establish upper and lower bound approximations (including an update rule) for the pairwise distances and evaluate many domain-independent query strategies. The observations from the experiments are therefore general, and the selection strategies are ideal candidates to function as baseline in future research. We show that using the criterion *max degree* consistently ranks amongst the best strategies. By using this criterion, the pairwise distances of a new dataset can be labeled much more efficiently.

Keywords Active learning · Labeling · Pairwise distance · Optimal strategy · Human expert

1 Introduction

A dataset plays a critical part when solving a practical problem using machine learning (ML). Often, the goal is to predict some target variable using measured features of other variables. When gathering the data, it would be ideal if the target variable could be measured. For example, consider the task of forecasting the outside temperature using multiple other measurements, such as atmospheric pressure, wind speed and humidity. In this case, the label (temperature) can be determined efficiently. In other cases, the labels are not as easily acquired. For example, to predict if a face is visible in a photograph requires human expertise at some point to label a dataset. In such cases, human involvement is sometimes necessary, especially when a model is trained to replicate human knowledge or skills.

Labeling using a human expert is a time-consuming and costly undertaking. Therefore, efforts should be focused on maximizing the usefulness of the expert when it is too expensive to label everything. Typical questions are: 'How should the expert be deployed?' and 'Which samples should be labeled?' These questions are all part of the research field called *active learning* (AL) [1]. It is a subfield of ML dedicated to achieving the best prediction performance with as few labels as possible. To this end, a human expert can be queried about an instance each round. The expert then determines a label for this instance, which in turn can be used to update a prediction model and determine the next query. This cycle continues for a fixed number of rounds or until some other stopping criterion is met [2–4].

✉ Joris Pries
joris.pries@cwi.nl

Sandjai Bhulai
s.bhulai@vu.nl

Rob van der Mei
mei@cwi.nl

[1] Department of Stochastics, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, 1098 XG, Netherlands

[2] Department of Mathematics, Vrije Universiteit, De Boelelaan 1111, Amsterdam, 1081 HV, Netherlands

Published online: 28 July 2023

AL is useful in situations where simply labeling all data instances is too expensive. For example, suppose we want to label a dataset with many facial images and we are interested in learning the similarity/likeness between each combination of faces. If there are $M \in \mathbb{N}_{>0}$ faces, then there are already $\binom{M}{2} = M \cdot (M-1)/2$ pairwise combinations. To label all pairwise similarities of 1,000 faces would thus already require 499,500 comparisons. For large datasets, this quickly becomes too costly to label (either time or money wise), which is why AL techniques have been developed.

A critical aspect in AL is the selection algorithm (the so-called *query function*) that determines which samples should be given to the expert. The selection algorithm can be either pre-trained using other datasets (*transfer learning* [5, 6]), or it can be adjusted on-the-fly. AL techniques (almost always) use feature values to improve the query function, which is commonly some supervised learning method (e.g., a neural network). Yoo et al. [7] attached a module to a target network to predict the target losses for unlabeled data. Klein et al. [8] measured anomaly scores of feature values as guidance for the query function. Another common selection criterion is some kind of uncertainty sampling [9], whereby a prediction model is trained using the labeled data, and applied on the feature values of the unlabeled data. Uncertain predictions are then queried to the expert.

In this paper, we investigate an unexplored area within AL, that we call *active pairwise distance learning* (APDL). The objective in APDL is to actively learn the *pairwise distances* between all instances. Any distance function can be used, which means that APDL techniques can e.g., be used to determine likeness between faces or similarities between users for recommender systems. Furthermore, APDL methods can also be used in kinship recognition, deep fake detection, anomaly detection, dissimilarity sampling, and (pairwise) clustering. Studying APDL is therefore valuable for many research areas. It is important to emphasize that, we will not make any assumptions in this research about the relevance of the feature values to these distances (see Section 2.2 for more details), which makes our results highly generic and hence useful in many application areas.

The contribution of this research is three-fold. First, we introduce APDL, the problem of actively learning the pairwise distances between all instances. Second, we establish upper and lower bound approximations for the pairwise distances, and an update rule for these bounds. Third, we identify the best generic (domain-independent) baseline strategies for practical applications. This research can be seen as a pioneering contribution to the field of AL, which is expected to raise many follow-up studies in future research.

The remainder of this paper is organized as follows. In Section 2, we formally introduce APDL and discuss why no assumptions are made about the feature values. Consequently, we argue that techniques from unsupervised learning, semi-supervised learning and reinforcement learning are not applicable without these assumptions. Related research is discussed in Section 3. Section 4 defines notation for the selection strategies. Furthermore, it is discussed how each additional pairwise distance will update the upper and lower approximation bounds for all pairwise distances. A variety of selection strategies and selection criteria are defined in Section 5. Next, the experimental setup is addressed in Section 6. The experiments evaluate the selection strategies on multiple datasets to find the best performing strategy. The results of the experiments are discussed in Section 7. Section 9 gives an extensive overview of possible future research opportunities and addresses limitations of the results presented in this paper. Finally, Section 10 summarizes the findings.

2 Active pairwise distance learning

2.1 Definition of APDL

To start, we formally define *active pairwise distance learning* (APDL). Starting with an unlabeled dataset consisting of M instances, the objective of APDL is to learn as much as possible about the distance between *each pair of instances* in $T \in \mathbb{N}_{>0}$ rounds. Each round, an expert can be queried to label exactly one pairwise distance. After T rounds, a final prediction is made about all pairwise distances. Given a pre-determined loss function $\mathcal{L}$, the goal is to minimize the loss between the *actual* pairwise distance matrix $\mathcal{D}^{\text{true}}$ and the *predicted* pairwise distance matrix $\mathcal{D}^{\text{pred}}$. Thus, the target of any APDL algorithm is to minimize $\mathcal{L}\left(\mathcal{D}^{\text{pred}}, \mathcal{D}^{\text{true}}\right)$. In general, there are two critical components in APDL: (I) '*Which pair is queried each round?*' and (II) '*How to use this information to make the best prediction?*' The first question is the main focus of this research. The general approach of an APDL algorithm can be seen in Algorithm 1.

Input: # samples M, # rounds T, expert labeler Ω
Output: Pairwise distance prediction $\mathcal{D}^{\text{pred}}$
1: **for** $t \leftarrow 1$ to T **do**
2: Select $(i, j) \in \{1, \ldots, M\}^2$
3: Receive distance $d(i, j)$ from expert Ω
4: **end for**
5: Make final pairwise distance prediction $\mathcal{D}^{\text{pred}}$

Algorithm 1 General APDL algorithm.

2.2 No relevancy assumption

An important assumption that we make in this research is that no assumptions are made about the relevance of the feature values to the actual distance. As a consequence, only techniques that do not use the feature values are considered. Note that having similar feature values does not necessarily mean that the underlying distance between two instances is small. Insufficient features could mean that instances appear close, but are actually far apart. Having too many features could also be troublesome for measuring similarity, as instances in a high-dimensional space are often far away (due to the infamous *curse of dimensionality*). Furthermore, sufficient labeled data is required to accurately extract information from the feature values in order to make good predictions. Especially for high-dimensional data and complex prediction models, more labeled data is necessary to properly train the prediction model. Gal et al. [10] even identified the lack of scalability to high-dimensional data as one of the major remaining challenges for AL. However, in practice sufficient labeled data is not always available. In addition, a recent survey [11] stated that "research remains in its infancy at present, and there is still a long way to go in the future." A badly trained prediction model could steer the query selection in the wrong direction.

Without making any assumption about the relevancy of the feature values to the pairwise distance makes most known techniques from *unsupervised*, *semi-supervised* and *active learning* inappropriate. Chapelle et al. [12] identify in which cases *semi-supervised learning* is suitable. They determine the following three assumptions in order to apply semi-supervised learning techniques:

Smoothness assumption: "If two points x_1, x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2."

Cluster assumption: "If points are in the same cluster, they are likely to be of the same class."

Manifold assumption: "The (high-dimensional) data lie (roughly) on a low-dimensional manifold."

The *smoothness* and *cluster assumption* do not have to hold when the underlying distance metric (responsible for the actual labels) is very different from the metric that is used to measure if two points are close and if they belong to the same cluster. Consider for example determining if cars are similar using images. If the distance between two images is measured by comparing them pixel-by-pixel, it is highly likely that only the color of the car determines if two cars are similar (or even the background). Therefore, this is not a good approach.

The *manifold assumption* is important to combat the well-known curse of dimensionality problem. Without this assumption, a lot of data is necessary to learn the underlying distribution from the feature values. In such a situation, it might be better to make no assumptions than being steered in the wrong direction due to a lack of labeled data.

Techniques from *reinforcement learning* [13] have similar problems, when feature values are used. Given a specific dataset, the same action (i.e., querying the expert about a certain pair) is not repeated. Furthermore, no state is revisited and the state space can be really large. Thus, some mapping must be learned from the feature values. This inherently has the same assumption problems as discussed before.

When not to make relevancy assumption We identify six situations where it could be useful to make no assumptions about the relevancy of the feature values to the pairwise distance: (I) when there is not yet enough labeled data for supervised techniques; (II) when the underlying metric is unknown and could be too complex to predict using the given features; (III) when the features are not sufficient (IV) when there are too many features; (V) when the model should work across multiple domains; (VI) as baseline to evaluate techniques that do use feature values.

To elaborate on situation (VI), whenever for example a semi-supervised technique is developed, it should perform better than any method that does not use the feature values. Therefore, not using the feature values can be used to benchmark methods that do use feature values.

Advantages of not using feature values Not using feature values has its benefits. We list five advantages: (I) the dimensionality of data is irrelevant; (II) the quality of feature values is unimportant; (III) no hyper-parameter tuning based on feature values is needed; (IV) conclusions are not dependent on the application domain; (V) resulting baselines are ideal to be used as benchmark. As this research constitutes the first step in APDL, these are the reasons why we decide to only investigate selection strategies that do not use feature values.

3 Related research

To the best of our knowledge, APDL is a new research area within AL. However, there are related papers, which we will outline below.

APDL is not the same as *learning pairwise preferences* [14], where the goal is to make a ranking based on pairwise comparisons. In these pairwise comparisons, it is decided which sample is more preferable, which is a binary choice. A might be preferred over B, but it is not labeled by how much, which is an important distinction. Furthermore, the focus lies more on determining a good ranking function, not necessarily determining which samples should be labeled in

order to gain the most information. However, it is closely related and (non-binary) preference / desirability could also be used as a distance metric within APDL.

Dasarathy et al. [15] investigate binary label prediction on a graph. A non-parametric algorithm is developed to actively learn to predict binary labels in a graph. The objective for APDL is to learn *all* pairwise distances, thus the graph would be fully connected. The main difference with our research is that binary labels are assumed in [15], whereas we assume that the labels are generated by a distance metric. On the one hand, it makes the problem easier, as structure is added to the labels, because properties of a distance metric need to be satisfied. On the other hand, a label can now be real-valued and not only binary, which makes prediction much harder.

Actively learning pairwise similarities has also been studied for hierarchical clusterings [16]. The goal is to infer the hierarchical clustering using as few similarities as possible. These similarities are not necessarily from a distance metric, as e.g., the Pearson correlation is used in [16]. The performance is assessed by evaluating the constructed tree structures. This makes APDL different, as the objective is to predict all pairwise distances, not to identify the correct tree structure.

APDL is also closely related to *similarity learning* and *metric learning* [17–19]. These are supervised ML areas, where the goal is to learn from a labeled dataset a similarity function and a metric, respectively. The task of face verification is a practical example of these research areas. In [20], the triplet loss is used to learn a distance function from 0/1-labels to compare faces. The main difference with APDL is that similarity and metric learning require a labeled dataset in order to determine a generalized function that can be used for new samples. The objective in APDL is to gather as much distance-based information as possible about a fixed dataset, when there is yet no information about the labels. APDL is thus not concerned about finding a general function for samples outside the given dataset. APDL could be used to build the dataset that is later used by techniques from *similarity learning* and *metric learning*.

Metric learning has also been researched in an AL setting. Yang et al. [21] developed a Bayesian framework to actively learn a distance metric by selecting the unlabeled pairs with the greatest uncertainty in predicting whether the pair is in the same equivalence class or not. Kumaran et al. [22] actively learned a distance metric to identify outlier and boundary points per class, which are then given to the expert. Even more selection strategies are explored in [23]. Pasolli et al. [24] used an actively learned metric to reduce the dimensionality of hyperspectral images and to select uncertain samples. Again, the goal in *active metric learning* is to get a model to accurately predict if two samples belong to the same class, not to determine an accurate prediction for pairwise distances. This makes APDL a fundamentally different problem.

4 Definitions and bounds

First, we introduce some notation that is necessary to discuss selection strategies. As seen in Algorithm 1, in round t a pair of indices $\zeta_t := (i, j)$ is chosen from M indices and a corresponding distance $d(i, j)$ between these indices is obtained from the expert. Although it is possible to disregard previous requests to the expert, it is obvious that previous results should be taken into account when selecting the next pair of indices. If only to avoid asking the expert the same pair twice. Therefore, we introduce the notion of *history*.

Definition 1 (History) Name $\mathcal{H}_t = \{((i, j), d(i, j)) : \zeta_\tau = (i, j)\}_{\tau=1,\dots,t}$ the *history* of all chosen pairs of indices and their corresponding labeled distance up to and including round t. Furthermore, define $\mathcal{H}_0 := \emptyset \times \emptyset$.

Next, we will define what a selection strategy is. A selection strategy for T rounds consists of T functions that successively determine which pair of indices is chosen based on the given history.

Definition 2 (Selection strategy) We call σ a *selection strategy* if for each $t \in \{1, \dots, T\}$ it holds that $\sigma_t : \mathcal{H}_{t-1} \mapsto \zeta_t \in \{1, \dots, M\}^2$ and $\sigma = \{\sigma_t\}_{t=1,\dots,T}$.

4.1 Expert distance metric

After the selection strategy determines which pair of indices is chosen, the expert determines the distance between them. An important and strong assumption we make, is that the expert makes no mistakes and that the distances originate from an underlying metric $d : \{1, \dots, M\}^2 \to [0, d_{\max}]$, where $d_{\max} \in \mathbb{R}_{>0}$ is the maximum possible distance between two samples. In most instances, $d_{\max}$ can be estimated or determined. However, when the maximum distance cannot be bounded from above, consider $d_{\max}$ to be infinite. In our experiments, the underlying distance metric is the Euclidean distance between two samples and the expert simply returns the correct Euclidean distance.

4.2 Approximation bounds

To approximate the true distance between each pair of indices, we can make use of the fact that the underlying distance function d is a metric, to find upper and lower bounds. Each metric satisfies, by definition, the *triangle*

inequality and the subsequent *reverse triangle inequality*. Denote the upper and lower bound of (i, j) in round t as $\mathcal{D}_t^{\text{upp}}(i, j)$ and $\mathcal{D}_t^{\text{low}}(i, j)$, respectively. The metric d is symmetric (i.e., $d(x, y) = d(y, x)$), thus we enforce the upper and lower bounds to be symmetric as well. Therefore, it must always hold that $\mathcal{D}_t^{\text{upp}}(i, j) = \mathcal{D}_t^{\text{upp}}(j, i)$ and $\mathcal{D}_t^{\text{low}}(i, j) = \mathcal{D}_t^{\text{low}}(j, i)$. We will now discuss how triangle inequalities can be used to update the upper and lower bounds each time a new distance is obtained from the expert.

Initialization In the first round, there is no distance information yet. However, as d is a metric, it must hold that $d(i, i) = 0$ for each $i \in \{1, \ldots, M\}$. Furthermore, using the range of d, the upper and lower bounds are initialized as:

$$\mathcal{D}_1^{\text{upp}}(i, j) = \begin{cases} 0 & \text{if } i = j, \\ d_{\max} & \text{else.} \end{cases}$$
$$\mathcal{D}_1^{\text{low}}(i, j) = 0.$$

Triangle inequality The *triangle inequality* states that for all $a, b, c \in \{1, \ldots, M\}$ it must hold that $d(a, c) \leq d(a, b) + d(b, c)$. Expanding on this, for every round t it follows that

$$d(a, c) \leq d(a, b) + d(b, c) \leq \mathcal{D}_t^{\text{upp}}(a, b) + \mathcal{D}_t^{\text{upp}}(b, c).$$

In other words, $\mathcal{D}_t^{\text{upp}}(a, b) + \mathcal{D}_t^{\text{upp}}(b, c)$ is an upper bound for $d(a, c)$. Therefore, it must hold that

$$\mathcal{D}_t^{\text{upp}}(a, c) \leq \min\left\{d_{\max}, \mathcal{D}_t^{\text{upp}}(a, b) + \mathcal{D}_t^{\text{upp}}(b, c)\right\}. \tag{1}$$

Reverse triangle inequality The *reverse triangle inequality* states that $|d(a, b) - d(b, c)| \leq d(a, c)$ for all $a, b, c \in \{1, \ldots, M\}$. Now note that

$$|d(a, b) - d(b, c)| \geq \mathcal{D}_t^{\text{low}}(a, b) - \mathcal{D}_t^{\text{upp}}(b, c),$$
$$|d(a, b) - d(b, c)| \geq \mathcal{D}_t^{\text{low}}(b, c) - \mathcal{D}_t^{\text{upp}}(a, b).$$

Therefore, this gives a lower bound for (a, c). Thus,

$$\mathcal{D}_t^{\text{low}}(a, c) \geq \max\left\{0, \mathcal{D}_t^{\text{low}}(a, b) - \mathcal{D}_t^{\text{upp}}(b, c), \mathcal{D}_t^{\text{low}}(b, c) - \mathcal{D}_t^{\text{upp}}(a, b)\right\}. \tag{2}$$

Update rules In round t, we first set $\mathcal{D}_{t+1}^{\text{low}} := \mathcal{D}_t^{\text{low}}$, $\mathcal{D}_{t+1}^{\text{upp}} := \mathcal{D}_t^{\text{upp}}$. After the new distance $d(i, j)$ is given by the expert, the upper and lower bound collapse to $d(i, j)$, as it is assumed that the expert makes no mistakes. Thus,

$$\begin{aligned} \mathcal{D}_{t+1}^{\text{low}}(i, j) &:= d(i, j) =: \mathcal{D}_{t+1}^{\text{upp}}(i, j), \\ \mathcal{D}_{t+1}^{\text{low}}(j, i) &:= d(i, j) =: \mathcal{D}_{t+1}^{\text{upp}}(j, i). \end{aligned} \tag{U1}$$

This newly acquired information can have an effect on other bounds as well. For all $k \in \{1, \ldots, M\}$ (1) now gives the following *update rules*:

$$\begin{aligned} \mathcal{D}_{t+1}^{\text{upp}}(i, k) &:= \min\left\{d_{\max}, \mathcal{D}_{t+1}^{\text{upp}}(i, j) + \mathcal{D}_{t+1}^{\text{upp}}(j, k)\right\}, \\ \mathcal{D}_{t+1}^{\text{upp}}(j, k) &:= \min\left\{d_{\max}, \mathcal{D}_{t+1}^{\text{upp}}(i, j) + \mathcal{D}_{t+1}^{\text{upp}}(i, k)\right\}, \\ \mathcal{D}_{t+1}^{\text{upp}}(k, i) &:= \mathcal{D}_{t+1}^{\text{upp}}(i, k), \\ \mathcal{D}_{t+1}^{\text{upp}}(k, j) &:= \mathcal{D}_{t+1}^{\text{upp}}(j, k). \end{aligned} \tag{U2}$$

Note that this can lead to multiple updates, as $\mathcal{D}_{t+1}^{\text{upp}}(i, k)$ is updated in the first line and used in the second, whereas $\mathcal{D}_{t+1}^{\text{upp}}(j, k)$ is used in the first and updated in the second. For each bound that is now tighter than before, the same procedure should be repeated. Note that the order of the updates does not influence the end result as long as the effect of every tighter bound is evaluated.

Thereafter, lower bounds can be updated using (2). For all $k \in \{1, \ldots, M\}$, the updates are as follows:

$$\begin{aligned} \mathcal{D}_{t+1}^{\text{low}}(i, k) &:= \max\{0, \mathcal{D}_{t+1}^{\text{low}}(i, j) - \mathcal{D}_{t+1}^{\text{upp}}(j, k), \mathcal{D}_{t+1}^{\text{low}}(j, k) - \mathcal{D}_{t+1}^{\text{upp}}(i, j)\}, \\ \mathcal{D}_{t+1}^{\text{low}}(j, k) &:= \max\{0, \mathcal{D}_{t+1}^{\text{low}}(i, j) - \mathcal{D}_{t+1}^{\text{upp}}(i, k), \mathcal{D}_{t+1}^{\text{low}}(i, k) - \mathcal{D}_{t+1}^{\text{upp}}(i, j)\}, \\ \mathcal{D}_{t+1}^{\text{low}}(k, i) &:= \mathcal{D}_{t+1}^{\text{low}}(i, k), \\ \mathcal{D}_{t+1}^{\text{low}}(k, j) &:= \mathcal{D}_{t+1}^{\text{low}}(j, k). \end{aligned} \tag{U3}$$

Again, this can lead to multiple updates, similar to the upper bound updates. However, it is important to note that a new upper bound can lead to a new lower bound, but not vice versa. When an upper bound changes (e.g., $\mathcal{D}_{t+1}^{\text{upp}}(x, y)$), Update rules (U2) and (U3) should be evaluated (replacing (i, j) with (x, y)). Whenever a lower bound changes (e.g., $\mathcal{D}_{t+1}^{\text{low}}(x, y)$), only Update rules (U3) needs to be checked. The entire update procedure is summarized in Algorithm 2, that should be applied each time a new distance label is obtained from the expert.

5 Strategies

In this section, we discuss the selection strategies that will be evaluated. As the APDL problem is new, we will investigate relatively straightforward strategies based on naturally arising criteria to determine the baseline strategies for future research. Without previous literature, there is yet no evidence which strategies should perform well. However, we can argue e.g., that selecting indices, where the upper and lower bound are already close, is not a good idea. Thus, sometimes we investigate a strategy that maximizes a criterion, without looking into a strategy that minimizes the same criterion, or vice versa. On top of the general definition of a strategy (see Definition 2), it is necessary to introduce some concepts and definitions that are used by certain selection strategies.

Input: Distance $d(x, y)$, indices (x, y), round t
Output: Updated bounds $\mathcal{D}^{\text{upp}}_{t+1}, \mathcal{D}^{\text{low}}_{t+1}$

Initialization:
$\mathcal{D}^{\text{upp}}_{t+1} \leftarrow \mathcal{D}^{\text{upp}}_{t}, \mathcal{D}^{\text{low}}_{t+1} \leftarrow \mathcal{D}^{\text{low}}_{t}$
$\mathcal{D}^{\text{low}}_{t+1}(x, y) \leftarrow d(x, y), \mathcal{D}^{\text{low}}_{t+1}(y, x) \leftarrow d(x, y)$
$\mathcal{D}^{\text{upp}}_{t+1}(x, y) \leftarrow d(x, y), \mathcal{D}^{\text{upp}}_{t+1}(y, x) \leftarrow d(x, y)$
$U_{\text{update}} \leftarrow \{(x, y)\}, L_{\text{update}} \leftarrow \{(x, y)\}$

Update upper bounds:
while $U_{\text{update}} \neq \emptyset$ **do**
  Take $(i, j) \in U_{\text{update}}$
  for $k \leftarrow 1$ to M **do**
    Update $\mathcal{D}^{\text{upp}}_{t+1}$ with Update rules (U2)
  end for
  for every tighter bound **do**
    Add corresponding indices to U_{update} and L_{update} ▷ Every tighter upper bound could lead to other new upper or lower bounds
  end for
end while
Remove duplicates from L_{update}

Update lower bounds:
while $L_{\text{update}} \neq \emptyset$ **do**
  Take $(i, j) \in L_{\text{update}}$
  for $k \leftarrow 1$ to M **do**
    Update $\mathcal{D}^{\text{low}}_{t+1}$ with Update rules (U3)
  end for
  for every tighter bound **do**
    Add corresponding indices to L_{update} ▷ Every tighter lower bound could lead to other new lower bounds
  end for
end while

Algorithm 2 Update upper and lower bounds.

A selection strategy σ consists of functions σ_t for $t \in \{1, \ldots, T\}$ (see Definition 2). For all strategies that will be used, it holds that the same selection criterion is used for each σ_t. In other words, the strategy does not change for different rounds.

It is possible that multiple samples satisfy some selection criterion (for example, the *least chosen* strategy). If more than one sample is optimal for the selection criterion, a selection between these samples is made uniformly at random. The following notation is used for this.

Definition 3 (Drawn uniformly from set) Let $\mathcal{U}(A)$ denote the uniform distribution over a finite non-empty set A. Thus, when $X \sim \mathcal{U}(A)$ it must hold that $\mathbb{P}(X = a) = \frac{1}{|A|}$ for each $a \in A$.

Degree

It is also useful to track how often each index is chosen. Note that the problem can be visualized by a graph. Each sample is a vertex, and an edge is drawn between a pair of vertices, whenever the expert labels the distance between these pairs. How often each index is chosen is identical to the *degree* (from graph theory) of the corresponding vertex. Let $\deg_t(k)$ denote the *degree* of sample k in round t. This can be determined by

$$\deg_t(k) = |\{\zeta_\tau = (i, j) : i = k \vee j = k\}_{\tau=1,\ldots,t-1}|.$$

Predicted distance

Let $\mathcal{D}^{\text{pred}}_t(i, j)$ be the predicted distance between samples i and j in round t. We will later show (in Definition 4 below) how the distance is actually predicted. Strategies can use these predictions in a selection criterion.

Different kinds of strategies

Next, we divide the selection strategies into two groups, namely *simultaneous* and *sequential strategies*. Behind a *simultaneous strategy*, there is a singular selection criterion that determines which pair of indices is selected in round t out of all possible remaining pairs in

$$\mathcal{I}_t := \big\{(i, j) \in \{1, \ldots, M\}^2 : (\mathcal{H}_{\tau-1}) \notin \{(i, j), (j, i)\} \text{ for all } \tau \in \{1, \ldots, t-1\}\big\}.$$

For a *sequential strategy*, the indices are chosen one after the other by two (possibly different) selection criteria. To this end, if $\sigma_t(\mathcal{H}_{t-1}) = (i, j)$, let $\sigma_t(\mathcal{H}_{t-1})_1 := i$ and let $\sigma_t(\mathcal{H}_{t-1})_2 := j$ denote the first and second index respectively. $\sigma_t(\mathcal{H}_{t-1})_1$ is chosen from the remaining first indices, thus from

$$\mathcal{I}^{\text{uniq}}_{1,t} := \{i : \exists (i, \cdot) \in \mathcal{I}_t\}.$$

Whenever the first index is chosen, the remaining second indices reduce, as it is limited by the first chosen index $\sigma_t(\mathcal{H}_{t-1})_1$. The second index is chosen from

$$(\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq}} := \{j : \exists (\sigma_t(\mathcal{H}_{t-1})_1, j) \in \mathcal{I}_t\}.$$

5.1 Simultaneous strategies

First, we will discuss the *simultaneous strategies*, where both indices are chosen at the same time.

5.1.1 Random pair

Select a pair uniformly at random out of the remaining pairs.

Criterion 1 (Random pair)

$$\sigma_t(\mathcal{H}_{t-1}) \sim \mathcal{U}(\mathcal{I}_t). \tag{3}$$

5.1.2 Max bound gap

Select a pair uniformly at random out of the remaining pairs with the largest difference between the upper and lower bound of the predicted distance.

Criterion 2 (Max bound gap)

$$\sigma_t(\mathcal{H}_{t-1}) \sim \mathcal{U}\left(\underset{(i,j)\in\mathcal{I}_t}{\arg\max}\left\{\mathcal{D}_t^{\text{upp}}t(i,j) - \mathcal{D}_t^{\text{low}}t(i,j)\right\}\right). \tag{4}$$

5.1.3 Max combined total bound gap

First, determine for each sample the bound gap with all other samples and sum these into a combined bound gap. Then, select a pair uniformly at random out of the remaining pairs with the largest sum of combined bound gaps.

Criterion 3 (Max combined total bound gap)

$$\sigma_t(\mathcal{H}_{t-1}) \sim \mathcal{U}\Bigg(\underset{(i,j)\in\mathcal{I}_t}{\arg\max}\Bigg\{\sum_{k=1}^{M}\Big(\mathcal{D}_t^{\text{upp}}(i,k) - \mathcal{D}_t^{\text{low}}(i,k) + \mathcal{D}_t^{\text{upp}}(j,k) - \mathcal{D}_t^{\text{low}}(j,k)\Big)\Bigg\}\Bigg). \tag{5}$$

5.1.4 Max/min total degree

First, determine for each sample the degree, see Section 5. Then, select a pair uniformly at random out of all remaining pairs where the sum of the individual degrees is maximized/minimized.

Criterion 4 (Max total degree)

$$\sigma_t(\mathcal{H}_{t-1}) \sim \mathcal{U}\left(\underset{(i,j)\in\mathcal{I}_t}{\arg\max}\left\{\deg_t(i) + \deg_t(j)\right\}\right). \tag{6}$$

Criterion 5 (Min total degree)

$$\sigma_t(\mathcal{H}_{t-1}) \sim \mathcal{U}\left(\underset{(i,j)\in\mathcal{I}_t}{\arg\min}\left\{\deg_t(i) + \deg_t(j)\right\}\right). \tag{7}$$

5.2 Sequential strategies

Next, we will discuss the *sequential strategies*, where the second index is chosen after the first.

5.2.1 Random index

Draw uniformly at random an index out of the unique set of possible remaining indices.

Criterion 6 (Random index)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \mathcal{U}\left(\mathcal{I}_{1,t}^{\text{uniq.}}\right), \tag{8}$$

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left((\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}\right). \tag{9}$$

Note that choosing the first and second index using *random index* is not equivalent to using the *random pair* strategy, as *random index* uses the unique indices, where *random pair* does not.

5.2.2 Linked

This strategy can only be applied for the first index. Use the second index of the previous round as the first index of this round, unless there are no remaining pairs with this index. In this case and in the first round, choose the first index uniformly at random from the unique first indices, equivalent to the *random index* strategy, see (8).

Criterion 7 (Linked)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \begin{cases} \mathcal{U}\left(\sigma_{t-1}(\mathcal{H}_{t-2})_2\right) & \text{if } t > 1 \text{ and } \sigma_{t-1}(\mathcal{H}_{t-2})_2 \in \mathcal{I}_{1,t}^{\text{uniq.}} \\ \mathcal{U}\left(\mathcal{I}_{1,t}^{\text{uniq.}}\right) & \text{else.} \end{cases}, \tag{10}$$

5.2.3 Max/min degree

Choose uniformly at random an index with maximum degree (see Section 5) out of the unique set of possible remaining indices.

Criterion 8 (Max degree)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \mathcal{U}\left(\underset{i\in\mathcal{I}_{1,t}^{\text{uniq.}}}{\arg\max}\left\{\deg_t(i)\right\}\right), \tag{11}$$

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\max}\left\{\deg_t(j)\right\}\right). \tag{12}$$

Criterion 9 (Min degree)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \mathcal{U}\left(\underset{i\in\mathcal{I}_{1,t}^{\text{uniq.}}}{\arg\min}\left\{\deg_t(i)\right\}\right), \tag{13}$$

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\min}\left\{\deg_t(j)\right\}\right). \tag{14}$$

5.2.4 Max total bound gap

First, determine for each sample the bound gap with all other samples and sum these into a combined bound gap. Then, choose uniformly at random an index with maximum combined bound gap.

Criterion 10 (Max total bound gap)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \mathcal{U}\left(\underset{i\in\mathcal{I}_{1,t}^{\text{uniq.}}}{\arg\max}\left\{\sum_{k=1}^{M}\left(\mathcal{D}_t^{\text{upp}}(i,k)-\mathcal{D}_t^{\text{low}}(i,k)\right)\right\}\right), \tag{15}$$

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\max}\left\{\sum_{k=1}^{M}\left(\mathcal{D}_t^{\text{upp}}(j,k)-\mathcal{D}_t^{\text{low}}(j,k)\right)\right\}\right). \tag{16}$$

5.2.5 Max previous expected distance

In the first round, this strategy simplifies to the *random index* strategy (Section 5.2.1). Thereafter, choose uniformly at random an index out of the unique set of the possible remaining indices, such that the predicted distance to the indices of the previous round is maximized.

Criterion 11 (Max previous expected distance)

$$\sigma_t(\mathcal{H}_{t-1})_1 \sim \begin{cases} \mathcal{U}\Big(\arg\max_{i\in\mathcal{I}_t}\Big\{ \\ \quad \mathcal{D}_t^{\text{pred}}(\sigma_{t-1}(\mathcal{H}_{t-2})_1, i) & \text{if } t>1, \\ \quad + \mathcal{D}_t^{\text{pred}}(\sigma_{t-1}(\mathcal{H}_{t-2})_2, i)\Big\}\Big) \\ \hline \mathcal{U}(\mathcal{I}_t) & \text{else.} \end{cases} \tag{17}$$

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \begin{cases} \mathcal{U}\Big(\arg\max_{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}\Big\{ \\ \quad \mathcal{D}_t^{\text{pred}}(\sigma_{t-1}(\mathcal{H}_{t-2})_1, j) & \text{if } t>1, \\ \quad + \mathcal{D}_t^{\text{pred}}(\sigma_{t-1}(\mathcal{H}_{t-2})_2, j)\Big\}\Big) \\ \hline \mathcal{U}\left((\mathcal{I}_t \mid \sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}\right) & \text{else.} \end{cases} \tag{18}$$

5.2.6 Max/min/median expected distance

This strategy can only be applied for the second index. Select uniformly at random an index out of the unique set of remaining possible indices that belong to the maximum/minimum/median of the predicted distance (see Section 6.4) to the first index.

Criterion 12 (Max expected distance)

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\max}\left\{\mathcal{D}_t^{\text{pred}}(\sigma_t(\mathcal{H}_{t-1})_1, j)\right\}\right). \tag{19}$$

Criterion 13 (Min expected distance)

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\min}\left\{\mathcal{D}_t^{\text{pred}}(\sigma_t(\mathcal{H}_{t-1})_1, j)\right\}\right). \tag{20}$$

Criterion 14 (Median expected distance)

$$\sigma_t(\mathcal{H}_{t-1})_2 \sim \mathcal{U}\left(\underset{j\in(\mathcal{I}_t|\sigma_t(\mathcal{H}_{t-1})_1)^{\text{uniq.}}}{\arg\text{median}}\left\{\mathcal{D}_t^{\text{pred}}(\sigma_t(\mathcal{H}_{t-1})_1, j)\right\}\right). \tag{21}$$

6 Experimental setup

6.1 Strategies

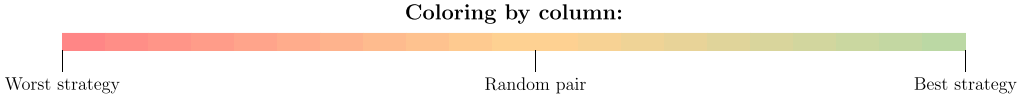
The goal of the experiments is to find which strategies perform well for which dataset. In Section 5, all used criteria are explained and defined. With *simultaneous* strategies, an index pair (i, j) is chosen at once. With *sequential* strategies, a separate decision is made for the first and second index sequentially. For example, one strategy uses Criterion 8 (max degree) to select the first index, and Criterion 9 (min degree) for the second index. In total, this leads to 5 (simultaneous) + $6 \cdot 8$ (sequential) $=$ 53 different strategies (see Table 1). Furthermore, all strategies are stochastic. Therefore, each strategy is repeated ten times for each dataset. Thereafter, results are averaged to reduce stochastic outliers. It is desirable that a strategy performs generally well, not only coincidentally.

6.2 Data

To evaluate the performance of different strategies, fourteen two-dimensional datasets are used. To reduce computational time, the maximum allowed size of a dataset is 1,000 samples. Whenever a dataset is larger, a subset of 1,000 samples is drawn uniformly at random. The coordinates are scaled (min-max) for each dataset to be within $[0, 1]^2$. The following datasets are used, where the number of samples is denoted in round brackets: *S1* (1,000), *S2* (1,000), *S3* (1,000), *S4* (1,000) [25], *Unbalance* (1,000) [26], *Birch2-1*

Table 1 Average performance: for each dataset and repetition, the prediction error of a strategy is averaged in rounds $i \cdot M_{MST}$ with $i \in \{1, \ldots, 10\}$. The ranking (by column) of each average prediction error is noted in brackets. Coloring of each column is done linearly between the worst and baseline (random pair) score and linearly between the baseline (random pair) and the best score

Strategy	1 M_{MST}	2 M_{MST}	3 M_{MST}	4 M_{MST}	5 M_{MST}	6 M_{MST}	7 M_{MST}	8 M_{MST}	9 M_{MST}	10 M_{MST}
max degree/max degree	0.040 (*02*)	0.024 (*07*)	0.017 (*10*)	0.012 (*10*)	0.009 (*09*)	0.006 (*09*)	0.005 (*12*)	0.004 (*12*)	0.003 (*13*)	0.003 (*13*)
max degree/min degree	0.041 (*09*)	0.024 (*09*)	0.016 (*07*)	0.011 (*07*)	0.008 (*06*)	0.006 (*06*)	0.005 (*08*)	0.004 (*18*)	0.003 (*15*)	0.003 (*17*)
max degree/max exp. dist.	0.040 (*03*)	0.022 (*04*)	0.013 (*04*)	0.009 (*04*)	0.007 (*04*)	0.005 (*04*)	0.004 (*05*)	0.003 (*11*)	0.002 (*11*)	0.002 (*11*)
max degree/max prev. exp. dist.	0.041 (*07*)	0.022 (*05*)	0.014 (*06*)	0.010 (*05*)	0.007 (*05*)	0.006 (*05*)	0.005 (*09*)	0.004 (*13*)	0.003 (*14*)	0.003 (*15*)
max degree/max total bound gap	0.041 (*08*)	0.022 (*03*)	0.013 (*02*)	0.008 (*02*)	0.005 (*02*)	0.004 (*03*)	0.003 (*03*)	0.002 (*10*)	0.002 (*09*)	0.001 (*09*)
max degree/median exp. dist.	0.042 (*11*)	0.023 (*06*)	0.014 (*05*)	0.010 (*06*)	0.008 (*07*)	0.006 (*07*)	0.005 (*11*)	0.004 (*16*)	0.003 (*17*)	0.003 (*16*)
max degree/min exp. dist.	**0.039** (*01*)	0.032 (*13*)	0.024 (*13*)	0.020 (*13*)	0.017 (*14*)	0.015 (*21*)	0.012 (*21*)	0.010 (*21*)	0.009 (*22*)	0.008 (*23*)
max degree/random index	0.042 (*10*)	0.025 (*12*)	0.017 (*09*)	0.012 (*09*)	0.009 (*11*)	0.007 (*12*)	0.005 (*17*)	0.005 (*19*)	0.004 (*18*)	0.003 (*19*)
linked/max degree	0.056 (*14*)	0.024 (*08*)	0.021 (*12*)	0.011 (*08*)	0.010 (*12*)	0.006 (*08*)	0.006 (*18*)	0.004 (*17*)	0.004 (*19*)	0.003 (*18*)
linked/min degree	0.067 (*44*)	0.054 (*29*)	0.045 (*30*)	0.035 (*30*)	0.026 (*31*)	0.019 (*30*)	0.015 (*29*)	0.012 (*29*)	0.010 (*28*)	0.008 (*29*)
linked/max exp. dist.	0.063 (*16*)	0.053 (*16*)	0.043 (*17*)	0.031 (*16*)	0.020 (*16*)	0.012 (*15*)	0.004 (*06*)	**0.001** (*03*)	**0.000** (*02*)	**0.000** (*03*)
linked/max prev. exp. dist.	0.063 (*21*)	0.053 (*19*)	0.044 (*24*)	0.033 (*25*)	0.022 (*20*)	0.013 (*18*)	0.005 (*14*)	0.002 (*06*)	0.001 (*06*)	**0.000** (*06*)
linked/max total bound gap	0.067 (*42*)	0.056 (*36*)	0.050 (*42*)	0.041 (*39*)	0.032 (*39*)	0.024 (*35*)	0.018 (*33*)	0.014 (*33*)	0.011 (*31*)	0.009 (*31*)
linked/median exp. dist.	0.063 (*18*)	0.058 (*48*)	0.050 (*39*)	0.042 (*44*)	0.036 (*48*)	0.033 (*48*)	0.030 (*48*)	0.029 (*48*)	0.028 (*48*)	0.027 (*48*)
linked/min exp. dist.	0.066 (*29*)	0.065 (*49*)	0.063 (*49*)	0.061 (*49*)	0.060 (*49*)	0.059 (*49*)	0.058 (*49*)	0.057 (*49*)	0.057 (*49*)	0.056 (*49*)
linked/random index	0.062 (*15*)	0.053 (*18*)	0.042 (*16*)	0.033 (*20*)	0.025 (*26*)	0.019 (*24*)	0.015 (*27*)	0.012 (*30*)	0.010 (*30*)	0.008 (*30*)
min degree/max degree	0.040 (*04*)	0.025 (*11*)	0.018 (*11*)	0.012 (*12*)	0.009 (*10*)	0.007 (*11*)	0.005 (*15*)	0.004 (*15*)	0.003 (*12*)	0.003 (*12*)
min degree/min degree	0.067 (*43*)	0.054 (*27*)	0.045 (*29*)	0.034 (*29*)	0.026 (*29*)	0.019 (*29*)	0.015 (*26*)	0.012 (*27*)	0.010 (*29*)	0.008 (*27*)
min degree/max exp. dist.	0.066 (*35*)	0.055 (*33*)	0.044 (*23*)	0.031 (*17*)	0.021 (*17*)	0.011 (*14*)	0.004 (*07*)	**0.001** (*05*)	0.001 (*07*)	**0.000** (*07*)
min degree/max prev. exp. dist.	0.066 (*33*)	0.056 (*40*)	0.048 (*35*)	0.038 (*35*)	0.030 (*34*)	0.024 (*36*)	0.019 (*40*)	0.016 (*42*)	0.013 (*43*)	0.011 (*43*)
min degree/max total bound gap	0.067 (*47*)	0.057 (*42*)	0.050 (*43*)	0.042 (*43*)	0.033 (*42*)	0.026 (*45*)	0.020 (*43*)	0.016 (*41*)	0.013 (*41*)	0.010 (*39*)
min degree/median exp. dist.	0.066 (*31*)	0.054 (*22*)	0.044 (*21*)	0.033 (*23*)	0.025 (*23*)	0.019 (*26*)	0.016 (*30*)	0.013 (*31*)	0.012 (*38*)	0.010 (*41*)
min degree/min exp. dist.	0.067 (*53*)	0.067 (*53*)	0.066 (*53*)	0.066 (*53*)	0.066 (*53*)	0.065 (*53*)	0.065 (*53*)	0.065 (*53*)	0.064 (*53*)	0.064 (*53*)
min degree/random index	0.066 (*30*)	0.054 (*24*)	0.045 (*27*)	0.034 (*28*)	0.026 (*30*)	0.020 (*31*)	0.015 (*28*)	0.012 (*28*)	0.010 (*25*)	0.008 (*25*)
max prev. exp. dist./max degree	0.041 (*05*)	0.022 (*02*)	0.013 (*03*)	0.008 (*03*)	0.005 (*03*)	0.004 (*02*)	0.003 (*02*)	0.002 (*09*)	0.002 (*10*)	0.002 (*10*)
max prev. exp. dist./min degree	0.066 (*34*)	0.056 (*41*)	0.048 (*34*)	0.038 (*36*)	0.031 (*35*)	0.024 (*38*)	0.020 (*41*)	0.016 (*44*)	0.013 (*44*)	0.011 (*44*)
max prev. exp. dist./max exp. dist.	0.064 (*27*)	0.054 (*25*)	0.045 (*25*)	0.033 (*24*)	0.022 (*19*)	0.013 (*19*)	0.005 (*13*)	**0.001** (*04*)	**0.000** (*03*)	**0.000** (*02*)
max prev. exp. dist./max prev. exp. dist.	0.064 (*28*)	0.056 (*34*)	0.048 (*36*)	0.039 (*37*)	0.031 (*36*)	0.023 (*32*)	0.016 (*32*)	0.010 (*22*)	0.007 (*21*)	0.005 (*20*)
max prev. exp. dist./max total bound gap	0.067 (*39*)	0.058 (*46*)	0.052 (*48*)	0.043 (*47*)	0.035 (*46*)	0.027 (*46*)	0.021 (*46*)	0.016 (*45*)	0.013 (*45*)	0.011 (*45*)
max prev. exp. dist./median exp. dist.	0.064 (*26*)	0.055 (*31*)	0.047 (*31*)	0.038 (*34*)	0.031 (*37*)	0.025 (*43*)	0.022 (*47*)	0.019 (*47*)	0.017 (*47*)	0.015 (*47*)
max prev. exp. dist./min exp. dist.	0.067 (*51*)	0.067 (*52*)	0.066 (*52*)	0.066 (*52*)	0.065 (*52*)	0.065 (*52*)	0.064 (*51*)	0.064 (*51*)	0.064 (*51*)	0.063 (*51*)
max prev. exp. dist./random index	0.064 (*23*)	0.055 (*30*)	0.047 (*33*)	0.038 (*33*)	0.030 (*33*)	0.023 (*34*)	0.018 (*35*)	0.014 (*36*)	0.011 (*34*)	0.009 (*34*)
max total bound gap/max degree	0.041 (*06*)	**0.017** (*01*)	**0.008** (*01*)	**0.005** (*01*)	**0.004** (*01*)	**0.003** (*01*)	**0.002** (*01*)	0.002 (*08*)	0.002 (*08*)	0.001 (*08*)
max total bound gap/min degree	0.067 (*46*)	0.057 (*43*)	0.050 (*44*)	0.042 (*42*)	0.033 (*43*)	0.025 (*44*)	0.020 (*42*)	0.016 (*40*)	0.013 (*40*)	0.010 (*38*)
max total bound gap/max exp. dist.	0.067 (*38*)	0.056 (*35*)	0.049 (*37*)	0.037 (*31*)	0.024 (*21*)	0.014 (*20*)	0.006 (*19*)	0.002 (*07*)	0.001 (*05*)	**0.000** (*05*)
max total bound gap/max prev. exp. dist.	0.067 (*41*)	0.058 (*45*)	0.052 (*47*)	0.043 (*48*)	0.035 (*47*)	0.027 (*47*)	0.020 (*45*)	0.016 (*43*)	0.013 (*42*)	0.010 (*40*)
max total bound gap/max total bound gap	0.067 (*49*)	0.057 (*44*)	0.051 (*45*)	0.043 (*45*)	0.033 (*44*)	0.025 (*41*)	0.019 (*38*)	0.015 (*38*)	0.011 (*36*)	0.009 (*36*)
max total bound gap/median exp. dist.	0.067 (*36*)	0.056 (*37*)	0.050 (*38*)	0.041 (*38*)	0.032 (*40*)	0.025 (*40*)	0.020 (*44*)	0.017 (*46*)	0.015 (*46*)	0.014 (*46*)
max total bound gap/min exp. dist.	0.067 (*48*)	0.066 (*50*)	0.066 (*50*)	0.066 (*50*)	0.065 (*50*)	0.065 (*50*)	0.064 (*50*)	0.064 (*50*)	0.063 (*50*)	0.063 (*50*)
max total bound gap/random index	0.067 (*40*)	0.056 (*38*)	0.050 (*41*)	0.041 (*40*)	0.032 (*38*)	0.024 (*37*)	0.018 (*36*)	0.014 (*35*)	0.011 (*32*)	0.009 (*32*)
random index/max degree	0.055 (*13*)	0.041 (*14*)	0.029 (*14*)	0.021 (*14*)	0.016 (*13*)	0.012 (*16*)	0.009 (*20*)	0.007 (*20*)	0.006 (*20*)	0.005 (*21*)
random index/min degree	0.066 (*32*)	0.054 (*23*)	0.045 (*26*)	0.034 (*26*)	0.025 (*27*)	0.019 (*25*)	0.015 (*24*)	0.012 (*23*)	0.009 (*23*)	0.008 (*22*)
random index/max exp. dist.	0.064 (*22*)	0.053 (*20*)	0.042 (*15*)	0.030 (*15*)	0.019 (*15*)	0.010 (*13*)	0.003 (*04*)	**0.001** (*01*)	**0.000** (*04*)	**0.000** (*04*)
random index/max prev. exp. dist.	0.064 (*24*)	0.055 (*32*)	0.047 (*32*)	0.037 (*32*)	0.029 (*32*)	0.023 (*33*)	0.018 (*34*)	0.014 (*34*)	0.011 (*33*)	0.009 (*33*)
random index/max total bound gap	0.067 (*37*)	0.056 (*39*)	0.050 (*40*)	0.041 (*41*)	0.032 (*41*)	0.024 (*39*)	0.019 (*37*)	0.014 (*37*)	0.011 (*35*)	0.009 (*35*)
random index/median exp. dist.	0.063 (*19*)	0.053 (*21*)	0.043 (*19*)	0.032 (*18*)	0.025 (*22*)	0.019 (*27*)	0.016 (*31*)	0.013 (*32*)	0.012 (*39*)	0.010 (*42*)
random index/min exp. dist.	0.067 (*52*)	0.067 (*51*)	0.066 (*51*)	0.066 (*51*)	0.065 (*51*)	0.065 (*51*)	0.064 (*52*)	0.064 (*52*)	0.064 (*52*)	0.063 (*52*)
random index/random index	0.063 (*20*)	0.053 (*15*)	0.043 (*18*)	0.033 (*21*)	0.025 (*24*)	0.019 (*23*)	0.015 (*23*)	0.012 (*24*)	0.010 (*27*)	0.008 (*28*)
max bound gap	0.064 (*25*)	0.054 (*26*)	0.044 (*22*)	0.032 (*19*)	0.021 (*18*)	0.012 (*17*)	0.005 (*10*)	**0.001** (*02*)	**0.000** (*01*)	**0.000** (*01*)
max total degree	0.043 (*12*)	0.025 (*10*)	0.016 (*08*)	0.012 (*11*)	0.009 (*08*)	0.006 (*10*)	0.005 (*16*)	0.004 (*14*)	0.003 (*16*)	0.003 (*14*)
min total degree	0.067 (*45*)	0.054 (*28*)	0.045 (*28*)	0.034 (*27*)	0.026 (*28*)	0.019 (*28*)	0.015 (*25*)	0.012 (*26*)	0.010 (*26*)	0.008 (*24*)
max combined total bound gap	0.067 (*50*)	0.058 (*47*)	0.051 (*46*)	0.043 (*46*)	0.034 (*45*)	0.025 (*42*)	0.019 (*39*)	0.015 (*39*)	0.011 (*37*)	0.009 (*37*)
random pair	0.063 (*17*)	0.053 (*17*)	0.043 (*20*)	0.033 (*22*)	0.025 (*25*)	0.019 (*22*)	0.015 (*22*)	0.012 (*25*)	0.010 (*24*)	0.008 (*26*)



(1,000) [27], *Aggregation* (788) [28], *Compound* (399) [29], *Pathbased* (300), *Spiral* (312) [30], *D31* (1,000), *R15* (600) [31], *Jain* (373) [32], *Flame* (240) [33]. All these datasets are used as clustering benchmarks [34]. A visualizathon of these datasets can be seen in Fig. 1. The *Euclidean distance* is used as underlying distance metric for each dataset.

Observe that these datasets are all two-dimensional. In other words, they have two features. Note that this is not

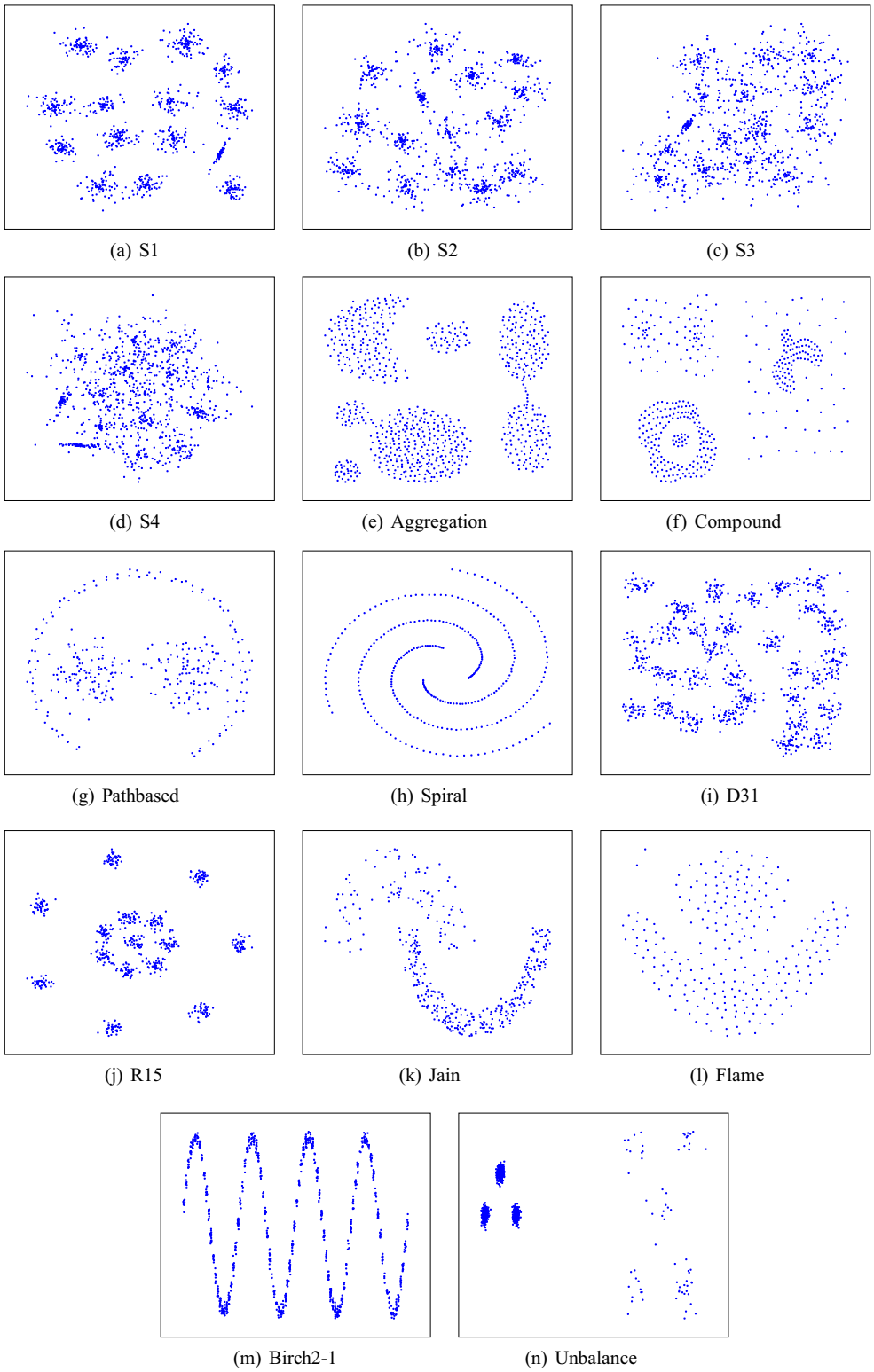


Fig. 1 Visualization datasets: Each two-dimensional dataset that is used to test different strategies.

a shortcoming for this experiment, as it is assumed that features are not relevant for the APDL techniques (see Section 2.2 above). As long as the calculated pairwise distances remain the same, these datasets could have any dimension. Two-dimensional datasets were chosen, because they can be visualized easily.

6.3 Number of rounds

The number of samples M is dependent on the dataset. Especially for increasingly large datasets, it is undesirable to keep on labeling until all labels are given. Namely, $\binom{M}{2} = M \cdot (M-1)/2$ pairwise combinations can be made in total. If e.g., ten percent of the combinations should be labeled, the total number of rounds T grows exponentially in the number of samples. This gives much more opportunities to determine good upper and lower bound approximations for a large dataset compared to a small dataset. Therefore, we decide to choose the total number of rounds for a dataset in a linear-growing fashion. A *minimum spanning tree* (MST) in graph theory is a subset of edges in an undirected graph, such that all vertices are connected without any cycles. In total, $M-1$ edges are necessary to make an MST for a graph with M vertices. For each $M-1$ labels given by the expert, a minimum spanning tree could have been formed. Now, let $M_{MST} := M-1$ and define the total number of rounds T as $10 \cdot M_{MST}$. This reflects a scenario where it is not possible to determine many labels, which will often be the case in practice.

6.4 Performance evaluation of strategies

In order to compare the different strategies, it is important to discuss how the performance of the strategies is evaluated. Each strategy is applied ten times on each dataset. Each round a prediction is made by averaging the upper and lower bound.

Definition 4 (Predicted distance matrix) Let $\mathcal{D}_t^{\text{pred}}$ be the predicted distance matrix in round t, such that

$$\mathcal{D}_t^{\text{pred}}(i,j) := \left(\mathcal{D}_t^{\text{upp}}(i,j) + \mathcal{D}_t^{\text{low}}(i,j)\right)/2.$$

Note that if (i,j) was labeled by the expert, it holds that

$$\begin{aligned}\mathcal{D}_t^{\text{pred}}(i,j) &= \left(\mathcal{D}_t^{\text{upp}}(i,j) + \mathcal{D}_t^{\text{low}}(i,j)\right)/2 = \left(d(i,j) + d(i,j)\right)/2 \\ &= d(i,j).\end{aligned}$$

Definition 5 (True distance matrix) Let $\mathcal{D}^{\text{true}}$ be the true distance matrix.

The *prediction error* between the predicted distance matrix $\mathcal{D}_t^{\text{pred}}$ and the true distance matrix $\mathcal{D}^{\text{true}}$ can now be calculated. To compare these two matrices, the *mean squared error* is used. This leads to the following definition.

Definition 6 (Prediction error) The error ϵ_t in round t is determined as

$$\epsilon_t = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(\mathcal{D}^{\text{true}}(i,j) - \mathcal{D}_t^{\text{pred}}(i,j)\right)^2.$$

After collecting all prediction error results, three approaches are undertaken to compare the performance of each strategy: (I) *average performance*, (II) *Borda count*; (III) *area under the curve* (AUC). Each approach will now be explained.

6.4.1 Average performance

To average the prediction error results over different datasets, the error is determined at predefined rounds, specific for each dataset. As discussed in Section 6.3, the total number of rounds is dependent on the size of the dataset. Thus, in round $i \cdot M_{MST}$ with $i \in \{1, \ldots, 10\}$, the prediction error is determined. Averaging the results for a fixed i produces the final score. Summarizing, all prediction errors of a single strategy at predefined rounds are averaged for all ten repetitions and all fourteen datasets.

6.4.2 Borda count

A drawback of the previous approach is that certain datasets might be harder to predict correctly, making these datasets influence the average performance heavily, as the prediction error is relatively large, and all datasets are weighted equally. Thus, *Borda count* [35] (a voting method) is used to rank the prediction error of each strategy in the following way. First, order all strategies based on the prediction error for each dataset and repetition. The strategy with the highest prediction error gets 1 point. The second worst gets 2 points. The third highest gets 3 points and so on. This is done for each dataset and repetition in the predefined rounds $\{i \cdot M_{MST}\}_{i=1,\ldots,10}$. The final *Borda count* results are obtained by averaging over all datasets and repetitions for a fixed round. A higher score indicates better performance, and the maximum possible score is equal to the total number of strategies.

6.4.3 Area under the curve

Instead of comparing the results at specified iterations, it is also possible to evaluate the performance of a strategy by measuring the so-called *area under the curve* (AUC) for each iteration using the trapezoidal rule. For each strategy, dataset and repetition, the area under the prediction error is measured up to and including the maximum number of

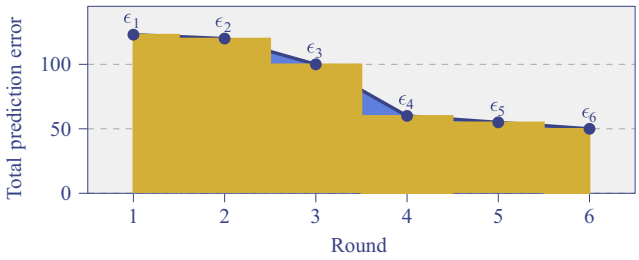


Fig. 2 Example AUC: Each round the prediction error is measured. The AUC of the prediction error is then determined by using the trapezoidal rule (22), which adds the area of the golden rectangles. Note that this is exactly equal to the blue area under the prediction error curve

rounds ($10 \cdot M_{MST}$). As the rounds are equally spaced, AUC reduces to

$$\sum_{t=2}^{10 \cdot M_{MST}-1} \epsilon_t + \frac{\epsilon_1 + \epsilon_{10 \cdot M_{MST}}}{2}. \tag{22}$$

Note that the AUC is not necessarily bounded by [0,1]. By averaging over the repetitions, an average AUC score can be derived for each strategy and dataset. A lower score indicates better performance, as the prediction error must be minimized and the sooner this is achieved the better. A fictitious example of how the AUC is measured can be seen in Fig. 2.

7 Results

Following the experimental setup from Section 6, all 53 strategies outlined in Section 5 are evaluated on fourteen different datasets (see Section 6.2). The results are summarized into three tables: Section 1 gives the average prediction error results (Section 6.4.1); Table 2 shows the average *Borda count* score for each strategy (Section 6.4.2); Table 3 displays the *area under the curve* results for each strategy and dataset (Section 6.4.3). These tables all provide a different angle on the performance of the selection strategies.

Next, we will discuss the most important observations backed by evidence from Tables 1, 2 and 3.

Observation 1 There are better strategies than simply choosing a *random pair*.

Evidence: The best rank *random pair* achieves is 13^{th} in Table 3 on the dataset *Unbalance*. Often it ranks around the mid-twenties in Tables 1, 2 and 3. This means that there are (many) strategies that perform better than *random pair*.

Observation 2 Max *total bound gap* / max *degree* is the best strategy for earlier rounds.

Evidence: The ranked scores of strategy max *total bound gap* / max *degree* are highlighted in Table 4. For rounds $2 \cdot M_{MST}$ up to $7 \cdot M_{MST}$, the strategy ranks the best out of all evaluated strategies. When one has really limited labeling capabilities, this strategy performs very well across all datasets. It always has the best *AUC* score out of all tested strategies, except for the dataset *Unbalance* (see Table 3).

Observation 3 Max *degree* is generally a good criterion, especially in the earlier rounds.

Evidence: In Tables 1, 2 and 3 a lot of green cells belong to a strategy with max *degree*. This means that it performs close to or equal to the best performance. Thus, it is a good strategy to choose at least one of the indices based on max *degree*. Especially in the earlier rounds. In round $3 \cdot M_{MST}$, strategies with max *degree* rank in Table 1: (10^{th}, 07^{th}, 04^{th}, 06^{th}, 02^{nd}, 05^{th}, 13^{th}, 09^{th}, 12^{th}, 11^{th}, 03^{rd}, 01^{st}, 14^{th}). Thus, the entire top 14 is filled by strategies with max *degree* except for the eight place, which is obtained by max *total degree*. This criterion is thus highly effective in the earlier rounds.

Observation 4 Min exp. *distance* and *median* exp. *distance* are bad criteria.

Evidence: Both min exp. *distance* and *median* exp. *distance* perform terrible. After $10 \cdot M_{MST}$, strategies with min exp. *distance* and with *median* exp. *distance* are ranked (23^{rd}, 49^{th}, 53^{rd}, 51^{st}, 50^{th}, 52^{nd}) and (16^{th}, 48^{th}, 41^{st}, 47^{th}, 46^{th}, 42^{nd}), respectively in Table 1. Only combining with max *degree* can save the performance. Min exp. *distance* is for all other combinations colored red in Tables 1, 2 and 3, which means that it is (or close to) the worst performance.

Observation 5 Although the prediction is directly dependent on the bound gap, max *bound gap* is only a good strategy after $> 7 \cdot M_{MST}$ rounds.

Evidence: The ranked scores of strategy max *bound gap* are highlighted in Table 5. In the early rounds (up to $4 \cdot M_{MST}$), this strategy performs even worse than *random pair*. After that, it quickly becomes one of the best performing strategies, even ranking first in the later rounds. Due to the slow start, the *AUC* scores are remarkably mediocre, see Table 3.

Observation 6 Max exp. *distance* is a late bloomer.

Evidence: Whilst min exp. *distance* and *median* exp. *distance* perform bad, max exp. *distance* gets increasingly

Table 2 Borda count: for each dataset and repetition, Borda count is used to rank the prediction error of the strategies and averaged in rounds $i \cdot M_{MST}$ with $i \in \{1, \ldots, 10\}$. The ranking (by column) of each Borda count score is noted in brackets. Coloring of each column is done linearly between the worst and baseline (random pair) score and linearly between the baseline (random pair) and the best score

Strategy	1 M_{MST}	2 M_{MST}	3 M_{MST}	4 M_{MST}	5 M_{MST}	6 M_{MST}	7 M_{MST}	8 M_{MST}	9 M_{MST}	10 M_{MST}
max degree/max degree	43.46 (*02*)	46.62 (*06*)	46.31 (*07*)	45.19 (*08*)	43.91 (*08*)	42.75 (*09*)	40.91 (*14*)	38.84 (*14*)	38.37 (*14*)	38.11 (*14*)
max degree/min degree	40.66 (*13*)	46.34 (*07*)	46.10 (*10*)	45.59 (*06*)	45.17 (*05*)	43.00 (*06*)	40.57 (*15*)	38.19 (*18*)	37.83 (*15*)	37.11 (*19*)
max degree/max exp. dist.	**43.51** (*01*)	47.64 (*02*)	48.22 (*02*)	47.34 (*04*)	46.90 (*04*)	45.37 (*04*)	43.78 (*09*)	41.74 (*11*)	41.36 (*11*)	41.06 (*11*)
max degree/max prev. exp. dist.	40.16 (*14*)	46.31 (*08*)	47.04 (*05*)	45.94 (*05*)	44.96 (*06*)	42.97 (*07*)	41.30 (*12*)	39.01 (*12*)	38.79 (*12*)	38.28 (*13*)
max degree/max total bound gap	42.29 (*11*)	47.13 (*04*)	47.68 (*04*)	47.52 (*03*)	48.19 (*03*)	47.06 (*03*)	45.74 (*06*)	43.69 (*10*)	43.11 (*09*)	43.26 (*09*)
max degree/median exp. dist.	42.34 (*09*)	46.69 (*05*)	46.94 (*06*)	45.37 (*07*)	43.93 (*07*)	42.90 (*08*)	40.94 (*13*)	38.58 (*16*)	37.78 (*16*)	37.99 (*15*)
max degree/min exp. dist.	43.24 (*03*)	41.34 (*13*)	40.09 (*13*)	37.11 (*15*)	34.78 (*19*)	31.71 (*21*)	29.17 (*21*)	27.84 (*21*)	27.58 (*22*)	26.54 (*26*)
max degree/random index	42.99 (*07*)	45.83 (*12*)	46.21 (*08*)	44.57 (*11*)	43.78 (*09*)	41.44 (*14*)	39.87 (*18*)	37.56 (*19*)	37.66 (*18*)	37.47 (*18*)
linked/max degree	42.58 (*08*)	46.14 (*09*)	43.04 (*12*)	44.79 (*10*)	41.71 (*12*)	42.57 (*10*)	39.32 (*19*)	38.39 (*17*)	37.01 (*19*)	37.48 (*17*)
linked/min degree	13.89 (*45*)	20.79 (*34*)	24.35 (*34*)	27.42 (*31*)	27.72 (*31*)	27.12 (*29*)	26.93 (*28*)	27.07 (*25*)	26.97 (*26*)	26.51 (*27*)
linked/max exp. dist.	38.21 (*16*)	35.71 (*16*)	34.23 (*16*)	35.76 (*16*)	38.23 (*14*)	42.17 (*11*)	47.11 (*03*)	50.71 (*02*)	51.16 (*03*)	51.29 (*03*)
linked/max prev. exp. dist.	37.60 (*18*)	34.09 (*18*)	29.05 (*23*)	30.17 (*24*)	34.40 (*20*)	39.13 (*18*)	43.35 (*10*)	46.46 (*06*)	47.74 (*06*)	47.81 (*06*)
linked/max total bound gap	14.34 (*41*)	12.41 (*44*)	11.38 (*44*)	12.71 (*42*)	15.25 (*39*)	17.66 (*35*)	19.81 (*33*)	21.41 (*31*)	22.93 (*31*)	24.12 (*31*)
linked/median exp. dist.	37.86 (*17*)	34.88 (*17*)	33.81 (*19*)	28.28 (*27*)	22.55 (*32*)	17.09 (*38*)	11.54 (*47*)	07.94 (*48*)	06.32 (*48*)	06.14 (*48*)
linked/min exp. dist.	18.53 (*35*)	11.21 (*47*)	07.69 (*49*)	05.49 (*49*)	04.35 (*49*)	03.73 (*49*)	03.59 (*50*)	03.49 (*50*)	03.39 (*50*)	03.38 (*51*)
linked/random index	38.90 (*15*)	36.02 (*15*)	35.28 (*15*)	32.00 (*20*)	29.36 (*26*)	27.19 (*28*)	26.11 (*29*)	26.05 (*29*)	25.69 (*30*)	25.42 (*30*)
min degree/max degree	43.01 (*06*)	45.94 (*11*)	45.27 (*11*)	44.14 (*12*)	43.70 (*11*)	41.82 (*13*)	40.36 (*17*)	38.96 (*13*)	38.56 (*13*)	38.40 (*12*)
min degree/min degree	14.14 (*42*)	21.27 (*33*)	24.84 (*31*)	27.71 (*29*)	27.88 (*29*)	27.81 (*26*)	27.34 (*24*)	27.03 (*26*)	26.72 (*27*)	26.82 (*25*)
min degree/max exp. dist.	21.81 (*29*)	23.04 (*31*)	27.22 (*26*)	33.44 (*17*)	36.76 (*15*)	39.98 (*17*)	44.06 (*08*)	46.84 (*05*)	46.91 (*07*)	46.43 (*07*)
min degree/max prev. exp. dist.	20.96 (*34*)	20.28 (*37*)	20.19 (*37*)	19.99 (*36*)	18.81 (*35*)	17.20 (*37*)	16.35 (*39*)	16.24 (*41*)	16.46 (*41*)	16.61 (*39*)
min degree/max total bound gap	13.29 (*47*)	12.62 (*43*)	12.29 (*43*)	12.34 (*44*)	12.50 (*43*)	12.94 (*45*)	13.34 (*42*)	14.14 (*43*)	14.99 (*43*)	15.60 (*41*)
min degree/median exp. dist.	21.01 (*33*)	23.68 (*29*)	28.20 (*24*)	31.29 (*22*)	29.77 (*24*)	26.88 (*30*)	23.87 (*30*)	20.34 (*32*)	17.09 (*39*)	14.72 (*42*)
min degree/min exp. dist.	05.39 (*53*)	01.86 (*53*)	01.61 (*53*)	01.51 (*53*)	01.49 (*53*)	01.40 (*53*)	01.43 (*53*)	01.44 (*53*)	01.45 (*53*)	01.47 (*53*)
min degree/random index	21.49 (*30*)	23.05 (*30*)	25.66 (*29*)	27.71 (*29*)	27.74 (*30*)	27.25 (*27*)	27.19 (*26*)	27.28 (*24*)	27.21 (*25*)	26.90 (*23*)
max prev. exp. dist./max degree	43.21 (*04*)	47.47 (*03*)	47.84 (*03*)	48.09 (*02*)	48.55 (*02*)	47.88 (*02*)	46.40 (*04*)	43.76 (*09*)	42.84 (*10*)	42.66 (*10*)
max prev. exp. dist./min degree	21.46 (*31*)	20.29 (*36*)	20.30 (*35*)	19.91 (*37*)	18.75 (*37*)	17.57 (*36*)	17.04 (*38*)	17.01 (*40*)	17.38 (*38*)	17.51 (*38*)
max prev. exp. dist./max exp. dist.	31.54 (*24*)	29.64 (*24*)	27.73 (*25*)	29.32 (*25*)	34.91 (*18*)	40.27 (*16*)	45.31 (*07*)	49.99 (*04*)	51.44 (*02*)	51.74 (*02*)
max prev. exp. dist./max prev. exp. dist.	30.52 (*27*)	26.53 (*28*)	20.22 (*36*)	17.09 (*38*)	16.27 (*38*)	17.99 (*33*)	22.00 (*32*)	26.03 (*30*)	28.09 (*21*)	29.36 (*21*)
max prev. exp. dist./max total bound gap	14.62 (*39*)	11.40 (*45*)	10.76 (*46*)	10.26 (*46*)	10.16 (*48*)	10.49 (*48*)	11.24 (*48*)	11.81 (*45*)	12.46 (*45*)	12.64 (*45*)
max prev. exp. dist./median exp. dist.	30.02 (*28*)	28.54 (*26*)	25.76 (*28*)	22.29 (*32*)	18.78 (*36*)	15.84 (*41*)	12.90 (*44*)	10.39 (*46*)	08.44 (*47*)	07.79 (*47*)
max prev. exp. dist./min exp. dist.	07.80 (*51*)	02.94 (*52*)	02.90 (*52*)	02.96 (*52*)	03.06 (*52*)	03.31 (*51*)	03.30 (*51*)	03.38 (*51*)	03.51 (*49*)	03.50 (*49*)
max prev. exp. dist./random index	31.28 (*25*)	28.76 (*25*)	24.59 (*32*)	20.96 (*34*)	18.87 (*34*)	17.81 (*34*)	17.79 (*36*)	18.49 (*37*)	19.06 (*37*)	20.40 (*37*)
max total bound gap/max degree	42.34 (*09*)	**49.59** (*01*)	**51.37** (*01*)	**51.10** (*01*)	**50.51** (*01*)	**49.70** (*01*)	47.79 (*02*)	45.59 (*08*)	44.06 (*08*)	43.91 (*08*)
max total bound gap/min degree	13.41 (*46*)	12.66 (*42*)	12.32 (*42*)	12.53 (*43*)	12.35 (*44*)	12.71 (*46*)	13.31 (*43*)	14.29 (*42*)	15.00 (*42*)	15.78 (*40*)
max total bound gap/max exp. dist.	14.11 (*43*)	13.06 (*39*)	14.75 (*38*)	21.00 (*33*)	30.49 (*21*)	37.36 (*19*)	42.57 (*11*)	46.46 (*06*)	48.69 (*05*)	49.04 (*05*)
max total bound gap/max prev. exp. dist.	14.58 (*40*)	11.34 (*46*)	11.01 (*45*)	10.49 (*45*)	10.36 (*47*)	10.91 (*47*)	11.58 (*46*)	12.45 (*44*)	13.26 (*44*)	13.55 (*44*)
max total bound gap/max total bound gap	09.29 (*49*)	07.44 (*48*)	08.31 (*47*)	09.25 (*47*)	11.72 (*45*)	13.91 (*42*)	15.70 (*41*)	17.39 (*39*)	19.23 (*35*)	20.52 (*35*)
max total bound gap/median exp. dist.	16.00 (*36*)	13.35 (*38*)	14.35 (*39*)	14.64 (*39*)	14.33 (*41*)	13.51 (*44*)	11.68 (*45*)	09.90 (*47*)	08.80 (*46*)	08.23 (*46*)
max total bound gap/min exp. dist.	12.62 (*48*)	04.34 (*50*)	04.16 (*50*)	03.98 (*50*)	03.84 (*50*)	03.72 (*50*)	03.61 (*49*)	03.51 (*49*)	03.34 (*52*)	03.21 (*52*)
max total bound gap/random index	15.07 (*38*)	13.03 (*40*)	12.48 (*40*)	13.14 (*40*)	14.75 (*40*)	17.09 (*39*)	18.56 (*35*)	19.94 (*34*)	21.00 (*32*)	21.87 (*32*)
random index/max degree	41.01 (*12*)	38.94 (*14*)	39.03 (*14*)	37.55 (*14*)	36.11 (*16*)	35.06 (*20*)	33.71 (*20*)	33.06 (*20*)	33.14 (*20*)	33.04 (*20*)
random index/min degree	21.44 (*32*)	23.00 (*32*)	26.15 (*27*)	28.52 (*26*)	28.31 (*28*)	27.96 (*23*)	27.30 (*25*)	27.46 (*22*)	27.58 (*22*)	27.12 (*22*)
random index/max exp. dist.	31.72 (*21*)	32.44 (*19*)	34.10 (*17*)	37.59 (*13*)	40.10 (*13*)	43.56 (*05*)	**47.88** (*01*)	50.64 (*03*)	50.46 (*04*)	49.96 (*04*)
random index/max prev. exp. dist.	30.94 (*26*)	28.54 (*26*)	24.94 (*30*)	20.72 (*35*)	19.06 (*33*)	18.68 (*32*)	18.89 (*34*)	19.34 (*35*)	20.33 (*33*)	21.24 (*33*)
random index/max total bound gap	15.25 (*37*)	13.02 (*41*)	12.45 (*41*)	12.71 (*41*)	13.61 (*42*)	15.94 (*40*)	17.47 (*37*)	18.74 (*36*)	19.73 (*34*)	20.91 (*34*)
random index/median exp. dist.	31.89 (*20*)	32.00 (*22*)	33.86 (*18*)	33.01 (*18*)	29.82 (*23*)	26.51 (*31*)	23.29 (*31*)	20.17 (*33*)	16.83 (*40*)	14.17 (*43*)
random index/min exp. dist.	07.36 (*52*)	03.06 (*51*)	03.07 (*51*)	03.09 (*51*)	03.09 (*51*)	03.12 (*52*)	03.19 (*52*)	03.27 (*52*)	03.38 (*51*)	03.49 (*50*)
random index/random index	32.49 (*19*)	32.31 (*20*)	33.79 (*20*)	32.31 (*19*)	29.74 (*25*)	27.91 (*24*)	27.07 (*27*)	26.89 (*28*)	26.67 (*28*)	26.26 (*29*)
max bound gap	31.59 (*23*)	30.11 (*23*)	29.77 (*22*)	31.09 (*23*)	36.02 (*17*)	40.90 (*15*)	46.05 (*05*)	**51.18** (*01*)	**52.54** (*01*)	**52.64** (*01*)
max total degree	43.04 (*05*)	46.07 (*10*)	46.17 (*09*)	44.92 (*09*)	43.71 (*10*)	42.01 (*12*)	40.38 (*16*)	38.59 (*15*)	37.70 (*17*)	37.83 (*16*)
min total degree	13.90 (*44*)	20.76 (*35*)	24.48 (*33*)	28.05 (*28*)	28.33 (*27*)	27.84 (*25*)	27.66 (*22*)	27.39 (*23*)	27.25 (*24*)	26.86 (*24*)
max combined total bound gap	09.19 (*50*)	07.39 (*49*)	08.14 (*48*)	09.16 (*48*)	11.67 (*46*)	13.89 (*43*)	15.91 (*40*)	17.75 (*38*)	19.19 (*36*)	20.51 (*36*)
random pair	31.65 (*22*)	32.10 (*21*)	33.49 (*21*)	31.90 (*21*)	29.86 (*22*)	28.38 (*22*)	27.44 (*23*)	26.91 (*27*)	26.55 (*29*)	26.30 (*28*)

Coloring by column:

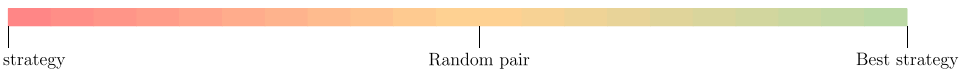


Table 3 **AUC:** for each repetition, the area under the curve (AUC) of the prediction error for a strategy is measured and averaged in rounds $i \cdot M_{MST}$ with $i \in \{1, \ldots, 10\}$. The ranking (by column) of each AUC score is noted in brackets. Coloring of each column is done linearly between the worst and baseline (random pair) score and linearly between the baseline (random pair) and the best score

Strategy	aggregation	Birch2-1	compound	D31	flame	jain	pathbased	R15	s1	s2	s3	s4	spiral	Unbalance
max degree/max degree	172.7 (12)	144.3 (06)	046.0 (03)	234.6 (07)	010.6 (08)	055.0 (10)	053.7 (11)	086.4 (04)	204.1 (06)	188.2 (06)	218.4 (12)	144.0 (09)	058.3 (12)	123.5 (04)
max degree/min degree	157.2 (07)	158.2 (10)	056.1 (09)	242.0 (08)	011.0 (10)	048.5 (07)	046.4 (06)	107.7 (12)	208.1 (07)	200.9 (12)	168.8 (03)	151.0 (11)	054.5 (06)	123.9 (06)
max degree/max exp. dist.	134.3 (04)	104.9 (02)	048.5 (05)	201.6 (04)	008.2 (02)	044.2 (04)	040.7 (02)	086.0 (03)	178.1 (04)	153.2 (04)	178.3 (05)	120.9 (03)	048.1 (04)	120.4 (02)
max degree/max prev. exp. dist.	157.0 (06)	115.4 (04)	049.0 (06)	216.4 (05)	010.9 (09)	047.3 (06)	044.3 (05)	091.4 (06)	177.4 (02)	162.4 (05)	174.8 (04)	133.9 (07)	049.1 (05)	123.7 (05)
max degree/max total bound gap	124.1 (03)	107.1 (03)	046.3 (04)	188.6 (02)	008.6 (03)	043.2 (02)	040.8 (03)	088.2 (05)	181.3 (05)	150.5 (03)	162.4 (02)	125.7 (05)	045.1 (02)	120.6 (03)
max degree/median exp. dist.	154.3 (05)	148.2 (08)	053.1 (07)	219.1 (06)	011.3 (11)	045.8 (05)	047.8 (08)	105.0 (10)	210.2 (08)	188.4 (07)	191.7 (08)	144.7 (10)	054.8 (08)	157.5 (11)
max degree/min exp. dist.	220.9 (13)	175.5 (13)	075.4 (13)	344.9 (14)	015.9 (21)	073.7 (14)	089.5 (34)	140.3 (15)	318.1 (14)	286.1 (14)	267.2 (13)	186.8 (13)	072.7 (14)	131.5 (08)
max degree/random index	168.7 (11)	164.4 (12)	057.0 (10)	256.6 (12)	010.1 (06)	053.9 (09)	052.5 (10)	097.7 (07)	219.2 (10)	189.5 (08)	213.8 (10)	132.3 (06)	055.3 (09)	141.3 (10)
linked/max degree	160.2 (09)	156.8 (09)	067.6 (12)	251.1 (11)	011.4 (12)	058.5 (11)	049.9 (09)	102.9 (09)	226.2 (12)	199.1 (11)	217.5 (11)	152.8 (12)	054.8 (07)	234.9 (16)
linked/min degree	298.5 (30)	273.9 (26)	098.9 (26)	489.7 (30)	017.8 (29)	093.6 (28)	083.0 (25)	188.2 (28)	426.8 (27)	370.7 (30)	443.1 (31)	321.6 (29)	099.6 (32)	238.8 (18)
linked/max exp. dist.	238.5 (16)	209.0 (15)	082.3 (16)	408.3 (16)	013.6 (14)	075.6 (16)	063.6 (13)	138.3 (14)	344.6 (17)	298.5 (17)	388.3 (16)	289.0 (16)	075.0 (17)	232.3 (15)
linked/max prev. exp. dist.	250.5 (20)	218.9 (19)	084.2 (18)	421.2 (20)	014.3 (19)	078.3 (20)	069.3 (18)	145.6 (18)	354.6 (18)	312.8 (19)	398.8 (19)	297.0 (18)	077.7 (20)	238.3 (17)
linked/max total bound gap	329.2 (42)	301.8 (37)	109.5 (38)	527.9 (41)	018.7 (39)	104.0 (37)	092.1 (35)	213.2 (36)	467.0 (44)	405.0 (42)	495.2 (42)	362.2 (44)	103.0 (38)	283.1 (31)
linked/median exp. dist.	314.5 (35)	400.5 (48)	137.0 (48)	489.9 (31)	019.8 (48)	121.5 (48)	095.2 (41)	209.0 (34)	431.8 (33)	374.6 (32)	465.1 (33)	330.6 (32)	102.6 (36)	1176.5 (48)
linked/min exp. dist.	474.1 (49)	594.0 (49)	219.9 (49)	718.0 (49)	029.0 (49)	191.7 (49)	158.1 (52)	311.9 (49)	616.2 (49)	540.3 (49)	750.0 (49)	535.5 (49)	145.8 (49)	1445.3 (49)
linked/random index	288.5 (22)	263.2 (22)	095.2 (22)	470.0 (22)	017.5 (23)	091.4 (22)	082.7 (24)	179.7 (22)	413.2 (22)	358.0 (22)	419.9 (21)	305.7 (21)	096.2 (22)	277.2 (30)
min degree/max degree	166.3 (10)	162.1 (11)	053.2 (08)	250.3 (10)	010.0 (05)	059.8 (12)	047.5 (07)	106.0 (11)	210.6 (09)	189.8 (09)	212.5 (09)	137.5 (08)	057.8 (11)	124.8 (07)
min degree/min degree	296.5 (27)	277.4 (30)	099.4 (29)	490.6 (32)	017.8 (25)	094.3 (31)	084.6 (29)	188.7 (29)	427.5 (29)	374.0 (31)	439.6 (29)	318.3 (27)	098.5 (30)	251.6 (26)
min degree/max exp. dist.	246.7 (17)	217.0 (17)	084.6 (20)	413.4 (17)	014.0 (16)	077.5 (17)	070.3 (19)	151.1 (20)	340.6 (16)	296.7 (16)	396.9 (17)	290.8 (17)	075.3 (18)	273.9 (29)
min degree/max prev. exp. dist.	315.5 (36)	292.9 (35)	104.7 (34)	510.2 (38)	018.3 (34)	101.5 (35)	094.6 (37)	219.6 (38)	442.6 (35)	385.2 (34)	475.8 (38)	340.8 (34)	102.0 (35)	410.2 (43)
min degree/max total bound gap	329.1 (41)	306.3 (41)	111.9 (40)	523.0 (39)	018.8 (41)	107.3 (43)	102.0 (48)	249.4 (48)	464.4 (41)	401.2 (41)	504.6 (45)	362.2 (43)	104.5 (39)	317.9 (38)
min degree/median exp. dist.	305.1 (32)	300.4 (36)	100.4 (30)	479.9 (25)	018.3 (35)	094.2 (30)	082.5 (22)	196.3 (32)	428.2 (30)	369.5 (28)	437.5 (27)	313.5 (22)	098.5 (29)	245.3 (22)
min degree/min exp. dist.	488.9 (53)	720.8 (53)	244.0 (53)	739.8 (53)	030.2 (53)	219.4 (53)	158.8 (53)	330.8 (53)	643.1 (53)	551.3 (53)	806.5 (53)	565.6 (53)	152.7 (53)	1638.4 (53)
min degree/random index	299.3 (31)	275.1 (27)	098.9 (27)	487.4 (28)	017.7 (24)	093.7 (29)	082.7 (23)	190.2 (30)	429.8 (32)	376.0 (33)	439.4 (28)	321.9 (30)	098.5 (28)	244.6 (21)
max prev. exp. dist./max degree	122.0 (02)	124.1 (05)	043.8 (02)	197.1 (03)	009.2 (04)	043.5 (03)	044.0 (04)	080.6 (02)	177.7 (03)	142.6 (02)	186.4 (06)	119.3 (02)	047.1 (03)	**118.2** (01)
max prev. exp. dist./min degree	316.3 (37)	291.2 (34)	104.9 (35)	504.6 (34)	018.0 (31)	101.1 (34)	093.3 (36)	216.7 (37)	443.9 (36)	389.4 (36)	472.2 (37)	344.3 (38)	101.9 (34)	443.8 (46)
max prev. exp. dist./max exp. dist.	249.7 (19)	222.6 (20)	084.4 (19)	420.5 (19)	014.2 (18)	077.8 (18)	067.8 (17)	147.3 (19)	359.9 (20)	313.7 (20)	403.9 (20)	298.2 (20)	075.9 (19)	249.7 (24)
max prev. exp. dist./max prev. exp. dist.	293.4 (25)	268.7 (25)	115.4 (47)	480.9 (26)	019.4 (46)	102.4 (36)	089.2 (32)	183.7 (24)	416.1 (23)	363.5 (24)	461.7 (32)	343.3 (37)	097.5 (25)	336.1 (42)
max prev. exp. dist./max total bound gap	332.3 (43)	313.2 (43)	112.5 (42)	534.5 (46)	019.5 (47)	108.2 (46)	096.9 (43)	229.6 (45)	479.6 (48)	412.9 (48)	506.7 (47)	366.8 (46)	106.3 (44)	438.2 (45)
max prev. exp. dist./median exp. dist.	321.8 (38)	329.3 (46)	107.1 (36)	507.2 (35)	018.5 (37)	105.9 (40)	094.7 (38)	224.7 (43)	445.1 (37)	389.6 (37)	467.2 (36)	339.3 (33)	105.7 (42)	458.5 (47)
max prev. exp. dist./min exp. dist.	487.6 (51)	718.4 (51)	241.6 (52)	738.0 (52)	030.1 (51)	217.6 (51)	158.0 (51)	328.9 (50)	642.0 (52)	550.6 (52)	802.5 (50)	563.0 (50)	152.0 (52)	1613.6 (51)
max prev. exp. dist./random index	312.4 (34)	289.9 (33)	100.8 (32)	507.9 (36)	018.1 (32)	099.2 (33)	089.0 (31)	208.6 (33)	446.0 (38)	388.8 (35)	466.7 (34)	342.2 (36)	102.7 (37)	304.2 (35)
max total bound gap/max degree	**103.3** (01)	**093.7** (01)	**039.1** (01)	**159.1** (01)	**007.7** (01)	**037.8** (01)	**038.4** (01)	**074.3** (01)	**140.3** (01)	**137.9** (01)	**143.8** (01)	**109.8** (01)	**044.7** (01)	139.0 (09)
max total bound gap/min degree	334.2 (47)	305.4 (39)	112.4 (41)	525.7 (40)	018.7 (38)	107.9 (44)	100.3 (47)	249.3 (47)	458.0 (39)	401.0 (40)	500.4 (43)	361.5 (42)	105.4 (41)	320.6 (40)
max total bound gap/max exp. dist.	267.3 (21)	235.7 (21)	091.4 (21)	441.1 (21)	014.8 (20)	084.0 (21)	072.1 (20)	159.4 (21)	372.4 (21)	327.5 (21)	429.6 (25)	316.9 (25)	078.6 (21)	284.4 (32)
max total bound gap/max prev. exp. dist.	332.4 (44)	310.5 (42)	115.3 (46)	530.0 (44)	019.1 (44)	108.0 (45)	098.3 (44)	226.9 (44)	475.0 (46)	408.4 (44)	508.2 (48)	366.2 (45)	107.8 (48)	419.0 (44)
max total bound gap/max total bound gap	337.1 (48)	313.8 (44)	112.8 (43)	538.8 (48)	019.4 (45)	106.6 (41)	096.3 (42)	221.6 (40)	471.2 (45)	410.9 (46)	504.8 (46)	370.4 (48)	106.8 (45)	318.1 (39)
max total bound gap/median exp. dist.	333.6 (46)	349.9 (47)	114.7 (45)	529.0 (43)	018.9 (42)	110.6 (47)	099.8 (46)	238.6 (46)	465.7 (42)	409.7 (45)	491.4 (40)	357.5 (39)	106.9 (46)	314.4 (37)
max total bound gap/min exp. dist.	487.1 (50)	717.1 (50)	240.6 (50)	737.8 (51)	030.0 (50)	217.4 (50)	157.2 (49)	329.0 (51)	641.9 (50)	550.0 (50)	804.0 (52)	564.0 (52)	151.7 (50)	1602.1 (50)
max total bound gap/random index	327.6 (39)	303.0 (38)	108.8 (37)	530.5 (45)	018.4 (36)	105.1 (39)	095.0 (39)	221.0 (39)	462.7 (40)	400.4 (39)	489.6 (39)	359.3 (40)	104.8 (40)	295.6 (34)
random index/max degree	232.9 (14)	216.2 (16)	079.3 (14)	334.2 (13)	013.0 (13)	073.4 (13)	065.0 (15)	136.2 (13)	303.6 (13)	240.1 (13)	268.7 (14)	202.5 (14)	067.6 (13)	260.6 (28)
random index/min degree	297.7 (29)	275.2 (28)	098.7 (25)	487.8 (29)	017.9 (30)	093.3 (27)	084.8 (30)	187.2 (26)	429.2 (31)	369.2 (27)	432.9 (26)	320.8 (28)	098.2 (27)	244.2 (20)
random index/max exp. dist.	236.1 (15)	206.7 (14)	081.1 (15)	400.8 (15)	013.7 (15)	075.2 (15)	065.0 (14)	141.3 (16)	332.3 (15)	294.1 (15)	381.1 (15)	285.0 (15)	072.8 (15)	239.3 (19)
random index/max prev. exp. dist.	307.6 (33)	285.1 (31)	102.9 (33)	509.7 (37)	018.1 (33)	097.5 (32)	089.4 (33)	210.0 (35)	441.8 (34)	390.8 (38)	466.8 (35)	341.0 (35)	101.4 (33)	305.0 (36)
random index/max total bound gap	328.1 (40)	306.2 (40)	110.0 (39)	528.3 (42)	018.7 (40)	104.1 (38)	095.1 (40)	222.0 (41)	466.1 (43)	405.8 (43)	495.2 (41)	360.6 (41)	106.0 (43)	293.4 (33)
random index/median exp. dist.	296.4 (26)	288.9 (32)	100.5 (31)	478.0 (24)	017.8 (26)	091.8 (24)	083.0 (26)	192.8 (31)	421.4 (26)	359.5 (23)	426.6 (24)	313.5 (23)	098.1 (26)	258.8 (27)
random index/min exp. dist.	487.8 (52)	718.5 (52)	241.3 (51)	737.8 (50)	030.1 (52)	217.6 (52)	158.0 (50)	329.3 (52)	641.9 (51)	550.5 (51)	802.8 (51)	563.1 (51)	151.7 (51)	1614.7 (52)
random index/random index	292.5 (24)	264.5 (24)	095.8 (23)	476.2 (23)	017.8 (27)	091.4 (23)	081.4 (21)	183.8 (25)	421.0 (25)	367.8 (26)	425.6 (23)	314.0 (24)	096.9 (23)	224.6 (14)
max bound gap	248.6 (18)	218.0 (18)	084.2 (17)	417.3 (18)	014.1 (17)	078.2 (19)	066.5 (16)	145.5 (17)	354.8 (19)	307.6 (18)	398.5 (18)	298.1 (19)	075.0 (16)	248.9 (23)
max total degree	159.8 (08)	147.1 (07)	062.7 (11)	243.5 (09)	010.3 (07)	049.6 (08)	054.9 (12)	098.6 (08)	220.1 (11)	192.5 (10)	191.4 (07)	121.7 (04)	057.6 (10)	160.9 (12)
min total degree	297.3 (28)	275.2 (29)	099.3 (28)	491.3 (33)	017.8 (28)	092.9 (26)	084.1 (28)	188.1 (27)	426.9 (28)	369.7 (29)	440.9 (30)	321.9 (31)	099.2 (31)	250.8 (25)
max combined total bound gap	332.7 (45)	316.6 (45)	113.8 (44)	536.3 (47)	018.9 (43)	107.0 (42)	098.6 (45)	224.3 (42)	476.2 (47)	412.0 (47)	503.6 (44)	367.2 (47)	106.9 (47)	321.5 (41)
random pair	291.2 (23)	264.2 (23)	096.2 (24)	481.6 (27)	017.4 (22)	092.1 (25)	083.6 (27)	182.9 (23)	417.5 (24)	365.1 (25)	423.8 (22)	317.3 (26)	096.9 (24)	223.5 (13)

Coloring by column:

Worst strategy — Random pair — Best strategy

better. Comparing the ranks in Table 1 in round $5 \cdot M_{MST}$ with round $10 \cdot M_{MST}$ gives:

$$\begin{pmatrix} 04^{\text{th}} & 14^{\text{th}} & 15^{\text{th}} & 18^{\text{th}} & 21^{\text{st}} & 13^{\text{th}} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 11^{\text{th}} & 03^{\text{rd}} & 07^{\text{th}} & 02^{\text{nd}} & 05^{\text{th}} & 04^{\text{th}} \end{pmatrix}$$

Only *max degree / max exp. distance* loses terrain. After round $10 \cdot M_{MST}$, the top 7 contains five strategies with *max exp. distance*, which is noteworthy.

Observation 7 Performance is relatively robust across datasets (AUC scores).

Table 4 **Highlighted ranks:** the ranks of the strategy *max total bound gap / max degree* from Tables 1 and 2

	1 M_{MST}	2 M_{MST}	3 M_{MST}	4 M_{MST}	5 M_{MST}	6 M_{MST}	7 M_{MST}	8 M_{MST}	9 M_{MST}	10 M_{MST}
Average performance	06	01	01	01	01	01	01	08	08	08
Borda count	09	01	01	01	01	01	02	08	08	08

Table 5 Highlighted ranks: the ranks of the strategy *max bound gap* from Tables 1 and 2

	1 M_{MST}	2 M_{MST}	3 M_{MST}	4 M_{MST}	5 M_{MST}	6 M_{MST}	7 M_{MST}	8 M_{MST}	9 M_{MST}	10 M_{MST}
Average performance	25	26	22	19	18	17	10	02	01	01
Borda count	23	23	22	23	17	15	05	01	01	01

Evidence: In Table 3, every strategy has approximately the same color across datasets. This means that the relative performance is not very dependent on the dataset. However, *Unbalance* gives the most deviant results. This implies that the balancedness of the dataset could influence the performance of a strategy.

8 Real world experiment

In order to test if the observations also hold for *real world* datasets, we also evaluate the strategies on the *cifar10* [36] and *mnist* [37] datasets. These datasets consist of images of ten different categories. To limit memory space and running time, we only take the first 1,000 samples of the training set for each dataset. The distance between two images is determined by the Euclidean norm, which was also used in the previous experiments. The results can be found in Table 6, where the *average performance* is given (see Section 6.4.1).

Next, we discuss (using Table 6) if the observations from Section 7 also hold for these real world datasets. Still, there are many better strategies than simply choosing a *random pair* (Observation 1). *Max total bound gap / max degree* also remains the best strategy for earlier rounds (Observation 2), but now the performance falls off after $2 \cdot M_{MST}$ rounds. *Max degree* is generally a good criterion (Observation 3). The best strategies often use this criterion. *Min exp. distance* and *median exp. distance* are still bad criteria (Observation 4). But now, *max bound gap* is not a good strategy even after $> 7 \cdot M_{MST}$ rounds (Observation 5). After $10 \cdot M_{MST}$ rounds, it ranks 30$^{\text{th}}$, whilst simply selecting a random pair ranks 20$^{\text{th}}$. Perhaps, this strategy needs even more rounds to become good. *Max exp. distance* is also not longer a late bloomer (Observation 6), as multiple strategies with this criterion rank higher after $10 \cdot M_{MST}$ rounds, then after $5 \cdot M_{MST}$ rounds. Furthermore, it ranks worse after $10 \cdot M_{MST}$ rounds compared with the previous experiment. Perhaps, this strategy also needs more rounds to start blooming. We believe that the difference could be explained by the dimensionality of the datasets. The *cifar10* and *mnist* dataset have a higher dimensionality ($32 \times 32 \times 3$) and (28×28), respectively. It is well-known that in higher dimensional space, most points will be far away. Therefore, dimensionality could play a role in the distribution of pairwise distances. This in turn, could have an effect on some strategies such as *max bound gap* and *max exp. distance*, which is why we believe that these strategies may need more time to start performing well on these datasets. The AUC performance remains relatively stable for these datasets (Observation 7).

In general, most previous observations still hold for these real world datasets. Only some strategies that previously performed well in the later rounds, did not start improving as well on these datasets. It could be that more rounds are necessary.

8.1 Performance max degree

An important observation from both Section 7 and Table 6, is that *max degree* is a good criterion. The best performing methods often include this criterion. We briefly want to discuss why we believe that choosing a sample that has been already chosen often (max degree) is beneficial. In order to predict the actual distance, a lower and upper bound is established using the triangle inequality (Section 4.2). When the distance is labeled between i and j, the triangle inequality can be used to derive information about the distances between i and k if the distance between j and k is known. Therefore, labeling a sample with the highest degree, gives a lot of possible triangle inequality combinations that can be made, which could provide much information. This is why we believe that this criterion performs really well.

9 Discussion and future research

This research can be viewed as a pioneering contribution and is a significant first step in APDL. Below we elaborate on both the shortcomings of the approach proposed, and the related challenges for further research.

Perfect expert It is assumed that the expert does not make any mistake in determining the distance between two instances. This is a common, yet unreasonably optimistic, assumption in AL research. Settles [38] states that "we have often assumed that there is a single infallible annotator whose labels can be trusted" and views this assumption as one of the six practical challenges for AL. How to deal with a noisy expert remains a critical research problem. A way of mitigating the mistakes of the expert in APDL is to allow some ϵ-boundary around the labels and incorporating this into the approximation bounds. Still, there are many more

Table 6 Average performance (cifar10 & mnist): for each dataset (*cifar10* & *mnist*) and repetition, the prediction error of a strategy is averaged in rounds $i \cdot M_{MST}$ with $i \in \{1, \ldots, 10\}$. The ranking (by column) of each average prediction error is noted in brackets. Coloring of each column is done linearly between the worst and baseline (random pair) score and linearly between the baseline (random pair) and the best score

Strategy	1 M_{MST}	2 M_{MST}	3 M_{MST}	4 M_{MST}	5 M_{MST}	6 M_{MST}	7 M_{MST}	8 M_{MST}	9 M_{MST}	10 M_{MST}
max degree/max degree	12.961 (08)	7.957 (05)	6.469 (04)	5.568 (06)	4.905 (03)	**4.358** (01)	4.098 (03)	3.891 (04)	3.581 (02)	3.389 (03)
max degree/min degree	12.529 (06)	7.847 (04)	**6.328** (01)	5.554 (05)	**4.889** (01)	4.567 (04)	4.273 (08)	4.043 (09)	3.823 (09)	3.721 (09)
max degree/max exp. dist.	13.199 (10)	9.023 (10)	6.625 (07)	6.075 (11)	5.400 (11)	5.093 (11)	4.230 (06)	4.030 (08)	3.645 (05)	3.542 (06)
max degree/max prev. exp. dist.	11.332 (04)	8.231 (09)	6.871 (10)	5.695 (08)	4.992 (05)	4.655 (07)	4.449 (10)	4.234 (11)	4.055 (11)	3.910 (12)
max degree/max total bound gap	14.655 (11)	10.614 (13)	6.375 (02)	5.656 (07)	5.105 (07)	4.927 (10)	4.662 (11)	3.930 (05)	3.772 (07)	3.667 (07)
max degree/median exp. dist.	11.391 (05)	7.789 (03)	6.440 (03)	5.425 (04)	4.893 (02)	4.370 (03)	**4.033** (01)	3.807 (02)	**3.560** (01)	3.413 (04)
max degree/min exp. dist.	12.574 (07)	10.236 (12)	8.898 (13)	7.819 (14)	6.944 (14)	6.603 (14)	5.918 (14)	5.638 (14)	4.881 (13)	4.189 (13)
max degree/random index	13.170 (09)	9.060 (11)	7.501 (12)	5.373 (02)	5.090 (06)	4.620 (05)	4.260 (07)	3.947 (07)	3.794 (08)	3.688 (08)
linked/max degree	17.355 (14)	**7.641** (01)	7.075 (11)	5.376 (03)	5.114 (08)	4.367 (02)	4.186 (05)	**3.722** (01)	3.628 (03)	3.371 (02)
linked/min degree	20.476 (45)	20.332 (36)	20.074 (34)	19.679 (32)	19.132 (31)	18.451 (28)	17.655 (25)	16.776 (24)	15.839 (24)	14.900 (24)
linked/max exp. dist.	20.436 (17)	20.246 (17)	19.938 (17)	19.520 (17)	18.980 (19)	18.361 (19)	17.676 (28)	16.975 (31)	16.268 (31)	15.571 (31)
linked/max prev. exp. dist.	20.434 (15)	20.230 (16)	19.929 (16)	19.501 (16)	18.992 (20)	18.431 (24)	17.822 (36)	17.181 (36)	16.501 (36)	15.867 (36)
linked/max total bound gap	20.476 (47)	20.323 (29)	20.131 (38)	19.870 (39)	19.524 (40)	19.100 (42)	18.614 (43)	18.078 (43)	17.495 (44)	16.859 (42)
linked/median exp. dist.	20.440 (18)	20.249 (18)	19.942 (18)	19.520 (18)	18.973 (18)	18.368 (21)	17.664 (27)	16.892 (27)	15.931 (26)	14.787 (19)
linked/min exp. dist.	20.459 (29)	20.392 (49)	20.333 (51)	20.284 (51)	20.224 (51)	20.164 (52)	20.113 (53)	20.056 (53)	19.992 (53)	19.929 (53)
linked/random index	20.435 (16)	20.228 (15)	19.896 (15)	19.441 (15)	18.841 (15)	18.129 (15)	17.269 (15)	16.408 (15)	15.505 (15)	14.655 (15)
min degree/max degree	11.146 (03)	8.016 (06)	6.715 (08)	**5.297** (01)	4.978 (04)	4.714 (09)	4.067 (02)	3.873 (03)	3.655 (06)	**3.367** (01)
min degree/min degree	20.475 (44)	20.333 (38)	20.079 (37)	19.688 (35)	19.143 (33)	18.457 (29)	17.646 (23)	16.751 (22)	15.819 (22)	14.883 (22)
min degree/max exp. dist.	20.469 (31)	20.324 (30)	20.064 (29)	19.669 (29)	19.126 (30)	18.441 (27)	17.635 (21)	16.785 (25)	15.910 (25)	15.044 (26)
min degree/max prev. exp. dist.	20.470 (34)	20.328 (35)	20.074 (35)	19.698 (37)	19.187 (37)	18.535 (36)	17.785 (32)	16.970 (30)	16.133 (29)	15.323 (29)
min degree/max total bound gap	20.476 (46)	20.347 (39)	20.152 (44)	19.895 (46)	19.571 (46)	19.181 (47)	18.734 (48)	18.235 (48)	17.694 (48)	17.125 (48)
min degree/median exp. dist.	20.469 (33)	20.326 (33)	20.073 (32)	19.687 (34)	19.144 (34)	18.465 (31)	17.655 (26)	16.745 (21)	15.729 (18)	14.671 (16)
min degree/min exp. dist.	20.482 (53)	20.426 (53)	20.366 (53)	20.306 (53)	20.239 (53)	20.174 (53)	20.101 (52)	20.031 (52)	19.957 (52)	19.890 (52)
min degree/random index	20.471 (35)	20.328 (34)	20.071 (31)	19.678 (31)	19.135 (32)	18.435 (25)	17.639 (22)	16.744 (20)	15.804 (21)	14.889 (23)
max prev. exp. dist./max degree	11.034 (02)	8.138 (08)	6.758 (09)	6.105 (12)	5.294 (10)	4.657 (08)	4.172 (04)	3.934 (06)	3.633 (04)	3.481 (05)
max prev. exp. dist./min degree	20.468 (30)	20.326 (32)	20.074 (33)	19.691 (36)	19.171 (36)	18.525 (35)	17.758 (30)	16.957 (28)	16.117 (28)	15.318 (28)
max prev. exp. dist./max exp. dist.	20.457 (27)	20.291 (27)	20.009 (26)	19.593 (25)	19.068 (25)	18.458 (30)	17.791 (34)	17.111 (34)	16.443 (35)	15.798 (35)
max prev. exp. dist./max prev. exp. dist.	20.456 (25)	20.285 (24)	20.000 (24)	19.600 (26)	19.114 (28)	18.536 (37)	17.913 (37)	17.273 (37)	16.630 (37)	15.988 (37)
max prev. exp. dist./max total bound gap	20.474 (36)	20.350 (43)	20.151 (43)	19.879 (42)	19.535 (43)	19.108 (43)	18.609 (41)	18.066 (41)	17.486 (42)	16.863 (43)
max prev. exp. dist./median exp. dist.	20.453 (19)	20.277 (19)	19.989 (20)	19.577 (23)	19.057 (24)	18.440 (26)	17.786 (33)	17.078 (33)	16.364 (33)	15.599 (32)
max prev. exp. dist./min exp. dist.	20.477 (50)	20.407 (50)	20.328 (49)	20.246 (49)	20.151 (49)	20.049 (49)	19.943 (49)	19.832 (50)	19.705 (50)	19.577 (50)
max prev. exp. dist./random index	20.459 (28)	20.298 (28)	20.022 (28)	19.624 (28)	19.106 (27)	18.468 (32)	17.768 (31)	17.059 (32)	16.323 (32)	15.624 (33)
max total bound gap/max degree	**10.005** (01)	7.721 (02)	6.537 (06)	6.005 (10)	5.709 (12)	5.496 (13)	5.312 (13)	5.175 (13)	5.066 (14)	4.977 (14)
max total bound gap/min degree	20.476 (49)	20.347 (41)	20.154 (45)	19.891 (45)	19.559 (45)	19.164 (45)	18.709 (47)	18.201 (47)	17.656 (47)	17.070 (47)
max total bound gap/max exp. dist.	20.474 (39)	20.351 (45)	20.157 (46)	19.879 (43)	19.525 (41)	19.095 (40)	18.606 (40)	18.065 (40)	17.459 (40)	16.816 (40)
max total bound gap/max prev. exp. dist.	20.474 (38)	20.347 (40)	20.149 (41)	19.879 (44)	19.540 (44)	19.117 (44)	18.618 (44)	18.084 (44)	17.493 (43)	16.875 (44)
max total bound gap/max total bound gap	20.474 (37)	20.357 (48)	20.170 (47)	19.912 (48)	19.578 (47)	19.182 (48)	18.706 (46)	18.177 (46)	17.591 (46)	16.967 (46)
max total bound gap/median exp. dist.	20.476 (48)	20.351 (44)	20.148 (39)	19.869 (38)	19.515 (39)	19.085 (38)	18.587 (38)	18.030 (38)	17.432 (38)	16.783 (38)
max total bound gap/min exp. dist.	20.479 (52)	20.422 (52)	20.362 (52)	20.297 (52)	20.227 (52)	20.158 (51)	20.081 (51)	20.008 (51)	19.931 (51)	19.852 (51)
max total bound gap/random index	20.475 (41)	20.347 (42)	20.148 (40)	19.871 (40)	19.512 (38)	19.087 (39)	18.594 (39)	18.037 (39)	17.448 (39)	16.804 (39)
random index/max degree	16.248 (13)	11.815 (14)	8.985 (14)	7.204 (13)	6.047 (13)	5.294 (12)	4.745 (12)	4.374 (12)	4.089 (12)	3.876 (11)
random index/min degree	20.469 (32)	20.324 (31)	20.068 (30)	19.675 (30)	19.122 (29)	18.425 (23)	17.607 (19)	16.722 (19)	15.792 (20)	14.879 (21)
random index/max exp. dist.	20.456 (26)	20.284 (23)	20.002 (25)	19.585 (24)	19.035 (23)	18.361 (20)	17.611 (20)	16.812 (26)	16.031 (27)	15.269 (27)
random index/max prev. exp. dist.	20.455 (22)	20.290 (26)	20.014 (27)	19.607 (27)	19.088 (26)	18.480 (34)	17.815 (35)	17.127 (35)	16.418 (34)	15.721 (34)
random index/max total bound gap	20.475 (42)	20.352 (46)	20.151 (42)	19.875 (41)	19.528 (42)	19.098 (41)	18.610 (42)	18.066 (42)	17.467 (41)	16.825 (41)
random index/median exp. dist.	20.455 (20)	20.281 (20)	19.984 (19)	19.556 (20)	18.967 (17)	18.284 (17)	17.510 (18)	16.675 (18)	15.709 (17)	14.689 (17)
random index/min exp. dist.	20.477 (51)	20.410 (51)	20.332 (50)	20.250 (50)	20.157 (50)	20.061 (50)	19.948 (50)	19.829 (49)	19.700 (49)	19.570 (49)
random index/random index	20.456 (23)	20.287 (25)	19.997 (23)	19.552 (19)	18.965 (16)	18.260 (16)	17.443 (16)	16.555 (16)	15.619 (16)	14.699 (18)
max bound gap	20.456 (24)	20.283 (21)	19.997 (22)	19.574 (22)	19.029 (22)	18.369 (22)	17.680 (29)	16.962 (29)	16.228 (30)	15.524 (30)
max total degree	14.995 (12)	8.042 (07)	6.510 (05)	5.781 (09)	5.247 (09)	4.631 (06)	4.363 (09)	4.140 (10)	3.903 (10)	3.742 (10)
min total degree	20.475 (43)	20.332 (37)	20.075 (36)	19.686 (33)	19.149 (35)	18.473 (33)	17.654 (24)	16.759 (23)	15.834 (23)	14.903 (25)
max combined total bound gap	20.474 (40)	20.356 (47)	20.171 (48)	19.909 (47)	19.579 (48)	19.176 (46)	18.701 (45)	18.166 (45)	17.581 (45)	16.946 (45)
random pair	20.455 (21)	20.284 (22)	19.994 (21)	19.566 (21)	18.993 (21)	18.298 (18)	17.495 (17)	16.644 (17)	15.733 (19)	14.819 (20)

Coloring by column:

ways to deal with an imperfect expert, which should be investigated. Using properties of a metric, mistakes can be spotted and reevaluated.

Underlying distance metric In all experiments, the Euclidean distance was used as underlying distance metric. This might affect the conclusions that were drawn, as

alternative distance metrics might be favorable for different strategies. In future research, this could be investigated by changing the underlying distance metric and evaluating if the same strategies are always performing the best.

Complex strategies In our research, we have examined many selection algorithms based on straightforward criteria. Newer and more complex strategies could be developed, reducing the prediction error even more. Consider for example mixing strategies, where one strategy works well in the beginning (e.g., *max total bound gap / max degree*) and switch to another strategy (e.g., *max bound gap*) that works better later on. Another way, would be to select each round a specific strategy with a certain probability. Additionally, *transfer learning* [5, 6] can be applied to train an even more advanced model (e.g., a neural network) using labeled datasets. Such a model can be trained to choose a good strategy at a specific time, where the new prediction error can be used to either reward or penalize the selection. If the chosen strategy selected a pair that gave a lot of insight, the model can be updated to select this strategy more often in similar cases. When properly trained, the model could be applied to new datasets to determine the selection strategy. Whether this is a good approach, depends on the ability of the model to transfer the learned information over to the new dataset.

Running time In this research, we have used straightforward criteria that are easy to compute. However, when more complex strategies are designed, *running time* could start to play a role. The importance of running time is mostly task dependent. The cost of coming up with the next query should be balanced with the cost of the labeling done by the expert. We consider APDL to be particularly useful in situations where the expert can only be queried a limited number of times (due to high costs). However, running time is something that should be considered in future work when more complex strategies are used. When a strategy is too hard to compute, approximation algorithms could be developed.

Running time In this research, we have used straightforward criteria that are easy to compute. However, when more complex strategies are designed, *running time* could start to play a role. The importance of running time is mostly task dependent. The cost of coming up with the next query should be balanced with the cost of the labeling done by the expert. We consider APDL to be particularly useful in situations where the expert can only be queried a limited number of times (due to high costs). However, running time is something that should be considered in future work when more complex strategies are used. When a strategy is too hard to compute, approximation algorithms could be developed. The average running time of each strategy can be seen in Table 7. We believe that the difference in running time can mostly be explained by the following phenomenon. When there are more samples that satisfy the selection criterion, a random selection is made between these samples. This function takes more time, when there are more samples to choose from. Consider, for example, the difference between *random index / max degree* and *random index / min degree* that take on average 685 and 978 seconds, respectively.

Table 7 Average running time: the running time of each strategy averaged over all repetitions and datasets (including cifar10 & mnist). The ranking is noted in brackets. Coloring is done linearly between the worst and best score

Strategy	Time (s)	Strategy	Time (s)	Strategy	Time (s)
max degree/max degree	0720 (*16*)	max degree/min degree	0729 (*19*)	max degree/max exp. dist.	0678 (*09*)
max degree/max prev. exp. dist.	0741 (*20*)	max degree/max total bound gap	0701 (*13*)	max degree/median exp. dist.	0681 (*11*)
max degree/min exp. dist.	0706 (*14*)	max degree/random index	0678 (*10*)	linked/max degree	0722 (*18*)
linked/min degree	1009 (*41*)	linked/max exp. dist.	1247 (*49*)	linked/max prev. exp. dist.	1170 (*47*)
linked/max total bound gap	0947 (*36*)	linked/median exp. dist.	0915 (*32*)	linked/min exp. dist.	0583 (*05*)
linked/random index	0995 (*40*)	min degree/max degree	0751 (*21*)	min degree/min degree	1052 (*45*)
min degree/max exp. dist.	1228 (*48*)	min degree/max prev. exp. dist.	0889 (*27*)	min degree/max total bound gap	0891 (*29*)
min degree/median exp. dist.	1070 (*46*)	min degree/min exp. dist.	0583 (*04*)	min degree/random index	1022 (*43*)
max prev. exp. dist./max degree	0710 (*15*)	max prev. exp. dist./min degree	0902 (*30*)	max prev. exp. dist./max exp. dist.	1303 (*51*)
max prev. exp. dist./max prev. exp. dist.	1352 (*53*)	max prev. exp. dist./max total bound gap	0868 (*24*)	max prev. exp. dist./median exp. dist.	0828 (*22*)
max prev. exp. dist./min exp. dist.	0606 (*07*)	max prev. exp. dist./random index	0944 (*35*)	max total bound gap/max degree	0661 (*08*)
max total bound gap/min degree	0921 (*34*)	max total bound gap/max exp. dist.	1274 (*50*)	max total bound gap/max prev. exp. dist.	0882 (*26*)
max total bound gap/max total bound gap	0917 (*33*)	max total bound gap/median exp. dist.	0837 (*23*)	max total bound gap/min exp. dist.	0575 (*02*)
max total bound gap/random index	0889 (*28*)	random index/max degree	0685 (*12*)	random index/min degree	0978 (*39*)
random index/max exp. dist.	1321 (*52*)	random index/max prev. exp. dist.	0878 (*25*)	random index/max total bound gap	0911 (*31*)
random index/median exp. dist.	0960 (*38*)	random index/min exp. dist.	0577 (*03*)	random index/random index	0956 (*37*)
max bound gap	1017 (*42*)	max total degree	0720 (*17*)	min total degree	1044 (*44*)
max combined total bound gap	0598 (*06*)	random pair	**0563** (*01*)		

There are considerably more samples with the same *minimum* degree compared to the *maximum* degree. In Table 7, we observe that strategies consistently are slower when they have more samples that satisfy the criterion.

Space complexity In the experiments, at most $M = 1{,}000$ samples were used, as this already leads to 499,500 different pairs. To store the approximation bounds for each pair, $\mathcal{O}(M^2)$ is necessary. This can quickly become infeasible for large M. Although rather time expensive, these approximation bounds could be calculated every time they are needed. Yet, for large problems, a better solution is necessary. A major insight of this research is that choosing based on *max degree* consistently performs well. This criterion does not use any information from the approximation bounds, which is why this is ideal for large problems, as the approximation bounds are only necessary for the final predictions. More research is necessary to optimize large APDL problems.

Using feature values It was assumed in Section 2.2 that no feature values should be used. In this way, the observations from this research are not dependent on the application domain. Furthermore, if new methods are developed that do use feature values, our tested selection strategies can function as a good baseline. Adding information (using the feature values) should only increase the performance of an APDL method. Thus, when a model is performing worse than any one of our suggested strategies, it should be considered as a major warning sign. Additionally, during the APDL process, a model could be used to evaluate if the feature values could help the prediction. If so, feature values could be introduced into the query selection after some rounds.

Gaining insight Demystifying AL can give us critical insights. Which samples are useful to query? Can we understand why? Can we explain why certain selection algorithms perform better? Is the clusteredness/balancedness of a dataset relevant? Are there better indicators for the usefulness of a sample query? Answering these kinds of questions could lead to better performing models.

Error reduction rate The reduction rate in prediction error instigates many exciting research opportunities. Can guarantees be derived about the speed with which the prediction error converges for certain strategies? It would be especially useful for practical applications to know how many labels should be gathered to get at most a prediction error of $\delta > 0$. To derive such a guarantee, either theoretical proof or substantial numerical evidence is necessary. Additionally, the effect of a tight or loose initial upper bound for the maximum distance on the convergence speed could also be investigated.

Additional application We think that APDL can also be used to determine the complexity of a dataset. When a strategy needs more rounds to attain a certain prediction error, the dataset might be more complex, as it is harder to learn the pairwise distances. In this way, APDL can even be useful for fully labeled datasets. Which strategies to use and how complexity is exactly quantified with APDL are all interesting subjects for future research.

Prediction model Recall that there are two critical components in APDL, namely 'Which pair is queried each round?' and 'How to use this information to make the best prediction?' The focus of our research was to answer the first question. To make a prediction of a distance, we used the upper and lower bound approximation and took the average as prediction (see Definition 4). Therein lies a large opportunity for improvement, as a more advanced prediction model could improve the final prediction as well as the query selection. Using a tuned weighted average of the upper and lower approximation could already perform better.

10 Summary

We started by introducing the problem of APDL, where the goal is to actively learn the pairwise distances between all instances. We established upper and lower bound approximations using properties of a distance function. Furthermore, we presented an update rule that automatically updates the upper and lower bounds using the newest labeled distance. Then, we provided fourteen selection criteria, which gave us 53 query strategies combined. These strategies do not use feature values, making the observations from the experiments domain-independent. This makes these selection strategies ideal candidates for a baseline in future research.

The experiments led to valuable new insights. These observations were tested by evaluating all strategies on two real world datasets (*cifar10* & *mnist*). We found multiple strategies that perform better than simply randomly selecting a pair (Observation 1). This shows that it is indeed possible to 'smartly' select the indices. We determined that the performance of the strategies was not very dependent on the datasets (Observation 7). The performance only changed somewhat in a highly unbalanced case. We identified *max degree* to be a consistently good criterion. In Section 8.1, we explained why we believe that this criterion is useful. Consequently, we also discovered which strategies should not be chosen due to general bad performance (Observation 4). Choosing the right selection strategy could potentially

save many hours and resources. The findings from the experiments are not dependent on the dimensionality of the data or (noisy) feature values, as feature values were not taken into account. However, more dimensions could lead to higher sparsity (curse of dimensionality), which is why a mix of sparse and dense datasets were used.

Acknowledgements The authors wish thank the anonymous referees for their useful comments, which has led to a significant improvement of the readability and quality of the paper.

Author Contributions (*Contributor Roles Taxonomy* (CRediT)

- Joris Pries: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing - original draft, Writing - review & editing
- Sandjai Bhulai: Conceptualization, Supervision, Validation, Writing - review & editing
- Rob van der Mei: Conceptualization, Supervision, Validation, Writing - review & editing

Funding No funding was received for conducting this study.

Data Availability The datasets used in the experiments of this study are made openly available by [34] and can be accessed here: http://cs.uef.fi/sipu/datasets/. The real world datasets (*cifar10* [36] & *mnist* [37]) can be attained through https://keras.io/api/datasets/.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Settles B (2009) Active learning literature survey. University of Wisconsin–Madison, vol 1648
2. Vlachos A (2008) A stopping criterion for active learning. Computer Speech Lang 22(3):295–312. https://doi.org/10.1016/j.csl.2007.12.001
3. Ishibashi H, Hino H (2020) Stopping criterion for active learning based on deterministic generalization bounds. In: Chiappa S, Calandra R (eds) Proceedings of the twenty third international conference on artificial intelligence and statistics. vol 108 of proceedings of machine learning research. PMLR. pp 386–397. Available from: https://proceedings.mlr.press/v108/ishibashi20a.html
4. Callaghan MW, Müller-Hansen F (2020) Statistical stopping criteria for automated screening in systematic reviews. Systematic Rev 9(1):273. https://doi.org/10.1186/s13643-020-01521-4
5. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. http://www.deeplearningbook.org
6. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H et al (2021) A comprehensive survey on transfer learning. Proc IEEE 109(1):43–76. https://doi.org/10.1109/JPROC.2020.3004555
7. Yoo D, Kweon IS (2019) Learning loss for active learning. arXiv:1905.03677
8. Klein J, Bhulai S, Hoogendoorn M, Van der Mei R (2021) IEEE. Plusmine: dynamic active learning with semi-supervised learning for automatic classification. 2021 IEEE/WIC/ACM international conference on web intelligence
9. Aggarwal C, Kong X, Gu Q, Han J, Yu P (2014) In: Aggarwal C (ed) Active learning: a survey. CRC Press. pp 571–605. Publisher Copyright: © 2015 by Taylor & Francis Group, LLC
10. Gal Y, Islam R, Ghahramani Z (2017) Deep Bayesian active learning with image data. In: Precup D, Teh YW (eds) Proceedings of the 34th international conference on machine learning. vol. 70 of proceedings of machine learning research. PMLR. pp 1183–1192. Available from: https://proceedings.mlr.press/v70/gal17a.html
11. Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB et al (2021) A survey of deep active learning. ACM Comput Surv, vol 54(9). https://doi.org/10.1145/3472291
12. Chapelle O, Schlkopf B, Zien A (2006) Semi-supervised learning. IEEE Trans Neural Netw, vol 20
13. Sutton RS, Barto AG. (2018) Reinforcement learning: an introduction. Cambridge, MA USA: a bradford book
14. Hüllermeier E, Fürnkranz J, Cheng W, Brinker K (2008) Label ranking by learning pairwise preferences. Artif Intell 172(16):1897–1916. https://doi.org/10.1016/j.artint.2008.08.002
15. Dasarathy G, Nowak R, Zhu X (2015) S2: an efficient graph based active learning algorithm with application to nonparametric classification. In: Grünwald P, Hazan E, Kale S (eds) Proceedings of the 28th conference on learning theory, vol 40 of proceedings of machine learning research. Paris, France: PMLR. pp 503–522. Available from: https://proceedings.mlr.press/v40/Dasarathy15.html
16. Eriksson B, Dasarathy G, Singh A, Nowak R (2011) Active clustering: robust and efficient hierarchical clustering using adaptively selected similarities. In: Gordon G, Dunson D, Dudík M (eds) Proceedings of the fourteenth international conference on artificial intelligence and statistics, vol 15 of proceedings of machine learning research. Fort Lauderdale, FL, USA: PMLR. pp 260–268. Available from: https://proceedings.mlr.press/v15/eriksson11a.html
17. Zhang R, Lin L, Zhang R, Zuo W, Zhang L (2015) Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. IEEE Trans Image Process 24(12):4766–4779. https://doi.org/10.1109/TIP.2015.2467315
18. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244
19. Köstinger M, Hirzer M, Wohlhart P, Roth PM, Bischof H (2012) Large scale metric learning from equivalence constraints. In: 2012 IEEE conference on computer vision and pattern recognition, pp 2288–2295
20. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: a unified embedding for face recognition and clustering. arXiv:1503.03832
21. Yang L, Jin R, Sukthankar R (2012) Bayesian active distance metric learning. arXiv:1206.5283

22. Kumaran K, Papageorgiou D, Chang Y, Li M, Takáč M (2018) Active metric learning for supervised classification
23. Ebert S, Fritz M, Schiele B. (2012) Active metric learning for object recognition. In: Pinz A, Pock T, Bischof H, Leberl F (eds) Pattern recognition. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 327-336
24. Pasolli E, Yang HL, Crawford MM (2016) Active-metric learning for classification of remotely sensed hyperspectral images. IEEE Trans Geosci Remote Sensing 54(4):1925–1939. https://doi.org/10.1109/TGRS.2015.2490482
25. Fränti P, Virmajoki O (2006) Iterative shrinking method for clustering problems. Pattern Recognit 39(5):761–765. https://doi.org/10.1016/j.patcog.2005.09.012
26. Rezaei M, Fränti P (2016) Set-matching methods for external cluster validity. IEEE Trans Knowl Data Eng 28(8):2173–2186
27. Zhang T, Ramakrishnan R, Livny M (1997) BIRCH: a new data clustering algorithm and its applications. Data Mining Knowl Discover 1(2):141–182
28. Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. ACM Trans Knowl Discov Data 1(1):4–es. https://doi.org/10.1145/1217299.1217303
29. Zahn CT (1971) Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans Comput C-20(1):68–86. https://doi.org/10.1109/T-C.1971.223083
30. Chang H, Yeung DY (2008) Robust path-based spectral clustering. Pattern Recognit 41(1):191–203. https://doi.org/10.1016/j.patcog.2007.04.010
31. Veenman CJ, Reinders MJT, Backer E (2002) A maximum variance cluster algorithm. IEEE Trans Pattern Anal Mach Intell 24(9):1273–1280. https://doi.org/10.1109/TPAMI.2002.1033218
32. Jain AK, Law MHC (2005) Data clustering: a user's dilemma. In: Pal SK, Bandyopadhyay S, Biswas S (eds) Pattern recognition and machine intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 1–10
33. Fu L, Medico E (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. BMC Bioinformatics 8(1):3. https://doi.org/10.1186/1471-2105-8-3
34. Fränti P, Sieranoja S (2018) K-means properties on six clustering benchmark datasets. Available from: http://cs.uef.fi/sipu/datasets/
35. De Borda JC (1781) Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences
36. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Toronto, Ontario: University of Toronto, 0
37. LeCun Y, Cortes C, Burges C (2010) MNIST handwritten digit database. ATT Labs [Online] Available: http://yannlecuncom/exdb/mnist, vol 2
38. Settles B (2011) From theories to queries: active learning in practice. In: Guyon I, Cawley G, Dror G, Lemaire V, Statnikov A (eds) Active learning and experimental design workshop in conjunction with AISTATS 2010, vol 16 of proceedings of machine learning research. Sardinia, Italy: PMLR. pp 1–18. Available from: https://proceedings.mlr.press/v16/settles11a.html