JMLR: Workshop and Conference Proceedings 83:1–28, 2018 29th International Conference on Algorithmic Learning Theory

On Similarity Prediction and Pairwise Clustering

Stephen Pasteris S.Pasteris@cs.ucl.ac.uk
Department of Computer Science, University College London, London, UK

Fabio Vitale fabio.vitale@inria.fr
Sapienza University of Rome (Italy) & INRIA Lille (France)

Claudio Gentile cla.gentile@gmail.com
INRIA Lille Nord Europe (France) & Google NY (USA)

Mark Herbster M.Herbster@cs.ucl.ac.uk
Department of Computer Science, University College London, London, UK

Editor: Editor's name

Abstract

We consider the problem of clustering a finite set of items from pairwise similarity information. Unlike what is done in the literature on this subject, we do so in a passive learning setting, and with no specific constraints on the cluster shapes other than their size. We investigate the problem in different settings: i. an online setting, where we provide a tight characterization of the prediction complexity in the mistake bound model, and ii. a standard stochastic batch setting, where we give tight upper and lower bounds on the achievable generalization error. Prediction performance is measured both in terms of the ability to recover the similarity function encoding the hidden clustering and in terms of how well we classify each item within the set. The proposed algorithms are time efficient.

Keywords: Online clustering, similarity, clustering distance, interactive clustering

1. Introduction

In the problem of clustering through pairwise similarity, we have a finite set V of items that have to be suitably clustered into groups by means of information about the similarity/dissimilarity between pairs of such items. This information may come from diverse sources, depending on the specific application. For instance, in the well-known Entity Resolution problem, the goal is to identify and group together records, possibly from different data sources, that refer to the same entity and/or individual, e.g., different pictures of the same person, different ways of addressing the same author of a scientific paper, different names of the same organization, different accounts of the same user of a recommendation service, etc. In these and many other examples of Entity Resolution, one often leverages attributes associated with these entities and, by comparing these attributes, determine very likely matches and mismatches between pairs of entities.

Another related application is community detection in Social Networks, where we naturally view V as the nodes of an edge-signed graph, and we are allowed to observe the sign of some edges carrying information about whether some pairs of nodes/individuals belong to the same community (i.e., they are "similar") or not (they are "dissimilar"). Coarsely speaking, the problem is then trying to reconstruct the communities based on the

observed relationships and, as a byproduct, inferring the sign of new pairwise relationships. Yet, this inference process is clearly prone to errors, for we do not observe all possible pairwise relationships over V. Moreover, small-world phenomena have to be taken into duly account, since this is a typical scenario where the clusters are strongly unbalanced in size, and we would like to be able to take advantage of this pattern as well.

In this paper, we consider the task of building a clustering of a finite set of items V from pairwise similarity/dissimilarity information. This information is assumed to be consistent with an unknown clustering $\mathcal{D} = \{D_1, \ldots, D_k\}$ that we want to reconstruct. We measure the reconstruction error in different ways, depending on whether our algorithms output after training a clustering $\mathcal{C}$ or merely similarity relationship Y (which need not be transitive). In the former case, our metric will be the misclassification error between $\mathcal{C}$ and $\mathcal{D}$, which is essentially the number of items in the learner's output clustering $\mathcal{C}$ that are misclassified, as compared to the ground truth $\mathcal{D}$. In the latter case, our reconstruction error will be the Hamming (distance) error between Y and $\mathcal{D}$ ($\mathcal{D}$ being viewed as a transitive similarity relationship over V), which is essentially the number of item pairs labeled by ground truth $\mathcal{D}$ that are misclassified by $\mathcal{C}$. We investigate two learning settings, an online setting in the mistake bound model (Littlestone, 1987), and a batch stochastic setting with train/test split. In the online setting, we deliver a tight characterization of the predictive complexity of the problem when we have no specific constraints on the cluster shapes other than their size and their number. We give upper and lower mistake bounds, the upper bound being achieved by a novel and efficient algorithm whose predictive bias is towards the existence of a few large clusters and many small ones (just as in the above mentioned community detection scenario). By means of standard online-to-batch conversions, these upper bounds can immediately be turned to generalization error bounds holding in a batch stochastic setting w.r.t. the Hamming error. Yet, our focus in batch stochastic settings is on misclassification error, rather than Hamming error. Assuming randomly drawn training pairs, we prove tight upper and lower bounds on the achievable misclassification error on unseen pairs, the bridge between the two settings being established by a novel reduction turning small Hamming error into small misclassification error.

Related work. The problem we are considering here can be seen as an instance of matrix & metric learning through noise-free labels. The field includes a fair amount of work, hence we can hardly do it justice here. We now outline some of the main contributions in matrix & metric learning, with an emphasis on those we believe are mostly related to the current paper. Relevant work in matrix learning, specifically in online settings, includes Tsuda et al. (2005); Warmuth (2007); Cavallanti et al. (2010); Kakade et al. (2012); Hazan et al. (2012); Gentile et al. (2013); Herbster et al. (2016). In all these works, a major effort is devoted to designing appropriate regularization methods to drive the online optimization process and/or to incorporate available side information, spectral sparseness being of special concern. Yet, the resulting algorithms appear to be too general to deliver tight bounds for the special problem we are considering here and, moreover, their scaling properties make them unfit to practical usage for large problem instances. Early work in the mistake bound setting considered the problem of learning a binary relation over a finite set Goldman et al. (1993); Goldman and Warmuth (1993). This setting generalizes online similarity prediction, but however leads to weaker bounds for similarity prediction. Metric learning is the special case of matrix learning where the matrix to be learned is positive semi-definite.

Representative works include Xing et al. (2002); Shalev-Shwartz et al. (2004); Maurer (2008); Cao et al. (2016). In particular, Cao et al. (2016) work out generalization bounds based on a Rademacher complexity analysis which is again too general to deliver tight bounds for our specific problem; besides, they essentially consider only a Hamming distance-like error. In short, what distinguishes our work from the above previous work is that we produce not just similarity functions, but instead construct clusterings with an associated tight misclassification error analysis.

The so-called matrix completion task is also related to our work (see, e.g., Koltchinskii et al. (2016) for a representative work, as well as references therein). Yet, the typical goal there is to come up with matrix recovery methods whose error rates are again measured through a Hamming distance-like error (i.e., the Frobenious norm distance), with no specific concerns in clustering.

Closer in spirit to our work is the general line of research on semi-supervised clustering/clustering with side information, where the must-link/cannot-link constraints can be seen as similarity feedback. Known references, mostly application-oriented, include Ben-Dor et al. (1999); Demiriz et al. (1999); Basu et al. (2004); Kulis et al. (2009), the formal statements therein being fairly different from ours. Somewhat closer to our paper from the formal standpoint within semi-supervised clustering is the thread on interactive clustering/clustering with queries. Here, the learner is allowed to interactively ask for feedback in the form of suitable *queries*, e.g., split and merge queries (Balcan and Blum, 2008; Awasthi et al., 2017), and similarity/same-cluster queries (Davidson et al., 2014; Ashtiani et al., 2016; Mazumdar and Saha, 2017a,b). Unlike our paper, in many of these works, the goal is to achieve *exact*, rather than approximate, reconstruction of the ground-truth clustering by asking as few queries as possible. As a sample of the available results in this literature, in the interactive feedback paper of Balcan and Blum (2008), the authors focus on clusters having specific shapes, that is, coming from a specific collection $\mathcal{B}$ of subsets of V. They show that the number of queries that suffice is either constant or logarithmic in n, but is also dependent on the "descriptional" complexity of $\mathcal{B}$, e.g., if V is a set of n points on the real line and $\mathcal{B}$ is the set of intervals, then $k \log n$ queries suffice; more generally, $\mathcal{O}(k^3 \log |\mathcal{B}|)$ queries are enough when computationally inefficient algorithms are also considered. Yet, generalizing their argument to clusters of arbitrary shapes (i.e., $\mathcal{B} = 2^V$), as we have them here, easily leads in general to the trivial bound of $\mathcal{O}(n)$ queries. Davidson et al. (2014) show that $\Theta(k\,n)$ similarity queries are both necessary and sufficient to achieve exact reconstruction, the lower bound holding specifically in the case of (almost) equally-sized clusters. Yet, unlike our paper, they work in an active learning setting. Still in the active setting, Mazumdar and Saha (2017a,b) consider ways to sharpen the above bound by means of further side information available to the learner beyond queries. Their results are generally incomparable to ours, for besides dealing with active learning and exact reconstruction, they also allow for similarity labels to be noisy.

Finally, correlation clustering (Bansal et al., 2004) is also similar in flavor to some of our results. In correlation clustering, a similarity relationship Y on item pairs is given, and the goal is to find a clustering $\mathcal{C}$ which minimizes the Hamming error between Y and $\mathcal{C}$. The problem, as stated, is NP-hard, and a number of approximation algorithms exist (e.g., Bansal et al. (2004); Demaine et al. (2006)). Although inspired by these results as well,

our focus here is slightly different: we seek to provide efficient algorithms that compute a clustering with associated predictive performance guarantees.

2. Preliminaries and Notation

We now introduce our main notation along with basic preliminaries. Given a finite set $V = \{1, \ldots, n\}$, a *clustering* $\mathcal{D}$ over V is a partition of V into sets $\mathcal{D} = \{D_1, \ldots, D_k\}$. Each D_j is called a *cluster*. A *similarity* graph $G = (V, \mathcal{P})$ over V is an undirected (but not necessarily connected) graph where, for each pairing $(v, w) \in V^2$, v and w are *similar* if $(v, w) \in \mathcal{P}$, and *dissimilar*, otherwise. Notice that the similarity relationship so defined need not be transitive. We shall interchangeably represent a similarity graph over V through a binary $n \times n$ *similarity* matrix $Y = [y_{v,w}]_{v,w=1}^{n \times n}$ whose entry $y_{v,w}$ is 1 if items v and w are similar, and $y_{v,w} = 0$, otherwise. A clustering $\mathcal{D}$ over V can be naturally associated with a similarity graph $G = (V, \mathcal{P}_\mathcal{D})$ whose edge set $\mathcal{P}_\mathcal{D}$ is defined as follows: given $v, w \in V$, then $(v, w) \in \mathcal{P}_\mathcal{D}$ if and only if there exists a cluster $D \in \mathcal{D}$ with $v, w \in D$. In words, G is made up of k disjoint cliques. It is only in this case that the similarity relationship defined through G is transitive. Matrix Y represents a clustering if, after permutation of rows and columns, it ends up being block-diagonal, where the i-th block is a $d_i \times d_i$ matrix of ones, d_i being the size of the i-th cluster. Given clustering $\mathcal{D}$, we find it convenient to define a map $\mu_\mathcal{D} : V \to \{1, \ldots, k\}$ in such a way that for all $v \in V$ we have $v \in D_{\mu_\mathcal{D}(v)}$. In words, $\mu_\mathcal{D}$ is a class assignment mapping, so that v and w are similar w.r.t. $\mathcal{D}$ if and only if $\mu_\mathcal{D}(v) = \mu_\mathcal{D}(w)$.

Given two similarity graphs $G = (V, \mathcal{P})$ and $G' = (V, \mathcal{P}')$, the Hamming (distance) error between G and G', denoted here as $\mathrm{HA}(\mathcal{P}, \mathcal{P}')$, is defined as

$$\mathrm{HA}(\mathcal{P}, \mathcal{P}') = \left|\{(v, w) \in V^2 \,:\, (v, w) \in \mathcal{P} \wedge (v, w) \notin \mathcal{P}' \vee (v, w) \notin \mathcal{P} \wedge (v, w) \in \mathcal{P}'\}\right| ,$$

where $|A|$ is the cardinality of set A. The same definition applies in particular to the case when either G or G' (or both) represent clusterings over V. By abuse of notation, if $\mathcal{D}$ is a clustering and $G = (V, \mathcal{P})$ is a similarity graph, we will often write $\mathrm{HA}(\mathcal{D}, \mathcal{P})$ to denote $\mathrm{HA}(\mathcal{P}_\mathcal{D}, \mathcal{P})$, where $(V, \mathcal{P}_\mathcal{D})$ is the similarity graph associated with $\mathcal{D}$, so that $\mathrm{HA}(\mathcal{P}_\mathcal{D}, \mathcal{D}) = 0$. Moreover, if the similarity graphs G and G' are represented by similarity matrices, we may equivalently write $\mathrm{HA}(Y, Y')$, $\mathrm{HA}(Y, \mathcal{D})$, and so on. The quantity $\mathrm{HA}(,)$ is very closely related to the so-called Mirkin metric (Mirkin, 1996) over clusterings, as well as to the (complement of the) Rand index (Rand, 1971), see, e.g., Meila (2012).

Another "distance" that applies specifically to clusterings is the misclassification error distance, denoted here as $\mathrm{ER}(,)$, and defined as follows. Given two clusterings $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ and $\mathcal{D} = \{D_1, \ldots, D_k\}$ over V, repeatedly add the empty set to the smaller of the two so as to obtain $\ell = k$. Then

$$\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \min_f \sum_{D \in \mathcal{D}} |D \setminus f(D)| ,$$

the minimum being over all bijections from $\mathcal{D}$ to $\mathcal{C}$. In words, $\mathrm{ER}(\mathcal{C}, \mathcal{D})$ measures the smallest number of classification mistakes over all class assignments of clusters in $\mathcal{D}$ w.r.t. clusters in $\mathcal{C}$. This is basically an unnormalized version of the classification error (distance) considered, e.g., in Meila (2007). An illustrative example to gain intuition on the ER metric is the following: Suppose $V = \{1, 2, 3, 4, 5, 6\}$, and we have clusterings $\mathcal{C} = \{\{1, 2, 3\}, \{4, 5\}, \{6\}\}$ and $\mathcal{D} = \{\{1, 2\}, \{3, 4\}, \{5\}, \{6\}\}$. In order to turn $\mathcal{C}$ into $\mathcal{D}$, we operate on $\mathcal{C}$ by moving

item 3 from the first cluster $\{1,2,3\}$ to the second cluster $\{4,5\}$, and item 5 from the second cluster $\{4,5\}$ to a new cluster. Since this is the minimal number of moves we need to make, we have $\mathrm{ER}(\mathcal{C},\mathcal{D}) = 2$.

Finally, the (Jaccard) distance $\textsc{dist}(A,B)$ between sets A and B, with $A,B \subseteq V$ is defined as

$$\textsc{dist}(A,B) = \frac{|A \setminus B| + |B \setminus A|}{|A \cup B|} .$$

Recall that $\textsc{dist}(,)$ is a proper metric on the collection of all finite sets. Moreover, observe that $\textsc{dist}(A,B) = 1$ if and only if A and B are disjoint.

2.1. Learning settings

We are interested in predicting similarities and/or computing clusterings over $V = \{1,\ldots,n\}$ based on binary similarity/dissimilarity information contained in a similarity matrix Y, whose entries are only partially observed. In particular, we are given a training set S of m binary-labeled pairs $\langle (v,w), y_{v,w}\rangle \in V^2 \times \{0,1\}$, drawn uniformly at random[1] from V^2, and our goal is to either build a clustering $\mathcal{C}$ over V so as to achieve small misclassification error $\mathrm{ER}(\mathcal{C},Y)$, or a similarity prediction model (that is, an estimated similarity matrix) $\hat{Y}$ so as to achieve small Hamming error $\mathrm{HA}(\hat{Y},Y)$. In both cases, the error is computed *on the whole* matrix Y (clearly enough, this is sensible only when m is significantly smaller than n^2). The similarity matrix Y will always be consistent with a given unknown clustering $\mathcal{D}$ over V. Notice that the number of clusters k will also be unknown to the prediction/clustering algorithms. Hence, if the goal is to simply predict similarities over V, the error of our inference procedures will be measured through HA. On the other hand, if our final goal is not to merely predict similarities, but to produce a clustering over V then our error will be measured through ER. In fact, we shall do the above in different ways, corresponding to the content of the next three sections:

1. By investigating the problem of online prediction of pairwise similarities in the standard mistake bound model of online learning (Littlestone, 1987). Using a simple online-to-batch conversion, we will make the online algorithm compute a similarity prediction model $\hat{Y}$ for the unseen entries of Y. Since $\hat{Y}$ itself is not guaranteed to be a clustering over V, error in this case will be measured through $\mathrm{HA}(\hat{Y},Y)$.

2. By an indirect approach that exhibits a (tight) reduction from similarity prediction methods measured through HA (like those coming from Item 1 above) to clustering methods measured through ER. The machinery developed in Item 1 may thus be a possible input to this reduction.

3. Through a direct approach, by presenting a specific baseline algorithm whose goal is to build a clustering $\mathcal{C}$ over V based on a training set S drawn at random, and the discrepancy between $\mathcal{C}$ and $\mathcal{D}$ will be measured through $\mathrm{ER}(\mathcal{C},\mathcal{D})$.

1. For simplicity of presentation, we will assume throughout that, when drawn at random from V, the samples in S are drawn with replacement.

3. Online Similarity Prediction

In this section, we consider similarity prediction in the online mistake bound model, where an example sequence $S = \langle (v_1, w_1), y_{v_1,w_1} \rangle, \ldots, \langle (v_m, w_m), y_{v_m,w_m} \rangle$ is revealed incrementally in rounds. In the t-th round the algorithm is compelled to predict the similarity label $y_{v_t,w_t} \in \{0, 1\}$, given the previous $t-1$ examples $\langle (v_1, w_1), y_{v_1,w_1} \rangle, \ldots, \langle (v_{t-1}, w_{t-1}), y_{v_{t-1},w_{t-1}} \rangle$, and (v_t, w_t). Denote by $\hat{y}_t \in \{0, 1\}$ the prediction issued by the algorithm in round t. After prediction, the true label $y_t = y_{v_t,w_t}$ is revealed, and we say that the algorithm has made a prediction mistake if $\hat{y}_t \neq y_t$. The aim is to minimize the number of mistaken predictions on any example sequence S. Sequence S can be generated by an adaptive adversary, but the labels y_{v_t,w_t} are assumed to be consistent with an underlying clustering $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ over V, with cluster sizes $d_1, d_2, \ldots, d_k$. Throughout, we assume w.l.o.g. that $d_1 \leq d_2 \leq \ldots \leq d_k$.

This problem has been studied before, and can be cast as a special case of online matrix/metric learning (e.g., Shalev-Shwartz et al. (2004); Tsuda et al. (2005); Warmuth (2007); Cavallanti et al. (2010); Kakade et al. (2012); Hazan et al. (2012); Gentile et al. (2013)). Yet, a direct application of techniques available from those papers would give rise to suboptimal results in terms of both mistake bound and running time. For instance, we may use the adaptation of the Matrix Winnow algorithm (Warmuth, 2007) from Gentile et al. (2013), that provides a mistake bound of the form $|\Phi^G| \left(\max_{v,w \in V} R^G_{v,w}\right) \log n$, where $|\Phi^G|$ is the cutsize determined by a graph G over V, and the $R^G_{v,w}$ are the corresponding effective resistance factors. If G therein is the complete graph[2] then it is easy to see that $|\Phi^G| = \sum_{i=1}^k d_i(n - d_i) = n^2 - \sum_{i=1}^k d_i^2$, and all effective resistance factors $R^G_{v,w}$ are $2/n$. Up to multiplicative constants, this results in a mistake bound of the form

$$\left(n - \frac{1}{n} \sum_{i=1}^{k} d_i^2\right) \log n \,. \tag{1}$$

Moreover, the running time per round of this algorithm is $\mathcal{O}(n^3)$, since it requires to maintain and update at every round the SVD of a $n \times n$ matrix.

An even simpler baseline (belonging to folklore) – see pseudocode in Algorithm 1 – is one where the algorithm maintains on the fly a spanning forest over V (i.e., a collection of disjoint trees covering V, each tree being a cluster), and at round t the algorithm predicts 1–"similar" on (v_t, w_t) if v_t and w_t belong to the same tree, and 0–"dissimilar", otherwise. If the actual label y_{v_t,w_t} is 1 and the algorithm is mistaken, then the two nodes v_t and w_t get connected by an edge, thereby merging two trees of the current forest into a bigger tree. It is easy to argue that if we start off from the degenerate forest made up of n singletons (as in Algorithm 1), this algorithm makes at most $n - k$ mistakes. Thus this algorithm has a predictive bias towards many "small" clusters.

Given the above state of affairs, a natural question is what is the optimal mistake bound for the problem of online learning k clusters of sizes $d_1 \leq d_2 \leq \ldots \leq d_k$ over $V = \{1, \ldots, n\}$ through pairwise similarity labels. In the rest of this section, we describe and analyze the time-efficient algorithm Oppa (Online Pairwise Prediction Algorithm) which incorporates a different predictive bias than the folklore algorithm mentioned above. We then complement

2. This choice of G seems best in the absence of further information.

Algorithm 1 Folk online clustering algorithm.

Input: Item set $V = \{1, \ldots, n\}$.
Initialization:

- $\mathcal{C} = \{\{v\} : v \in V\}$; `// Let` C_v `denote the cluster containing` v.

For $t = 1, \ldots, m$:

1. Get pair $(v_t, w_t) \in V^2$;
2. If $C_{v_t} = C_{w_t}$ then $\hat{y}_t = 1$, else $\hat{y}_t = 0$:
3. Observe $y_t := y_{v_t, w_t}$; if $\hat{y}_t \neq y_t$ then $\mathcal{C} \leftarrow \mathcal{C} \setminus C_w$ and $C_v \leftarrow C_v \cup C_w$;

Output: Clustering $\mathcal{C}$.

our analysis by showing through a lower bounding argument that no online algorithm can do better in terms of mistake bounds up to constant factors, thereby showing that Oppa is essentially optimal.

Oppa is described in Algorithm 2. Whereas the folklore algorithm is predictively biased to the case of many small clusters (large k). Oppa is predictively biased to the case of few large "majority" clusters (implying a large d_k). Thus the algorithms incorporate distinct predictive biases which are indirectly in opposition. In common with the folklore algorithm we maintain a clustering $\mathcal{C}$, and we predict consistently so that when two points are known to be in the same cluster we predict "similar." However, unlike the folklore algorithm if we do not know the two points to be "similar" we do not necessarily predict "dissimilar", in fact if it is the first time we have encountered both points we predict "similar." Intuitively this follows from the different predictive biases, as e.g., if we make a mistake on two novel points we now know the cardinality d_k of the majority cluster is reduced by one. Oppa maintains both a clustering $\mathcal{C}$ and a tag in $\{\emptyset, A, B\}$ for each cluster $C \in \mathcal{C}$, the tags controlling both prediction and updates, as seen in Algorithm 2. The tagging through an amortized analysis (see proof of Theorem 1) enables one to prove that the potential cardinality of the majority cluster is reduced by one every five mistakes. Both the folklore algorithm and Oppa can be implemented with a standard disjoint set data-structure ensuring a cumulative time complexity for m predictions of $\mathcal{O}((n+m)\log^* n)$ for both algorithms.

We have the following result.[3]

Theorem 1 *Let $\mathcal{D} = \{D_1, \ldots, D_k\}$ be any clustering of $V = \{1, \ldots, n\}$, with $d_i = |D_i|$, $i = 1, \ldots, k$, and $d_1 \leq d_2 \leq \ldots d_k$. Then for any sequence of examples S labeled according to $\mathcal{D}$, the number of mistakes made by* Oppa *on S is upper bounded by $5(n - d_k)$.*

It is useful to contrast the upper bound for Oppa in Theorem 1 to the bound for Matrix Winnow in (1). Since $\sum_{i=1}^k d_i^2 \leq \left(\sum_{i=1}^k d_i\right)(\max_{i=1\ldots k} d_i) = nd_k$, one can readily see that (1) is weaker than Theorem 1 by at least a logarithmic factor. On the other hand, the latter is again generally incomparable to the folklore bound $n - k$. By constrast, the following (almost matching) lower bound holds.

3. All proofs are given in the appendix.

Algorithm 2 The Online Pairwise Prediction Algorithm (Oppa).

Input: Set $V = \{1, \ldots, n\}$.
Initialization:

- $\mathcal{C} = \{\{v\} : v \in V\}$; `// Let` C_v `denote the cluster containing` v.
- For any $v \in V$, set tag $\omega(C_v) \leftarrow \emptyset$.

For $t = 1, \ldots, m$:

1. Get pair $(v_t, w_t) \in V^2$;
2. If $C_{v_t} = C_{w_t}$ then $\hat{y}_t = 1$; `//predict "similar"`
3. Else define ζ as the pair of tags $(\omega(C_{v_t}), \omega(C_{w_t}))$; then predict as in the table (second column) and, upon observing y_t, if $\hat{y}_t \neq y_t$ update as in the table (third column):

Value of ζ	**Prediction**	**On mistake execute:**
$(\emptyset, \emptyset)$	Similar ($\hat{y}_t = 1$)	$\zeta \leftarrow (A, A)$
$(\emptyset, A)$	Similar ($\hat{y}_t = 1$)	$\zeta \leftarrow (A, B)$
$(A, \emptyset)$	Similar ($\hat{y}_t = 1$)	$\zeta \leftarrow (B, A)$
$(\emptyset, B)$	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_{v_t}, C_{w_t})$
$(B, \emptyset)$	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_{v_t}, C_{w_t})$
(A, B)	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_{v_t}, C_{w_t})$
(B, A)	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_{v_t}, C_{w_t})$
(A, A)	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_{v_t}, C_{w_t})$
(B, B)	Dissimilar ($\hat{y}_t = 0$)	$\mathcal{C} \leftarrow \text{MERGE}_B(\mathcal{C}, C_{v_t}, C_{w_t})$

Algorithm 3 The $\text{MERGE}_A(,,)/\text{MERGE}_B(,,)$ subroutine.

Input: Clustering $\mathcal{C}$, clusters $C, C' \in \mathcal{C}$.

1. Set $C^* \leftarrow C \cup C'$;
2. Set $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C, C'\}) \cup \{C^*\}$;
3. If operation is $\text{MERGE}_A(\mathcal{C}, C, C')$ set $\omega(C^*) \leftarrow A$;
4. If operation is $\text{MERGE}_B(\mathcal{C}, C, C')$ set $\omega(C^*) \leftarrow B$;

Output: Clustering $\mathcal{C}$, tagging $\omega(\cdot)$.

Theorem 2 *Let $\mathcal{D} = \{D_1, \ldots, D_k\}$ be any clustering of $V = \{1, \ldots, n\}$, with $d_i = |D_i|$, $i = 1, \ldots, k$, and $d_1 \leq d_2 \leq \ldots d_k$. Then, for any online similarity prediction algorithm A, a sequence of examples S exists which is labeled according to $\mathcal{D}$, and such that the number of mistakes made by A on S is at least $n - k - d_k$.*

Theorems 1 and 2 combined together provide a reasonably tight characterization of the complexity of online learning clusterings through pairwise similarity labels.

Randomly drawn sequences. Suppose now that the training sequence S is drawn uniformly at random. According to a standard online-to-batch conversion (Helmbold and Warmuth, 1995), one can pick at random one of the m similarity prediction models[4] produced by Oppa during its online functioning, call this model $\hat{Y}_S$, and have a guarantee that the probability of making a mistake on an unseen pair (v, w) drawn again uniformly at random is $\mathcal{O}(\frac{n-d_k}{m})$. The probability also takes into account the random draw of S and the random draw of the model within the sequence of m models. Then, recalling the definition of $\mathrm{HA}(,)$ from Section 2, the upper bound in Theorem 1 immediately entails the following.

Corollary 3 *Let $\mathcal{D} = \{D_1, \ldots, D_k\}$ be any clustering of $V = \{1, \ldots, n\}$, with $d_i = |D_i|$, $i = 1, \ldots, k$, and $d_1 \leq d_2 \leq \ldots d_k$. Let $S = \langle (v_1, w_1), y_{v_1,w_1} \rangle, \ldots, \langle (v_m, w_m), y_{v_m,w_m} \rangle$ be labeled according to $\mathcal{D}$, and be such that (v_t, w_t) are drawn uniformly at random from V^2. Moreover, let $\hat{Y}_t$ be the similarity prediction model produced by* Oppa *at the beginning of round t, for $t = 1, \ldots, m$. If $\hat{Y}_S$ is drawn uniformly at random from the sequence $\hat{Y}_1, \ldots, \hat{Y}_m$, then*

$$\mathbb{E}\,\mathrm{HA}(\hat{Y}_S, Y) = \mathcal{O}\left(\frac{n^2}{m}\,(n - d_k)\right). \tag{2}$$

In the above, the expectation is over the random draw of S and the random draw of $\hat{Y}_S$ from the sequence $\hat{Y}_1, \ldots, \hat{Y}_m$.

4. From HA to ER

This section exhibits a clustering algorithm that takes as input a similarity graph $(V, \mathcal{P})$ (like one produced after a training phase as in Corollary 3 of Section 3) and gives in output a clustering $\mathcal{C}$ over V. The primary goal here is to show that for any other clustering $\mathcal{D}$ over V, $\mathrm{ER}(\mathcal{C}, \mathcal{D})$ is tightly related to $\mathrm{HA}(\mathcal{P}, \mathcal{D})$. This algorithm will be a building block for later results, but it can also be of independent interest.

Our algorithm, called Robust Greedy Clustering Algorithm (rgca, for brevity), is displayed in Algorithm 4. The algorithm has two stages. The first stage is a robustifying stage where the similarity graph $(V, \mathcal{P})$ is converted into a (more robust) similarity graph $(V, \mathcal{Q})$ as follows: Given two distinct vertices $v, w \in V$, we have $(v, w) \in \mathcal{Q}$ if and only if the Jaccard distance of their neighbourhoods (in $(V, \mathcal{P})$) is not bigger than $1 - a$, for some distance parameter $a \in [0, 1]$. The second stage uses a greedy method to convert the graph $(V, \mathcal{Q})$ into a clustering $\mathcal{C}$. This stage proceeds in "rounds". At each round t we have a set A_t of all vertices which have not yet been assigned to any clusters. We then choose α_t to be the vertex in A_t which has the maximum number of neighbours (under the graph $(V, \mathcal{Q})$) in A_t, and take this set of neighbours (including α_t) to be the next cluster.

From a computational standpoint, the second stage of rgca runs in $\mathcal{O}(n^2 \log n)$ time, since on every round t we single out α_t (which can be determined in $\log n$ time by maintaining a suitable heap data-structure), and erase all edges emanating from α_t in the similarity

4. Unlike the folklore algorithm, Oppa is not guaranteed to deliver after training a prediction function which is itself a clustering over V.

Algorithm 4 The Robust Greedy Clustering Algorithm

Input: Similarity graph $(V, \mathcal{P})$; distance parameter $a \in [0, 1]$.

1. For all $v \in V$, set $\Gamma(v) \leftarrow \{v\} \cup \{w \in V : (v, w) \in \mathcal{P}\}$;
2. Construction of graph $(V, \mathcal{Q})$: `//First stage`
   For all $v, w \in V$ with $v \neq w$:
   If $\text{DIST}\,(\Gamma(v), \Gamma(w)) \leq 1 - a$ then $(i, j) \in \mathcal{Q}$, otherwise $(i, j) \notin \mathcal{Q}$;
3. Set $A_1 \leftarrow V$, and $t \leftarrow 1$; `//Second stage`
4. While $A_t \neq \emptyset$:
   - For every $v \in A_t$ set $N_t(v) \leftarrow \{v\} \cup \{w \in A_t : (v, w) \in \mathcal{Q}\}$,
   - Set $\alpha_t \leftarrow \text{argmax}_{v \in A_t} |N_t(v)|$,
   - Set $C_t \leftarrow N_t(\alpha_t)$,
   - Set $A_{t+1} \leftarrow A_t \setminus C_t$,
   - $t \leftarrow t + 1$;

Output: $C_1, C_2, ..., C_\ell$, where $\ell = t - 1$.

graph $(V, \mathcal{Q})$. On the other hand, the first stage of RGCA runs in $\mathcal{O}(n^3)$ time, in the worst case, though standard techniques exist that avoid the all-pairs comparison, like a Locality Sensitive Hashing scheme applied to the Jaccard distance (e.g., (Rajaraman and Ullman, 2010, Ch.3)). We have the following result.

Theorem 4 *Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be the clustering produced in output by RGCA when receiving as input similarity graph $(V, \mathcal{P})$, and distance parameter $a = 2/3$. Then for any clustering $\mathcal{D} = \{D_1, \ldots, D_k\}$, with $d_i = |D_i|$, $i = 1, \ldots, k$, and $d_1 \leq d_2 \leq \ldots d_k$ we have*

$$\text{ER}(\mathcal{C}, \mathcal{D}) \leq \min_{j=1,\ldots,k} \left(\frac{12}{d_j} \text{HA}(\mathcal{P}, \mathcal{D}) + \sum_{i=1}^{j-1} d_i \right) .$$

Hence, if the chosen $\mathcal{D}$ is the best approximation to $\mathcal{P}$ w.r.t. HA(,), and we interpret $(V, \mathcal{P})$ as a noisy version of $\mathcal{D}$, then small $\text{HA}(\mathcal{P}, \mathcal{D})$ implies small $\text{ER}(\mathcal{C}, \mathcal{D})$. In particular, $\text{HA}(\mathcal{P}, \mathcal{D}) = 0$ implies $\text{ER}(\mathcal{C}, \mathcal{D}) = 0$ (simply pick $j = 1$ in the minimum). Notice that this result only applies to the case when the similarity graph $(V, \mathcal{P})$ is fully observed by our clustering algorithm. As we already said, $(V, \mathcal{P})$ may in turn be the result of a similarity learning process when the similarity labels are provided by an unknown clustering $\mathcal{D}$. In this sense, Theorem 4 will help us delivering generalization bounds (as measured by $\text{ER}(\mathcal{C}, \mathcal{D})$), as a function of the generalization ability of this similarity learning process (as measured by $\text{HA}(\mathcal{P}, \mathcal{D})$).

The problem faced by RGCA is also related to the standard correlation clustering problem (Bansal et al., 2004). Yet, the goal here is somewhat different, since a correlation clustering algorithm takes as input $(V, \mathcal{P})$, but is aimed at producing a clustering $\mathcal{C}$ such that $\text{HA}(\mathcal{P}, \mathcal{C})$ is as small as possible.

In passing, we next show that the construction provided by RGCA is essentially optimal (up to multiplicative constants). Let $G_{\mathcal{D}} = (V, E_{\mathcal{D}})$ be the similarity graph associated with clustering $\mathcal{D}$. We say that a clustering algorithm that takes as input a similarity graph over V and gives in output a clustering over V is *consistent* if and only if for every clustering $\mathcal{D}$ over V the algorithm outputs $\mathcal{D}$ when receiving as input $G_{\mathcal{D}}$. Observe that RGCA is an example of a consistent algorithm. We have the following lower bound.

Theorem 5 *For any finite set V, any clustering $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ over V, any positive constant σ, and any consistent clustering algorithm, there exists a similarity graph $(V, \mathcal{P})$ such that $\mathrm{HA}(\mathcal{P}, \mathcal{D}) \leq \sigma$, while*

$$\mathrm{ER}(\mathcal{C}, \mathcal{D}) \geq \min_{j=1,\ldots,k} \left(\frac{1}{2d_j}\,\sigma - 1 + \frac{1}{4}\sum_{i=1}^{j-1} d_i \right) , \tag{3}$$

or $\mathrm{ER}(\mathcal{C}, \mathcal{D}) \geq \frac{n}{2}$, where $\mathcal{C}$ is the output produced by the algorithm when given $(V, \mathcal{P})$ as input, and $d_i = |D_i|$, $i = 1, \ldots, k$, with $d_1 \leq d_2 \leq \ldots d_k$.

From the proof given in the appendix, one can see that the similarity graph $(V, \mathcal{P})$ used here is indeed a *clustering* over V so that, as the algorithm is consistent, the output $\mathcal{C}$ must be such a clustering. This result can therefore be contrasted to the results contained, e.g., in Meila (2012) about the equivalence between clustering distances, specifically Theorem 26 therein. Translated into our notation, that result reads as follows: $\mathrm{ER}(\mathcal{C}, \mathcal{D}) \geq \frac{\mathrm{HA}(\mathcal{P}, \mathcal{D})}{16 d_k}$. Our Theorem 5 is thus sharper but, unlike the one in Meila (2012), it *does not apply* to any possible pairs of clusterings $\mathcal{C}$ and $\mathcal{D}$, for in our case $\mathcal{C}$ is selected as a function of $\mathcal{D}$.

We finally tie things together. The similarity graph $(V, \mathcal{P})$ in input to RGCA may for instance be the predictor learned by a similarity learning algorithm, like Matrix Winnow (Warmuth, 2007) or OPPA (Section 3). Since none of the two algorithms output a clustering over V (despite the training labels are indeed consistent with a ground truth clustering $\mathcal{D}$), this is where our reduction RGCA comes into play. For the sake of concreteness, suppose that $(V, \mathcal{P})$ is the model $\hat{Y}_S$ generated by the online-to-batch conversion of Corollary 3 applied to OPPA after seeing m-many randomly drawn pairs (v_t, w_t) labeled according to $\mathcal{D}$, and let $\mathcal{C}$ be the corresponding clustering output by RGCA. Then, combining Theorem 4 with Corollary 3, we conclude that $\mathcal{C}$ satisfies

$$\begin{aligned}
\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) &\leq \mathbb{E}\left[\min_{j=1,\ldots,k} \left(\frac{12}{d_j}\,\mathrm{HA}(\mathcal{P}, \mathcal{D}) + \sum_{i=1}^{j-1} d_i \right) \right] \\
&\leq \min_{j=1,\ldots,k} \left(\frac{12}{d_j}\,\mathbb{E}\,\mathrm{HA}(\mathcal{P}, \mathcal{D}) + \sum_{i=1}^{j-1} d_i \right) \\
&= \mathcal{O}\left(\min_{j=1,\ldots,k} \left(\frac{1}{d_j}\,\frac{n^2}{m}\,(n - d_k) + \sum_{i=1}^{j-1} d_i \right) \right) ,
\end{aligned} \tag{4}$$

which is our generalization bound in the ER metric. As an example, if $d_i = n/k$ for all i, then we can pick $j = 1$ in Eq. (4) to achieve

$$\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \mathcal{O}\left(\frac{k\,n^2}{m} \right) . \tag{5}$$

On the other hand, if $\mathcal{D}$ has few big clusters and many small ones the resulting bound looks significantly different. As an extreme situation, let for concreteness $d_1 = d_2 = \dots d_{k-h} = 1$, and $d_{k-h+1} = d_{k-h+2} = \dots d_k = \frac{n-k+h}{h}$, for some small h such that $2 \leq h < k$. Then picking $j = k - h + 1$ in (4) yields $\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \mathcal{O}\left(\frac{h}{n-k+h}\,\frac{n^3}{m} + k\right)$, and if, say, $h = 3$ and k is either a constant ≥ 3 or $k = o(n)$, then we have

$$\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \mathcal{O}\left(\frac{n^2}{m} + k\right) \ , \tag{6}$$

as n grows large. Notice that (6) is typically smaller than (5), take for instance $k = \sqrt{n}$, and $m = n^{3/2}$: whereas Eq. (5) gives $\mathcal{O}(n)$ (which is vacuous, up to multiplicative constants, for this size of m), Eq. (6) yields a bound of $\mathcal{O}(\sqrt{n})$.

Next, we take a more direct route to obtain alternative ER-based statistical guarantees.

5. A direct approach to bounding ER and a lower bound

In the previous section, we showed an indirect route to prove ER bounds. This route specifically applies to algorithms that do not output a clustering after training, but only a similarity model. Yet, there are algorithms, like the folk online clustering algorithm of Section 3 (Algorithm 1), that do indeed produce a clustering at the end of each online round. Hence, one is wondering whether a direct analysis in the ER metric for such algorithms can be carried out. In the next theorem, we show a simple generalization bound achieved by Algorithm 1 after training with m randomly drawn examples. Unlike the one in Eq. (4), we have been unable to obtain a meaningful dependence on the cluster sizes d_i.

Theorem 6 *Let $\mathcal{D} = \{D_1, \dots, D_k\}$ be any clustering of $V = \{1, \dots, n\}$, with $d_i = |D_i|$, $i = 1, \dots, k$, and $d_1 \leq d_2 \leq \dots d_k$. Let $S = \langle (v_1, w_1), y_{v_1,w_1} \rangle, \dots, \langle (v_m, w_m), y_{v_m,w_m} \rangle$ be labeled according to $\mathcal{D}$, and be such that (v_t, w_t) are drawn uniformly at random from V^2. Then Algorithm 1 returns a clustering $\mathcal{C}$ such that $\mathrm{ER}(\mathcal{C}, \mathcal{D})$ is bounded as follows:*

$$\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \mathcal{O}\left(\frac{k\,n^2}{m}\,\log\frac{n^2}{m}\right) \ ,$$

the expectation with respect to a random draw of S.

It is instructive to compare the upper bounds contained in Theorem 6 to the one in Eq. (4), as specialized in (5) and (6). The bound in Theorem 6 tends to be weaker (by at least a log factor – compare to the equal-size case in (5)). Moreover, it does not show an explicit dependence on the cluster sizes d_i, making it unable to leverage cluster unbalancedness, like in (6). On the other hand, the folk algorithm is definitely much faster to run than the combination RGCA + OPPA.

To close this section, we complement our upper bounds in Eqs. (4)-(6), and Theorem 6 by the following lower bound result, showing that the dependence of $\mathrm{ER}(,)$ on k and n^2 cannot in general be eliminated.

Theorem 7 *Given any $k > 2$ and any $m < \frac{n^2}{4}$, there exists a clustering $\mathcal{D}$ of at most k clusters such that, for any algorithm having as input a randomly drawn training sequence of length m labeled according to $\mathcal{D}$ and giving in output clustering $\mathcal{C}$, we have that $\mathbb{E}\,\mathrm{ER}(\mathcal{C}, \mathcal{D}) = \Omega\left(\min\left\{\frac{k\,n^2}{m}, n\right\}\right)$, the expectation with respect to a random draw of the training sequence.*

6. Conclusions and Ongoing Research

We have investigated the general problem of learning a clustering over a finite set from pairwise similarity labels, with no specific constraints on the cluster shapes, other than their size. We did so in two settings:

1. An online setting, where we exhibited a novel characterization of the complexity of learning the clustering in the mistake bound model of online learning;

2. In a batch stochastic setting, where we took either an indirect route that steps through a reduction, called RGCA, establishing a tight bridge between the two clustering metrics HA and ER, or a direct route, where we have shown as a yardstick the kind of misclassification error bounds a standard baseline may achieve for this problem.

Finally, we complemented the above results with an almost matching lower bound that applies to the ER metric when the training sequence is randomly drawn.

Two extensions we are currently exploring are: i. extending the underlying statistical assumptions on data (e.g., sampling distribution-free guarantees) while retaining running time efficiency, and ii. studying other learning regimes, like active learning, under similar or broader statistical assumptions as those currently in this paper, possibly with side-information, as in, e.g., Mazumdar and Saha (2017a,b).

Acknowledgments

SP and MH have been supported in part by the U.S. Army Research Laboratory and the U.K. Defence Science and Technology Laboratory. This work was accomplished under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, ether expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Defence Science and Technology Laboratory or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. FV was supported by the ERC Starting Grant DMAP 680153. CG was partially supported by a Criteo Faculty Research Award.

References

H. Ashtiani, S. Kushagra, and S. Ben-David. Clustering with same-cluster queries. In *Proc. 30th NIPS*, 2016.

P. Awasthi, M. F. Balcan, and K. Voevodski. Local algorithms for interactive clustering. *Journal of Machine Learning Research*, 18, 2017.

M. F. Balcan and A. Blum. Clustering with interactive feedback. In *Proc. of the 19th International Conference on Algorithmic Learning Theory*, pages 316–328, 2008.

N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1):89–113, 2004.

S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, 2004.

A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4), 1999.

Q. Cao, Z. Guo, and Y. Ying. Generalization bounds for metric and similarity learning. *Machine Learning*, 102(1), 2016.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 11:2901–2934, 2010.

S. Davidson, S. Khanna, T. Milo, and S. Roy. Top-k and clustering with noisy comparisons. *ACM Trans. Database Syst.*, 39(4):35:1–35:39, 2014.

E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.

A. Demiriz, K. Bennett, and M.J. Embrechts. Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering (ANNIE-99)*, pages 809–814, 1999.

C. Gentile, M. Herbster, and S. Pasteris. Online similarity prediction of networked data from known and unknown graphs. In *Proceedings of the 23rd Conference on Learning Theory (26th COLT)*, 2013.

S. A. Goldman and M. K. Warmuth. Learning binary relations using weighted majority voting. In *Proceedings of the 6th Annual Conference on Computational Learning Theory*, pages 453–462, 1993.

S. A. Goldman, R. L. Rivest, and R. E. Schapire. Learning binary relations and total orders. *SIAM J. Comput.*, 22(5), 1993.

E. Hazan, S. Kale, and S. Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT'12)*, 2012.

D. Helmbold and M. Warmuth. On weak learning. *J. Comput. System Sci.*, 50:551–573, 1995.

Mark Herbster, Stephen Pasteris, and Massimiliano Pontil. Mistake bounds for binary matrix completion. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3954–3962. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6567-mistake-bounds-for-binary-matrix-completion.pdf.

S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, pages 1865–1890, 2012.

V. Koltchinskii, K. Lounici, and A. Tsybakov. Nuclear norm penalization and optimal rates for noisy matrix completion. In *arXiv:1011.6256v4*, 2016.

B. Kulis, S. Basu, I. Dhillon, and R. J. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine learning*, 74(1):1–22, 2009.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

A. Maurer. Learning similarity with operator-valued large-margin classifiers. *Journal of Machine Learning Research*, 9:1049–1082, 2008.

A. Mazumdar and B. Saha. Query complexity of clustering with side information. In *arXiv:1706.07719v1*, 2017a.

A. Mazumdar and B. Saha. Clustering with noisy queries. In *arXiv:1706.07510v1*, 2017b.

M. Meila. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98:873–895, 2007.

M. Meila. Local equivalences of distances between clusterings—a geometric perspective. *Machine Learning*, 86(3):369–389, 2012.

B. G. Mirkin. *Mathematical classification and clustering*. Dordrecht: Kluwer Academic, 1996.

A. Rajaraman and J. Ullman. *Mining of Massive Datasets*. 2010.

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on Machine learning*, ICML 2004. ACM, 2004.

K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projections. *Journal of Machine Learning Research*, 6: 995–1018, 2005.

M. K. Warmuth. Winnowing subspaces. In *Proceedings of the 24th International Conference on Machine Learning*, pages 999–1006, 2007.

E. P. Xing, A. Y. Ng, M. I. Jordan, and J. S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512, 2002.

Appendix A. Proofs

A.1. Proof of Theorem 1

Proof Recall that $D_k = \text{argmax}_{D \in \mathcal{D}} |D|$, and define for brevity $V_{\setminus k} := V \setminus D_k$. The rounds where Oppa can make mistakes is one of the cases contained in the following table.

Case	(v_t, w_t) in:	Value of ζ	After:	Round type
a	$D_k \times V_{\setminus k}$	$(\emptyset, \emptyset)$	$\zeta \leftarrow (A, A)$	τ_2
b	$V_{\setminus k} \times D_k$	$(\emptyset, \emptyset)$	$\zeta \leftarrow (A, A)$	τ_2
c	$D_k \times V_{\setminus k}$	$(\emptyset, A)$	$\zeta \leftarrow (A, B)$	τ_6
d	$V_{\setminus k} \times D_k$	$(A, \emptyset)$	$\zeta \leftarrow (B, A)$	τ_6
e	$D_k \times V_{\setminus k}$	$(A, \emptyset)$	$\zeta \leftarrow (B, A)$	τ_3
f	$V_{\setminus k} \times D_k$	$(\emptyset, A)$	$\zeta \leftarrow (A, B)$	τ_3
g	$D_k \times D_k$	$(\emptyset, B)$	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_4
h	$D_k \times D_k$	$(B, \emptyset)$	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_4
i	$D_k \times D_k$	(A, B)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_7
j	$D_k \times D_k$	(B, A)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_7
k	$D_k \times D_k$	(A, A)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_7
l	$D_k \times D_k$	(B, B)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_7
m	$V_{\setminus k} \times V_{\setminus k}$	$(\emptyset, B)$	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8
n	$V_{\setminus k} \times V_{\setminus k}$	$(B, \emptyset)$	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8
o	$V_{\setminus k} \times V_{\setminus k}$	(A, B)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8
p	$V_{\setminus k} \times V_{\setminus k}$	(B, A)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8
q	$V_{\setminus k} \times V_{\setminus k}$	(A, A)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8
r	$V_{\setminus k} \times V_{\setminus k}$	(B, B)	$\mathcal{C} \leftarrow \text{MERGE}_A(\mathcal{C}, C_v, C_w)$	τ_8

In what follows, we define suitable quantities. Notice that these quantities are dynamic, in that they change from round to round.

First, let $\omega(v)$ be a shorthand for $\omega(C_v)$. Let $\mathcal{X} := \{v \in V_{\setminus k} : \omega(v) \neq \emptyset\}$. Let τ_1 be the set of rounds on which an item v enters $\mathcal{X}$. Note that once an item has entered $\mathcal{X}$ it cannot leave, and hence $|\tau_1| \leq |V_{\setminus k}|$. Let τ_2 be the set of mistaken rounds of cases a and b, and τ_3 be the set of mistaken rounds of cases e and f. Observe that $\tau_2 \cap \tau_3 = \emptyset$ and that both τ_2 and τ_3 are subsets of τ_1. Hence

$$|\tau_2| + |\tau_3| \leq |\tau_1| \leq |V_{\setminus k}| \,.$$

Let $\delta = \{C \in \mathcal{C} : C \subseteq V_{\setminus k}\}$ and $\mathcal{Y} = \{C \in \delta : \omega(C) = B\}$. Note that the cardinality of $\delta \setminus \mathcal{Y}$ never increases over time, and is initially $|V_{\setminus k}|$, so it can only decrease $|V_{\setminus k}|$ times. Let τ_6 be the set of mistaken rounds of cases c and d. Since, for every round in τ_6 we have that the cardinality of $\delta \setminus \mathcal{Y}$ decreases, we must have

$$|\tau_6| \leq |V_{\setminus k}|.$$

Next, let $\epsilon = \{C \in \mathcal{C} \,:\, C \subseteq D_k\}$, and $\mathcal{W} = \{C \in \epsilon \,:\, \omega(C) = B\}$. Observe that $\mathcal{W}$ only increases on rounds in τ_3. Let τ_4 be the set of mistaken rounds with mistakes of cases g and

h. Note that for a round in τ_4 the cardinality of $\mathcal{W}$ decreases by one. Coupled with the fact that $\mathcal{W}$ only increases on rounds in τ_3, and only by one each time, we have that

$$|\tau_4| \leq |\tau_3| .$$

Let $\mathcal{U} = \{C \in \epsilon : \omega(C) \neq \emptyset\}$. Notice first that $\mathcal{U}$ is initially empty, and increases in cardinality on mistaken rounds in τ_2 (cases a and b) or τ_6 (cases c and d), where it only increases by one each time. Let τ_7 be the set of mistaken rounds of cases i to l, and observe that in such rounds $\mathcal{U}$ decreases in cardinality, hence directly from above we have

$$|\tau_7| \leq |\tau_2| + |\tau_6| .$$

Finally, let τ_8 be the set of mistaken rounds of cases m to r. Note that, for every round in τ_8, we have that $|\delta|$ deceases, so that since $|\delta|$ never increases, and is initially equal to $|V_{\setminus k}|$, we conclude that

$$|\tau_8| \leq |V_{\setminus k}| .$$

Now, because every mistaken round is either in τ_2, τ_3, τ_6, τ_4, τ_7 or τ_8, from the above displayed inequalities we have that the total number of mistakes

$$|\tau_2| + |\tau_3| + |\tau_6| + |\tau_4| + |\tau_7| + |\tau_8|$$

made by Oppa is bounded from above as follow:

$$\begin{aligned}
|\tau_2| + |\tau_3| + |\tau_6| + |\tau_4| + |\tau_7| + |\tau_8| &\leq |V_{\setminus k}| + |\tau_6| + |\tau_4| + |\tau_7| + |\tau_8| \\
&\leq 2|V_{\setminus k}| + |\tau_4| + |\tau_7| + |\tau_8| \\
&\leq 2|V_{\setminus k}| + |\tau_3| + |\tau_7| + |\tau_8| \\
&\leq 2|V_{\setminus k}| + |\tau_3| + |\tau_2| + |\tau_6| + |\tau_8| \\
&\leq 3|V_{\setminus k}| + |\tau_6| + |\tau_8| \\
&\leq 4|V_{\setminus k}| + |\tau_8| \\
&\leq 5|V_{\setminus k}| ,
\end{aligned}$$

thereby concluding the proof. ■

A.2. Proof of Theorem 2

Proof We describe an adversarial strategy forcing any algorithm to achieve the lower bound of Theorem 2. This strategy is described in Algorithm 5. Notice that the strategy only focuses on the first $n-1$ rounds of a potentially longer training sequence S. In fact, any learning algorithm is forced by this strategy to make at least $n-k-d_k$ mistakes over the first $n-1$ examples $\langle (v_1, w_1), y_{v_1,w_1} \rangle, \ldots, \langle (v_{n-1}, w_{n-1}), y_{v_{n-1},w_{n-1}} \rangle$ of S. These examples are such that $v_t \equiv w_{t-1}$ and $v_t \not\equiv v_{t-1}$ for all $t > 1$. In other words, the sequence of pairs (v_t, w_t) corresponding to the first $n-1$ examples of S is a line graph spanning the whole vertex set V.

The first vertex v_1 is assigned to a cluster D_i selected arbitrarily. At each round t, a *new* vertex w_t is assigned to a cluster selected according to the learner's prediction. This

Algorithm 5 Adversarial strategy for the online similarity prediction problem.

Input: Vertex set $V = \{1, \ldots, n\}$; sequence of cluster sizes $\langle d_1, d_2, \ldots, d_k \rangle$.
Initialization:

- For each $j = 1, \ldots, k$: Set $d'_j \leftarrow d_j$ and $D_j \leftarrow \emptyset$;
- Select arbitrarily $i \in [k]$; $D_i \leftarrow \{v_1\}$; $\widetilde{V} \leftarrow \{v_1\}$; $d'_i \leftarrow d'_i - 1$;

For $t = 1, \ldots, n-1$:

1. Select arbitrarily $w_t \in V \setminus \widetilde{V}$; $\widetilde{V} \leftarrow \widetilde{V} \cup \{w_t\}$;
2. Ask for label y_t;
3. If $(\hat{y}_t = 1 \vee d'_i = 0) \wedge (\exists j \neq i : d'_j \neq 0)$, then
   Set i to an arbitrarily selected cluster index $j \neq i$ such that $d'_j \neq 0$;
4. $D_i \leftarrow D_i \cup \{w_t\}$; $d'_i \leftarrow d'_i - 1$;

selection is accomplished in order to force a mistake whenever possible. The number of items that can be assigned to each cluster D_j for all $j \in [k]$, is initially equal to d_j, and decreases over time. At any time t, for all $j \in [k]$, let d'_j be the difference between d_j and the number of items assigned to D_j in the first $t-1$ rounds. In other words, for each $j \in [k]$, d'_j is the number of items that can be assigned to D_j during the remaining rounds $t, t+1, \ldots, n-1$.

The core of the adversarial strategy is simple. Given any round $t \in [n-1]$, let $\mathcal{D}_t$ be the set of all clusters D_j such that, at time t, $d'_j \neq 0$ and $j \neq \mu_{\mathcal{D}}(v_t)$. We then have the following cases:

- If the learner predicts 1 (similar) and $\mathcal{D}_t \not\equiv \emptyset$, then item w_t is assigned to a cluster D_j selected arbitrarily in $\mathcal{D}_t$. This assignment thus forces one mistake.
- If the learner predicts 0 (dissimilar) and $\mathcal{D}_t \not\equiv \emptyset$, then we have two sub-cases:
  - If $d'_{\mu_{\mathcal{D}}(v_t)} \neq 0$, then item w_t is assigned to cluster $D_{\mu_{\mathcal{D}}(v_t)}$, hence forcing one mistake.
  - If $d'_{\mu_{\mathcal{D}}(v_t)} = 0$, then w_t cannot be assigned to $D_{\mu_{\mathcal{D}}(v_t)}$, because $d_{\mu_{\mathcal{D}}(v_t)}$ items have already been assigned to this cluster. In this case, w_t is added to a cluster D_j selected arbitrarily in $\mathcal{D}_t$, such as when the learner predicts 1 and $\mathcal{D}_t \not\equiv \emptyset$. However, since now the learner predicts 0, no mistake can be forced.
- Finally, if $\mathcal{D}_t \equiv \emptyset$, according to the definition of $\mathcal{D}_t$, the only cluster D_j such that $d'_j \neq 0$ is $D_{\mu_{\mathcal{D}}(v_t)}$. Hence w_t must be assigned to $D_{\mu_{\mathcal{D}}(v_t)}$. Note that, in this case, no mistake can be forced if the learner predicts 1.

In order to quantify the effectiveness of this strategy, observe that, at each round $t \in [n-1]$ we have in the pseudocode of Algorithm 5 that $i = \mu_{\mathcal{D}}(v_t)$. Line 3 and 4 of the pseudocode ensure that one mistake is forced at any round such that both the following

conditions are satisfied: (a) $d'_i \neq 0$, and (b) $\exists j \neq i : d'_j \neq 0$. In fact, if both these conditions hold, then we have either (i) $\mu_{\mathcal{D}}(v_t) = \mu_{\mathcal{D}}(w_t)$ if $\hat{y}_t = 0$, which implies $y_t = 1$, or (ii) $\mu_{\mathcal{D}}(v_t) \neq \mu_{\mathcal{D}}(w_t)$ if $\hat{y}_t = 1$, which implies $y_t = 0$.

The number of rounds $t \in [n-1]$ such that condition (a) is violated cannot be larger than $k-1$. In fact, since $t \leq n-1$, we must always have at least one cluster index j such that $d'_j \neq 0$. On the other hand, the number of rounds $t \in [n-1]$ such that condition (b) is violated cannot be larger than $\max_j d_j = d_k$. Hence, the number of mistakes so forced is lower bounded by the difference between $n-1$ and $k-1+d_k$, i.e., $n-k-d_k$, as claimed. ■

A.3. Proof of Theorem 4

The following two lemmas are an immediate consequence of the triangle inequality for DIST.

Lemma 8 *Let $a, b \in [0,1]$ be such that $a+b \geq 3/2$, and sets U, W, X, Y, Z satisfy*

1. $\textsc{dist}(U,W) \leq 1-a$;
2. $\textsc{dist}(W,X) \leq 1-b$;
3. $\textsc{dist}(U,Y) \leq 1-a$;
4. $\textsc{dist}(Z,X) = 1$.

Then $\textsc{dist}(Y,Z) \geq 1-b$.

Proof We can write

$$\begin{aligned} 1 &= \textsc{dist}(Z,X) \\ &\leq \textsc{dist}(Z,Y) + \textsc{dist}(Y,U) + \textsc{dist}(U,W) + \textsc{dist}(W,X) \\ &\leq \textsc{dist}(Z,Y) + 1 - a + 1 - a + 1 - b \ , \end{aligned}$$

so that

$$\textsc{dist}(Z,Y) \geq 2a + b - 2 \geq 1 - b,$$

the last inequality using the assumption $a+b \geq 3/2$. This concludes the proof. ■

Lemma 9 *Let $a, b \in [0,1]$ be such that $2b \geq 1+a$, and sets X, Y, Z satisfy*

1. $\textsc{dist}(X,Y) \leq 1-b$:
2. $\textsc{dist}(Y,Z) \geq 1-a$.

Then $\textsc{dist}(X,Z) \geq 1-b$.

Proof We can write

$$1 - a \leq \textsc{dist}(Y,Z) \leq \textsc{dist}(Y,X) + \textsc{dist}(X,Z) \leq 1 - b + \textsc{dist}(X,Z),$$

so that

$$\textsc{dist}(X,Z) \geq b - a \geq 1 - b,$$

the last inequality deriving from $2b \geq 1 + a$. This concludes the proof. ■

With these two simple lemmas handy, we are now ready to analyze RGCA. The reader is compelled to refer to Algorithm 4 for notation. In what follows, $a \in [0,1]$ is RGCA's distance parameter, and $b \in [0,1]$ is a constant such that the two conditions on a and b required by Lemmas 8 and 9 simultaneously hold. It is easy to see that these conditions are equivalent to[5]

$$a \geq \frac{2}{3}, \qquad b \geq \frac{1+a}{2}\,. \tag{7}$$

The following definition will be useful.

Definition 10 *A b-anomaly in the similarity graph $(V, \mathcal{P})$ is a vertex $v \in V$ for which $\textsc{dist}(D_{\mu_{\mathcal{D}}(v)}, \Gamma(v)) \geq 1 - b$, for some constant $b \in [0,1]$ satisfying (7). We denote by Λ_b the set of all anomalies. A centered round of RGCA is any $t \leq \ell$ in which $N_t(\alpha_t) \not\subseteq \Lambda_b$. We denote by Ω_b the set of all centered rounds. A centered label is any class $i \in \{1, \ldots, k\}$ such that $D_i \not\subseteq \Lambda_b$. We denote by Δ_b the set of all centered labels.*

Lemma 11 *For any round $t \leq \ell$, there exists a class $j \in \{1, \ldots, k\}$ such that for every vertex $v \in N_t(\alpha_t) \setminus \Lambda_b$ we have $\mu_{\mathcal{D}}(v) = j$.*

Proof Suppose, for the sake of contradiction, that we have $v, w \in N_t(\alpha_t)$ with $v, w \notin \Lambda_b$ and $\mu_{\mathcal{D}}(v) \neq \mu_{\mathcal{D}}(w)$. Define $U := \Gamma(\alpha_t)$, $W := \Gamma(v)$, $X := D_{\mu_{\mathcal{D}}(v)}$, $Y := \Gamma(w)$, and $Z := D_{\mu_{\mathcal{D}}(v)}$. Since $v, w \in N_t(\alpha_t)$, by the way graph $(V, \mathcal{Q})$ is constructed, we have both $\textsc{dist}(U, W) \leq 1 - a$ and $\textsc{dist}(U, Y) \leq 1 - a$. Moreover, since $v \notin \Lambda_b$ we have $\textsc{dist}(W, X) < 1 - b$. Also, $\mu_{\mathcal{D}}(v) \neq \mu_{\mathcal{D}}(w)$ implies $\textsc{dist}(Z, X) = 1$. We are therefore in a position to apply Lemma 8 verbatim, from which we have $\textsc{dist}(Y, Z) \geq 1 - b$, i.e., $w \in \Lambda_b$. This is a contradiction, which implies the claimed result. ■

Lemma 11 allows us to make the following definition.

Definition 12 *Given a centered round $t \in \Omega_b$, we define $\gamma(t)$ to be the unique class j such that for every vertex $v \in N_t(\alpha_t) \setminus \Lambda_b$ we have $\mu_{\mathcal{D}}(v) = j$.*

Lemma 13 *For any round $t \leq \ell$ and vertices $v, w \in A_t$ with $v \notin \Lambda_b$, $w \notin N_t(v)$ and $\mu_{\mathcal{D}}(v) = \mu_{\mathcal{D}}(w)$ we have $w \in \Lambda_b$.*

Proof Define $X := D_{\mu_{\mathcal{D}}(v)}$, $Y := \Gamma(v)$ and $Z := \Gamma(w)$. Since $v \notin \Lambda_b$ we have $\textsc{dist}(X, Y) \leq 1 - b$. Moreover, $w \notin N_t(v)$ implies $\textsc{dist}(Y, Z) \geq 1 - a$. By Lemma 9 we immediately have $\textsc{dist}(X, Z) \geq 1 - b$. But since $X = D_{\mu_{\mathcal{D}}(v)} = D_{\mu_{\mathcal{D}}(w)}$, this equivalently establishes that $w \in \Lambda_b$. ■

Lemma 14 *For any centered round $t \in \Omega_b$, any vertex $v \in N_t(\alpha_t) \setminus \Lambda_b$, and any vertex $w \in N_t(\alpha_t) \setminus N_t(v)$, we have $w \in \Lambda_b$.*

5. For instance, we may set $a = 2/3$ and $b = 5/6$.

Proof If $\mu_{\mathcal{D}}(w) \neq \mu_{\mathcal{D}}(v)$ then by Lemma 11 we must have $w \in \Lambda_b$, so we are done. On the other hand, if $\mu_{\mathcal{D}}(w) = \mu_{\mathcal{D}}(v)$, we have $v \notin \Lambda_b$, $w \notin N_t(v)$ and $\mu_{\mathcal{D}}(v) = \mu_{\mathcal{D}}(w)$ which implies, by Lemma 13, that $w \in \Lambda_b$. ■

Lemma 15 *For any centered round $t \in \Omega_b$, we have $|(A_{t+1} \cap D_{\gamma(t)}) \setminus \Lambda_b| \leq |C_t \cap \Lambda_b|$.*

Proof Since $t \in \Omega_b$ there must exist a vertex $v \in N_t(\alpha_t)$ with $v \notin \Lambda_b$, so let us consider such a v. Note that by the way the algorithm works, we have $|N_t(\alpha_t)| \geq |N_t(v)|$, so that $|N_t(v) \setminus N_t(\alpha_t)| \leq |N_t(\alpha_t) \setminus N_t(v)|$. Next, by Lemma 14 we have $N_t(\alpha_t) \setminus N_t(v) \subseteq \Lambda_b$, hence $N_t(\alpha_t) \setminus N_t(v) \subseteq N_t(\alpha_t) \cap \Lambda_b$ and, consequently, $|N_t(\alpha_t) \setminus N_t(v)| \leq |N_t(\alpha_t) \cap \Lambda_b|$. Recalling that $C_t = N_t(\alpha_t)$, we have therefore obtained

$$|N_t(v) \setminus N_t(\alpha_t)| \leq |N_t(\alpha_t) \setminus N_t(v)| \leq |N_t(\alpha_t) \cap \Lambda_b| = |C_t \cap \Lambda_b| \,. \tag{8}$$

Now suppose we have some vertex $w \in (A_{t+1} \cap D_{\gamma(t)}) \setminus \Lambda_b$. For the sake of contradiction, let us assume that $w \notin N_t(v)$. Then $w \in A_{t+1}$ implies $w \in A_t$ which, combined with Lemma 13 together with the fact that $\mu_{\mathcal{D}}(v) = \gamma(t) = \mu_{\mathcal{D}}(w)$, implies that $w \in \Lambda_b$, which is a contradiction. Hence we must have $w \in N_t(v)$. Moreover, since $w \in A_{t+1}$ we must have $w \notin N_t(\alpha_t)$. We have hence shown that $w \in N_t(v) \setminus N_t(\alpha_t)$, implying that $|(A_{t+1} \cap D_{\gamma(t)}) \setminus \Lambda_b| \leq |N_t(v) \setminus N_t(\alpha_t)|$. Combining with (8) concludes the proof. ■

We now turn to considering centered labels.

Lemma 16 *For any centered label $i \in \Delta_b$ there exists some round $t \leq \ell$ such that $\gamma(t) = i$.*

Proof Since i is a centred label, pick $v \in D_i \setminus \Lambda_b$, Further, since $C_1, C_2, ..., C_\ell$ is a partition of $\mathcal{V}$, choose t such that $v \in C_t$. Now, since $v \in C_t \setminus \Lambda_b$ we have that $t \in \Omega_b$ and, by Lemma 11, that $\gamma(t) = \mu_{\mathcal{D}}(v) = i$. ■

Lemma 16 allows us to make the following definition.

Definition 17 *Given a centered label $i \in \Delta_b$, we define $\psi(i) := \min\{t : \gamma(t) = i\}$.*

Lemma 18 *For any centered label $i \in \Delta_b$, we have $D_i \setminus \Lambda_b \subseteq A_{\psi(i)}$.*

Proof Suppose, for contradiction, that there exists some $v \in D_i \setminus \Lambda_b$ with $v \notin A_{\psi(i)}$. Then, by definition of $A_{\psi(i)}$ there exists some round $t^o < \psi(i)$ with $v \in C_{t^o}$. As $v \notin \Lambda_b$ we have $t^o \in \Omega_b$ and, by Lemma 11, that $\mu_{\mathcal{D}}(v) = \gamma(t^o)$. Hence $\gamma(t^o) = \mu_{\mathcal{D}}(v) = i$ which, due to the condition $t^o < \psi(i)$, contradicts the fact that $\psi(i) := \min\{t : \gamma(t) = i\}$. ■

Lemma 19 *For any centred label $i \in \Delta_b$ we have $|D_i \setminus C_{\psi(i)}| \leq |D_i \cap \Lambda_b| + |C_{\psi(i)} \cap \Lambda_b|$.*

Proof Suppose we have some $v \in D_i \setminus C_{\psi(i)}$, and let us separate the two cases: (i) $v \notin \Lambda_b$ and, (ii) $v \in \Lambda_b$.

Case (i). Since $v \in D_i \setminus \Lambda_b$ we have, by Lemma 18, that $v \in A_{\psi(i)}$. Since $v \notin C_{\psi(i)}$ this implies that $v \in A_{\psi(i)+1}$. Notice that $\gamma(\psi(i)) = i$ so $D_i = D_{\gamma(\psi(i))}$ and hence $v \in$

$(A_{\psi(i)+1} \cap D_{\gamma(\psi(i))}) \setminus \Lambda_b$. By Lemma 15 the number of such vertices v is hence upper bounded by $|C_{\psi(i)} \cap \Lambda_b|$.

Case (ii). In this case, we simply have that $v \in D_i \cap \Lambda_b$, so the number of such vertices v is upper bounded by $|D_i \cap \Lambda_b|$.

Putting the two cases together gives us $|D_i \setminus C_{\psi(i)}| \leq |D_i \cap \Lambda_b| + |C_{\psi(i)} \cap \Lambda_b|$, as required. ■

Having established the main building blocks of the behavior of RGCA, we now turn to quantifying the resulting connection between ER and HA. To this effect, we start off by defining a natural map Υ associated with the clustering $\{C_1, \ldots, C_\ell\}$ generated by RGCA, along with a corresponding accuracy measure.

Definition 20 *The map* $\Upsilon : \{D_1, \ldots, D_k\} \to \{C_1, \ldots, C_\ell\}$ *is defined as follows:*

$$\Upsilon(D_i) = \begin{cases} C_{\psi(i)} & \text{if } i \in \Delta_b \\ \emptyset & \text{if } i \notin \Delta_b \end{cases}$$

Moreover, let $\mathcal{M}(\Upsilon) := \sum_{i=1}^{k} |D_i \setminus \Upsilon(D_i)|$.

We have the following lemma.

Lemma 21 $\mathcal{M}(\Upsilon) \leq 2|\Lambda_b|$.

Proof For $i \notin \Delta_b$ we have $D_i \subseteq \Lambda_b$ and $\Upsilon(D_i) = \emptyset$ so that

$$|D_i \setminus \Upsilon(D_i)| = |D_i| = |D_i \cap \Lambda_b| = |D_i \cap \Lambda_b| + |\emptyset| = |D_i \cap \Lambda_b| + |\Upsilon(D_i) \cap \Lambda_b|.$$

On the other hand, for $i \in \Delta_b$ we have $\Upsilon(D_i) = C_{\psi(i)}$ so that, by Lemma 19, we can write

$$|D_i \setminus \Upsilon(D_i)| \leq |D_i \cap \Lambda_b| + |\Upsilon(D_i) \cap \Lambda_b|\,.$$

Hence, in both cases, for all $i \in \{1, \ldots, k\}$ we have

$$|D_i \setminus \Upsilon(D_i)| \leq |D_i \cap \Lambda_b| + |\Upsilon(D_i) \cap \Lambda_b|\,,$$

implying

$$\mathcal{M}(\Upsilon) \leq \sum_{i=1}^{k} \Big(|D_i \cap \Lambda_b| + |\Upsilon(D_i) \cap \Lambda_b| \Big)\,. \tag{9}$$

Now, both $\{D_1, \ldots, D_k\}$ and $\{\Upsilon(D_1), \ldots, \Upsilon(D_k)\}$ are a partition of V, implying

$$|\Lambda_b| = \sum_{i=1}^{k} |D_i \cap \Lambda_b| = \sum_{i=1}^{k} |\Upsilon(D_i) \cap \Lambda_b|\,.$$

Plugging back into (9) yields the claimed result. ■

Next, observe that, by its very definition, $\mathrm{HA}(\mathcal{P}, \mathcal{D})$ can be rewritten as

$$\mathrm{HA}(\mathcal{P}, \mathcal{D}) = \sum_{v \in V} |(D_{\mu_{\mathcal{D}}(v)} \setminus \Gamma(v)) \cup (\Gamma(v) \setminus D_{\mu_{\mathcal{D}}(v)})|\,. \tag{10}$$

Lemma 22 *We have* $\mathrm{HA}(\mathcal{P},\mathcal{D}) \geq (1-b)\sum_{i=1}^{k} d_i |D_i \cap \Lambda_b|$.

Proof Fix class $i \in \{1,\dots,k\}$ and vertex $v \in D_i \cap \Lambda_b$. Then $v \in \Lambda_b$ implies $\textsc{dist}(D_{\mu_{\mathcal{D}}(v)}, \Gamma(v)) \geq 1-b$, which in turn yields

$$|(D_{\mu_{\mathcal{D}}(v)} \setminus \Gamma(v)) \cup (\Gamma(v) \setminus D_{\mu_{\mathcal{D}}(v)})| \geq (1-b)d_i \,,$$

thereby concluding that for all fixed i

$$\sum_{v \in D_i \cap \Lambda_b} |(D_{\mu_{\mathcal{D}}(v)} \setminus \Gamma(v)) \cup (\Gamma(v) \setminus D_{\mu_{\mathcal{D}}(v)})| \geq (1-b)\,|D_i \cap \Lambda_b|\, d_i \,.$$

Since $\Lambda_b = \bigcup_{i=1}^{k}(D_i \cap \Lambda_b)$, being the sets $D_i \cap \Lambda_b$, $i = 1,\dots,k$, pairwise disjoint, we can write

$$\begin{aligned}\sum_{v \in \Lambda_b} |(D_{\mu_{\mathcal{D}}(v)} \setminus \Gamma(v)) \cup (\Gamma(v) \setminus D_{\mu_{\mathcal{D}}(v)})| &= \sum_{i=1}^{k} \sum_{v \in D_i \cap \Lambda_b} |(D_{\mu_{\mathcal{D}}(v)} \setminus \Gamma(v)) \cup (\Gamma(v) \setminus D_{\mu_{\mathcal{D}}(v)})| \\ &\geq (1-b)\sum_{i=1}^{k} |D_i \cap \Lambda_b|\, d_i \,.\end{aligned}$$

Thus, from (10), and the fact that $\Lambda_b \subseteq \mathcal{V}$ the result immediately follows. ■

Lemma 23 *The number* $|\Lambda_b|$ *of b-anomalies can be upper bounded as*

$$|\Lambda_b| \leq \min_{j=1,\dots,k} \left(\frac{1}{d_j(1-b)}\,\mathrm{HA}(\mathcal{P},\mathcal{D}) + \sum_{i=1}^{j-1} d_i \right) .$$

Proof For any $j = 1,\dots,k$ we can write

$$|\Lambda_b| = \sum_{i=1}^{k} |D_i \cap \Lambda_b| = \sum_{i=1}^{j-1} |D_i \cap \Lambda_b| + \sum_{i=j}^{k} |D_i \cap \Lambda_b| \leq \sum_{i=1}^{j-1} d_i + \sum_{i=j}^{k} |D_i \cap \Lambda_b|$$

so all that is left to prove is that the last sum in the right-hand side is at most $\frac{1}{d_j(1-b)}\,\mathrm{HA}(\mathcal{P},\mathcal{D})$.

Since, for all classes i such that $i \geq j$, we have $d_i \geq d_j$, we can write

$$\sum_{i=j}^{k} |D_i \cap \Lambda_b| \leq \sum_{i=j}^{k} \frac{d_i}{d_j} |D_i \cap \Lambda_b| \leq \frac{1}{d_j} \sum_{i=1}^{k} d_i\, |D_i \cap \Lambda_b| \leq \frac{1}{d_j(1-b)}\,\mathrm{HA}(\mathcal{P},\mathcal{D}) \,,$$

where the last inequality derives from Lemma 22. This concludes the proof. ■

We are now ready to combine to above lemmas into the proof of Theorem 4.

Proof (Theorem 4) Direct from Lemmas 21 and 23 we have

$$\mathcal{M}(\Upsilon) \leq \min_{j=1,\dots,k} \left(\frac{2}{d_j\,(1-b)}\,\mathrm{HA}(\mathcal{P},\mathcal{D}) + \sum_{i=1}^{j-1} d_i \right) .$$

We then optimize for b by selecting $b = \frac{1+a}{2}$, and then for a by setting $a = 2/3$, so as to fulfil conditions (7). The result follows by the fact that $\mathrm{ER}(\mathcal{C},\mathcal{D}) \leq \mathcal{M}(\Upsilon)$, for $\mathrm{ER}(\mathcal{C},\mathcal{D})$ is a minimum over all possible cluster maps $\mathcal{D} \to \mathcal{C}$, while Υ is just the one in Definition 20. ■

A.4. Proof of Theorem 5

Proof For ease of proof, we assume that d_j is even for all j (adapting the proof to the general case is trivial). We consider two cases:

1. $\sigma \geq \frac{1}{2}\sum_{j=1}^{k} d_j^2$;
2. $\sigma < \frac{1}{2}\sum_{j=1}^{k} d_j^2$.

For the first case we choose, for every $j = 1, \ldots, k$, sets P_j^+ and P_j^- such that $|P_j^+| = |P_j^-| = d_j/2$ and $P_j^+ \cup P_j^- = D_j$. We then construct the similarity graph $(\mathcal{V}, E_{\mathcal{P}})$, where clustering $\mathcal{P}$ is made up of the $2k$ clusters $\{P_j^+ : j = 1, \ldots, k\} \cup \{P_j^- : j = 1, \ldots, k\}$. Since the algorithm is consistent, we must have $\mathcal{C} = \mathcal{P}$. Now, let f be an injection from $\mathcal{D}$ to $\mathcal{C}$, and consider any $j = 1, \ldots, k$. If $f(D_j) \in \{P_j^+, P_j^-\}$ then we have $|D_j \setminus f(D_j)| = d_j/2$, and otherwise $|D_j \setminus f(D_j)| = d_j$, so that

$$\sum_{j=1}^{k} |D_j \setminus f(D_j)| \geq \frac{1}{2}\sum_{j=1}^{k} d_j = n/2\,.$$

Since f is arbitrary, this shows that $\mathrm{ER}(\mathcal{C}, \mathcal{D}) \geq \frac{n}{2}$. Moreover, we observe that the only incorrect similarity/dissimilarity predictions of $\mathcal{P}$ with respect to $\mathcal{D}$ are those between P_j^+ and P_j^-, for every j, which gives us $2|P_j^+| \cdot |P_j^-| = d_j^2/2$ incorrect predictions for every j. This implies that $\mathrm{HA}(\mathcal{P}, \mathcal{D}) = \sum_{j=1}^{k} d_j^2/2$, which is no greater than σ, thereby completing the proof for the first case.

We now turn to the second case. Let $j^o \in \{1, \ldots, k\}$ be such that

$$\frac{1}{2}\sum_{i=1}^{j^o-1} d_i^2 \leq \sigma < \frac{1}{2}\sum_{i=1}^{j^o} d_i^2,$$

and $\omega := \sigma - \frac{1}{2}\sum_{i=1}^{j^o-1} d_i^2$. Notice that $\omega \leq d_{j^o}^2/2$. We choose, for every $j < j^o$, sets P_j^+ and P_j^- such that $|P_j^+| = |P_j^-| = d_j/2$ and $P_j^+ \cup P_j^- = D_j$. Let $c = \lfloor \omega/2d_{j^o} \rfloor$, and note that $c \leq d_{j^o}/4 < d_{j^o}/2$. We can hence define subsets $X, Y \subseteq D_{j^o}$ such that $|X| = c$, $X \cup Y = D_{j^o}$ and $X \cap Y = \emptyset$.

We construct the similarity graph $(\mathcal{V}, E_{\mathcal{P}})$, where clustering $\mathcal{P}$ is made up of the $k + j^o$ clusters

$$\{P_j^+ : j = 1, \ldots, j^o - 1\} \cup \{P_j^- : j = 1, \ldots, j^o - 1\} \cup \{X, Y\} \cup \{D_j : j > j^o\}\,.$$

Again, since the algorithm is consistent, we must have $\mathcal{C} = \mathcal{P}$. As before, let f be an arbitrary injection from $\mathcal{D}$ to $\mathcal{C}$, and consider any $j < j^o$. Then if $f(D_j) \in \{P_j^+, P_j^-\}$ we have $|D_j \setminus f(D_j)| = d_j/2$, otherwise $|D_j \setminus f(D_j)| = d_j$, so that $|D_j \setminus f(D_j)| \geq d_j/2$ holds for any $j < j^o$. Further, if $f(D_{j^o}) = X$ then $|D_{j^o} \setminus f(D_{j^o})| = d_{j^o} - c$, if $f(D_{j^o}) = Y$ then $|D_{j^o} \setminus f(D_{j^o})| = c$, and otherwise $|D_{j^o} \setminus f(D_{j^o})| = d_{j^o}$. In any case, since $c < d_{j^o}/2$, we have

$|D_{j^o} \setminus f(D_{j^o})| \geq c$. This allows us to conclude that

$$\begin{aligned}
\mathrm{ER}(\mathcal{C}, \mathcal{D}) &= \sum_{j=1}^{k} |D_j \setminus f(D_j)| \\
&\geq c + \frac{1}{2} \sum_{j=1}^{j^o-1} d_j \\
&= \lfloor \omega / 2d_{j^o} \rfloor + \frac{1}{2} \sum_{j=1}^{j^o-1} d_j \\
&\geq \frac{\omega}{2d_{j^o}} - 1 + \frac{1}{2} \sum_{j=1}^{j^o-1} d_j \\
&= \frac{\sigma}{2d_{j^o}} - 1 - \frac{1}{4d_j^o} \sum_{j=1}^{j^o-1} d_j^2 + \frac{1}{2} \sum_{j=1}^{j^o-1} d_j \\
&= \frac{\sigma}{2d_{j^o}} - 1 + \frac{1}{2} \sum_{j=1}^{j^o-1} d_j \left(1 - \frac{d_j}{2d_{j^o}}\right) \\
&\geq \frac{\sigma}{2d_{j^o}} - 1 + \frac{1}{4} \sum_{j=1}^{j^o-1} d_j \,.
\end{aligned}$$

Finally, notice that the only incorrect similarity/dissimilarity predictions of $\mathcal{P}$ with respect to $\mathcal{D}$ are those between P_j^+ and P_j^-, for every $j < j^o$, and those between X and Y, which gives us $2|P_j^+| \cdot |P_j^-| = d_j^2/2$ incorrect predictions for every $j < j^o$, and an additional $2|X| \cdot |Y| = 2c(d_{j^o} - c) \leq 2cd_{j^o} \leq \omega$ incorrect predictions between X and Y. This implies that

$$\mathrm{HA}(\mathcal{P}, \mathcal{D}) \leq \omega + \sum_{j=1}^{j^o-1} d_j^2/8$$

which is in turn bounded from above by σ. This completes the proof for the second case. ■

A.5. Proof of Theorem 6

The following simple lemma is of preliminary importance.

Lemma 24 *Let $H = (V, E)$ be an Erdos-Renyi $G(n, p)$ graph. For each subgraph $H'(V', E') \subseteq H$ with $n' = |V'|$ nodes, when $p = \frac{\lambda \log n'}{n'}$ the following* separation property *holds: As n' approaches infinity, the expected number z of isolated vertices in G' equals $(n')^{1-\lambda}$. Furthermore, in the special case when $n' = \frac{1}{p}$, we always have $z \geq \frac{1}{pe}$.*

Proof In order to prove this property, it suffices to observe that, given any node in V', the probability that it is isolated in G' is equal to $(1-p)^{n'-1}$, which in turn is equal to $e^{-\lambda \log n'} = (n')^{-\lambda}$ as n' approaches infinity. Hence we have $z = (n')^{1-\lambda}$. By a similar

argument, it is immediate to verify that in the case when $n' = \frac{1}{p}$ we have $z = \frac{1}{p}(1-p)^{\frac{1}{p}-1}$ which is never smaller than $\frac{1}{ep}$. ■

Proof (Theorem 6) Let $G' = (V, E')$ denote the undirected graph whose edge set E' is made up of all pairs of vertices drawn in S. Since S is drawn uniformly at random, G' turns out to be an Erdos-Renyi graph $G(n, p)$, with $p = m/n^2$.

Setting $\lambda = 2$ in Lemma 24, we have that for all clusters $C \in \mathcal{C}$ such that $\frac{2\log|C|}{|C|} \leq p$, cluster C can be *completely* detected by Algorithm 1 (line 3 therein) with probability at least $\frac{1}{|C|}$. Hence, the expected number of misclassification errors made when detecting such clusters is upper bounded by 1 per cluster. In order to satisfy the assumption $\frac{2\log|C|}{|C|} \leq p$, the size of these clusters must be equal to a value $\tau = \Omega(\rho \log \rho)$, where we set $\rho = \frac{1}{p}$.

Finally, we can conclude the proof observing that the total number of misclassification errors is bounded in expectation by the sum of the following two quantities: (i) the number of clusters larger than τ, which in turn is bounded by k, and (ii) the total number of nodes belonging to the clusters smaller or equal to τ, which in turn is bounded by $k\tau$:

$$\mathbb{E}\left[\mathrm{ER}(\mathcal{C}, \mathcal{D})\right] = \mathcal{O}\left(k(1+\tau)\right) = \mathcal{O}\left(k\rho \log \rho\right) , \tag{11}$$

thereby concluding the proof. ■

A.6. Proof of Theorem 7

Proof As in the proof of Theorem 6, we denote by $G' = (V, E')$ the undirected graph whose edge set E' is made up of all pairs of vertices drawn in the training set S. Since S is drawn uniformly at random, G' turns out to be an Erdos-Renyi graph.

The basic idea of this proof is to construct a collection $\mathcal{H}$ of z disjoint subsets of V, call them $H_1, H_2 \ldots, H_z$, and, for all $j \in \{1, \ldots, z\}$, to *randomly* label all nodes of each subset H_j using only a pair of classes of $\{1, \ldots, k\}$. These z pairs of classes must be distinct and disjoint. The random labeling is accomplished in such a way that no algorithm can exploit the training set to guess how each H_j is labeled. More specifically, $\mathcal{H}$ is created so as to satisfy the following two properties:

Property (i) For all $j = 1, \ldots, z$, no pair of nodes in H_j are connected by an edge in the training graph representation G', i.e. for each pair of nodes $u, v \in H_j$, we have $(u, v) \notin S$.

Property (ii) For all $j = 1, \ldots, z$, we have that the expected size of H_j (over the random draw of the training set S) is larger than $\frac{n^2}{2m\,e}$, if $m > \frac{n}{2}$, where e is the base of natural logarithms, while it is $\Theta(n)$ if $m \leq \frac{n}{2}$.

Figure 1 provides a pictorial explanation of the randomized labeling strategy we are going to describe.

We now describe in detail the randomized labeling strategy (a randomized clustering $\mathcal{D}$ representing a clustering with k clusters), and derive a lower bound for $\mathbb{E}_{S,\mathcal{D}}[\mathrm{ER}(\mathcal{C}, \mathcal{D})]$ when $\mathcal{H}$ satisfies both the above properties.

Let $z \leq \lfloor \frac{k-1}{2} \rfloor$. Once we constructed such a collection $\mathcal{H}$ of clusters, we associate a distinct pair of classes in $\{1, \ldots, k\}$ with each H_j in such a way that all these class pairs are distinct and disjoint. This allows us to always leave one class out (say, class k) for labeling

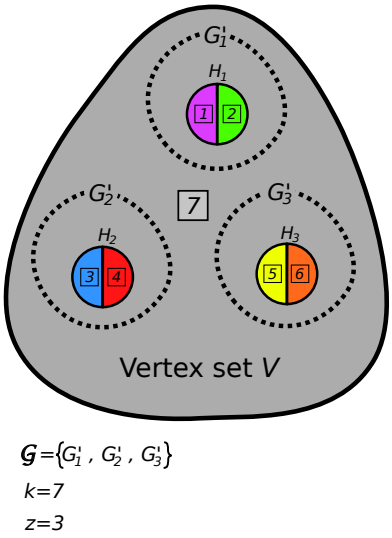

Figure 1: Illustration of the randomized labeling that achieves the lower bound in Theorem 7. The grey area includes all the nodes of V. In this example, we set $z = 3$ and $k = 7$. In this case we thus have $\lfloor \frac{k-1}{2} \rfloor = 3$. $\mathcal{G}$ is the collection of the 3 vertex-disjoint subgraphs G'_1, G'_2 and G'_3. The node set size of each of these subgraphs is equal to $\left\lfloor \frac{n^2}{2m} \right\rfloor$. The subsets of isolated vertices in these 3 subgraphs are H_1, H_2, and H_3, which are depicted in this figure by the bicoloured circles. Each color represents a class. For each j, the expected size of H_j must be linear in the size of the node set of G'_j. For $j = 1, 2, 3$, set H_j is labeled by selecting uniformly at random a class between the two classes (or colors) $2j - 1$ and $2j$. All the remaining nodes in the grey area of this picture are given the same class 7. Hence, for each pair of nodes u and v both belonging to H_j for some j, we must have $(u, v) \notin S$. On the contrary, for each pair u and v with $u \in H_j$, for some j, and $v \notin H_j$, we must have $y_{u,v} = 0$. The information of the training set cannot be used to predict how the nodes in H_1, H_2, and H_3, are labeled. In fact, *any* algorithm will make $\frac{1}{2}$ mistakes in expectation over the randomized labeling on each node contained in these subsets.

all remaining vertices in V. In particular, we associate with H_j the class pair $(2j - 1, 2j)$, and then adopt the following randomized strategy:

For all $j = 1, \ldots, z$, set H_j is split uniformly at random into two subsets H'_j and H''_j, and we label H'_j by class $2j - 1$ and H''_i by class $2j$. All remaining nodes in $V \setminus \cup_{j=1}^{z} H_j$ are labeled with class k.

This randomized labeling strategy ensures that, in order to guess the true clustering $\mathcal{D}$, no learning algorithm can exploit the information provided by S, since for all node pairs (v, w) with $v \in H_j$, for some $j \in \{1, \ldots, z\}$, one of two cases hold:

Case (a): $w \in H_j$, which implies that $(v, w) \notin S$, because of Property (i). We have therefore no training set information related to the similarity of nodes laying in the same set H_j.

Case (b): $w \notin H_j$. In this case, whenever $(v,w) \in S$, we *always* have $y_{v,w} = 0$, and this information cannot be exploited to guess the randomized labeling of H_j.

In short, no training information can be exploited to guess how each set H_j is split into the two subsets H'_j and H''_j. This entails that any clustering algorithm will incur an expected number of misclassification errors proportional to

$$\sum_{j=1}^{z} |H_j| = \Omega\left(\min\left\{\frac{n^2}{m}\, z, n\right\}\right),$$

the latter equality deriving from Property (ii).

We now turn to describing the detailed construction of $\mathcal{H}$. We will explain how to select the z subsets satisfying Property (i), and show that their size is bounded from below as required by Property (ii). This will lead to the claimed lower bound.

Definition of z. Let

$$z = \min\left\{f(n,m), \left\lfloor \frac{k-1}{2} \right\rfloor\right\}, \qquad \text{where} \quad f(n,m) = \max\left\{\left\lfloor \frac{n}{\lfloor\, n^2/2\,m\rfloor} \right\rfloor, 1\right\}.$$

Satisfaction of Property (i).
Let $\mathcal{H}'$ be a collection of disjoint subsets of V created as follows. $\mathcal{H}'$ is generated by selecting uniformly at random z disjoint subsets of V such that each node subset contains $\left\lfloor \frac{n^2}{2m} \right\rfloor$ nodes. The collection of subsets $\mathcal{H} = \{H_1, \ldots, H_z\}$ is constructed as described next. Let $\mathcal{G} \equiv \{G'_1, G'_2, \ldots, G'_z\}$, where G'_j is the subgraph of G' induced by the nodes in the j-th set of $\mathcal{H}'$. We create z-many disjoint subsets $H_1, H_2, \ldots, H_z$ by selecting all vertices that are *isolated* in each graph of $\mathcal{G}$, and set $\mathcal{H} \equiv \{H_1, H_2, \ldots, H_z\}$. Property (i) is therefore satisfied.

Satisfaction of property (ii).
By definition of $\mathcal{H}'$, each graph of the collection $\mathcal{G}$ has $\left\lfloor \frac{n^2}{2m} \right\rfloor$ nodes in expectation. Using the second part of Lemma 24, we conclude that the expected size of each set in $\mathcal{H}$ is not smaller than $\frac{\lfloor n^2/2m \rfloor}{e}$.

Hence the collection of sets $\mathcal{S}$ so generated fulfils at the same time both Properties (i) and (ii).

In order to conclude the proof, we compute our lower bound based on the definition of z and Property (ii). As anticipated, because of the randomized labeling strategy, the expected number of misclassification errors made by any algorithm is proportional to $\sum_{j=1}^{z} |H_j| = \Omega\left(\min\left\{\frac{n^2}{m} z, n\right\}\right)$. Plugging in the values of z yields

$$\sum_{j=1}^{z} |H_j| = \Omega\left(\min\left\{\frac{\min\left\{\max\left\{\left\lfloor \frac{n}{\lfloor\, n^2/2\,m\rfloor}\right\rfloor, 1\right\}, \left\lfloor \frac{k-1}{2}\right\rfloor\right\}}{m/n^2}, n\right\}\right) = \Omega\left(\min\left\{\frac{n^2}{m}\, k, n\right\}\right),$$

and the proof is concluded.

■