Active Learning Convex Halfspaces on Graphs

Anonymous Author(s)
Affiliation
Address
email

Abstract

We systematically study the query complexity of learning geodesically convex halfspaces on graphs. Geodesic convexity is a natural generalisation of Euclidean convexity and allows the definition of convex sets and halfspaces on graphs. We prove upper bounds on the query complexity linear in the treewidth and the minimum hull set size but only logarithmic in the diameter. We show tight lower bounds corresponding to well-established separation axioms and identify the Radon number as a central parameter bounding the query complexity and the VC dimension. While previous bounds typically depend on the cut size of the labelling, all parameters in our bounds can be computed from the unlabelled graph. We empirically compare our proposed approach with other active learning algorithms and provide evidence that ground-truth communities in real-world graphs are often convex.

1 Introduction

We present a systematic characterisation of active learning geodesically convex halfspaces on graphs. While unlabelled graphs such as social networks are readily accessible, obtaining many labels is a tedious and labour intensive task. Active learning reduces the labelling effort by iteratively and carefully selecting the vertices to be labelled. Abstract convexity spaces in general and geodesic convexity on graphs are natural generalisations of Euclidean convexity. However, while learning convex classes and halfspaces has been intensively researched in the machine learning community for over half a century, geodesic convexity on graphs has been largely overlooked so far. To test the hypothesis that geodesic convexity is indeed a natural concept on graphs, we performed a preliminary experimental study that confirms that many communities in social networks are indeed convex. For a concise presentation of our theoretical results in this paper, we concentrate on binary classification, that is, active learning geodesically convex halfspaces.

After introducing the terminology and tools of graph and abstract convexity theory (in Section 2), we show that general, worst-case upper and lower bounds holding for the set of all graphs are inherently loose. To derive tighter lower bounds, we structure the space of graphs along the so called separation axioms of abstract convexity theory. We show that an upper bound—related to treewidth, diameter, and hull set of the graph—is tight across all sets of graphs induced by the separation axioms and derive tighter lower bounds for each of these sets. This allows us to significantly close the gap between upper and lower bounds and present the first precise characterisation of active learning geodesically convex halfspaces on graphs (in Section 3). Note that a similar characterisation does not carry over to Euclidean convexity: While the separation axioms partition the space of all graphs into non-empty sets, all separation axioms hold in all linear spaces such as the Euclidean ones. We discuss related areas like active learning convex sets in Euclidean spaces, passive learning geodesically convex halfspaces, active learning bounds in terms of the cut size between the classes, and other related learning theoretic concepts (in Section 4). In particular, we also provide a novel bound on the VC-dimension of geodesically convex halfspaces. After presenting an empirical validation of our assumptions and approach (in Section 5), we conclude with a discussion of promising furture work.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Convexity spaces

In this section, we introduce important and necessary concepts of convexity theory. For a more thorough introduction on convexity theory we refer the reader to van de Vel [1993] and Pelayo [2013].

For a set X and a family $\mathcal{C} \subseteq 2^X$ of subsets, the tuple $(X, \mathcal{C})$ is a *convexity space* if *(i)* $\emptyset, X \in \mathcal{C}$, *(ii)* $\mathcal{C}$ is closed under intersection, and *(iii)* $\mathcal{C}$ is closed under nested union. For finite set systems, property *(iii)* always holds. Any set in $\mathcal{C}$ is called *convex*. If a set C and its complement $X \setminus C$ are convex, both are called *halfspaces*. Two disjoint sets A, B are *separated* by a halfspace C if $A \subseteq C$ and $B \subseteq X \setminus C$. A mapping $\sigma : 2^X \to 2^X$ is a *convex hull* (or *closure*) *operator* if for all $A, B \subseteq X$ with $A \subseteq B$ *(i)* $\sigma(\emptyset) = \emptyset$, *(ii)* $\sigma(A) \subseteq \sigma(B)$, *(iii)* $A \subseteq \sigma(A)$, and *(iv)* $\sigma(\sigma(A)) = \sigma(A)$. Any convexity space $(X, \mathcal{C})$ induces a convex hull operator by $\sigma(A) = \bigcap\{C \in \mathcal{C} \mid A \subseteq C\}$. A set $A \subseteq X$ is convex, that is $A \in \mathcal{C}$, if and only if is equal to its convex hull, $A = \sigma(A)$.

Convexity spaces can be characterised by their ability to separate sets via halfspaces, following a related notion in topological spaces.

Definition 1 (Separation Axioms [van de Vel, 1993])**.** *A convexity space* $(X, \mathcal{C})$ *is:*

> S_1 *if and only if each singleton* $x \in X$ *is convex.*
>
> S_2 *if and only if each pair of distinct elements* x*,* y *is halfspace separable.*
>
> S_3 *if and only if each convex set* C *and elements* $x \in X \setminus C$ *are halfspace separable.*
>
> S_4 *if and only if any two disjoint convex sets are halfspace separable.*

If S_1 holds, the remaining axioms are increasingly stronger, that is, $S_2 \Leftarrow S_3 \Leftarrow S_4$. For real vector spaces such as the Euclidean space, all four separation axioms hold [Kakutani, 1937].

A set $H \subseteq X$ is a *hull set* if its convex hull is the whole space, $\sigma(H) = X$. For $A, B \subseteq X$, the set $A/B = \{x \in X \mid A \cap \sigma(B \cup \{x\}) \neq \emptyset\}$ is the *extension of* A (*away from* B). For $a, b \in X$, the extension $\{a\}/\{b\}$ is also called a *ray* a/b. Two disjoint sets A_1, A_2 form a partition of $A \subseteq X$ if $A_1 \cup A_2 = A$. The partition A_1, A_2 of A is a *Radon partition* if $\sigma(A_1) \cap \sigma(A_2) \neq \emptyset$. The *Radon number* is the minimum number r such that any subset of X of size r or larger has a Radon partition.

A particular type of convexity is *interval convexity*. Apart from the convex hull $\sigma(\cdot)$ they have an *interval mapping* $I : X \times X \to 2^X$ such that for all $x, y \in X$, *(i)* $x, y \in I(x, y)$ and *(ii)* $I(x, y) = I(y, x)$. $I(x, y)$ is the *interval* between x and y. We denote $I(A) = \bigcup_{a,b \in A} I(a, b)$. A set A in an interval convexity space is convex if and only if $A = I(A)$. Denoting $I^1(\cdot) = I(\cdot)$ and $I^{k+1}(\cdot) = I(I^k(\cdot))$, it holds that $\sigma(A) = \bigcup_{k=1}^{\infty} I^k(A)$. A well-known instance of interval convexity spaces are metric spaces (X, d). There, the interval contains all the points for which the triangle inequality holds with equality: $I_d(x, y) = \{z \in X \mid d(x, y) = d(x, z) + d(z, y)\}$. In Euclidean space, this corresponds to all points on a line segment and leads to the classical notion of convex sets.

We are interested in the geodesic convexities induced by graphs. The geodesic convexity (or shortest path convexity) of a connected graph $G = (V, E)$ is given by the interval mapping I_d, where d is the *shortest path distance* in the graph. For unweighted graphs it is the number of edges on any shortest path and for graphs with edge weights, $w : E \to \mathbb{R}_{>0}$, it is the sum of the edge weights on any shortest path. Using this definition, a subset of vertices $C \subseteq V$ is (geodesically) convex if and only if every shortest path with endpoints in C stays in C, corresponding again to the Euclidean case.

We denote the Radon number of the induced geodesic convexity space of a graph as $r(G)$ and the size of the minimum hull set as $h(G)$. The notion of rays simplifies for graphs. For an edge $\{a, b\}$ in an unweighted graph $a/b = \{v \in V \mid d(a, v) < d(b, v)\}$ and for a weighted graph $a/b = \{v \in V \mid d(b, v) = d(b, a) + d(a, v)\}$. A vertex v is *extreme* if $V \setminus \{v\}$ is convex, that is, $\{v\}$ is a halfspace. We will denote the set of extreme vertices of a graph G as $\mathrm{Ext}(G)$. In the unweighted case, v is extreme if and only if its neighbours form a clique. The *diameter* $d(G)$ of a weighted or unweighted graph G is the maximum number of edges in any shortest path in G.

We say a graph is S_i, for $i = 1, \ldots, 4$, if for the induced geodesic convexity space the respective separation axiom holds. Moreover, there are graphs that are S_i but not S_{i+1} for $i = 1, 2, 3$ [Bandelt, 1989], see supplementary for some examples. In this regard, graph convexity spaces are more general than the Euclidean one, for which always all four separation axioms hold.

3 Active learning halfspaces on graphs

Having introduced the necessary concepts from convexity theory, we now present the main theoretical results of the paper. In active learning on graphs, an undirected graph $G = (V, E)$ with unknown vertex labels $\lambda : V \to \{0, 1\}$ is given and the goal is to accurately predict all labels using as few as possible iterative vertex queries $\lambda(v)$. Edges between vertices with different labels are called *cut edges*. In this paper, we consider active learning on graphs on which the vertices of the same class form a geodesically convex set. Our results in this section are upper and lower bounds on the number of queries required to deduce all labels of the graph for binary classification on connected undirected graphs. In this case, the classes form halfspaces, we call the labelling *halfspace separable*, and denote the least number of queries required to identify any halfspace separable labelling in the worst-case, the *query complexity*, by $\mathrm{qc}(G)$. We derive general upper (Theorem 6) and lower (Theorem 8) bounds on $\mathrm{qc}(G)$ that hold for any labelling. To reduce the gap (Proposition 9) between the upper and lower bounds, we inspect the graph families induced by the separation axioms and derive increasingly tighter lower bounds (Theorem 10). All proofs, multi-class settings scenarios, and adaptations for directed and disconnected graphs can be found in the supplementary material.

3.1 Upper bounds on the query complexity

To derive a first simple upper bound we consider that one immediate consequence of the halfspace assumption is that any shortest path can have at most one cut edge. Therefore, either the endpoints of a shortest path P have the same label or we can find a cut edge by binary search with at most $\lceil \log d(G) \rceil$ queries, as $|V(P)| - 1 \leq d(G)$, where log is the base 2 logarithm. We can generalise this approach to the whole graph using *shortest path covers*. A shortest path cover $\mathcal{S}$ is a set of shortest paths whose vertices jointly cover the graph: $\bigcup_{P \in \mathcal{S}} V(P) = V(G)$. Binary searching each of the paths in the cover gives our first upper bound.

Proposition 2. *For any weighted graph G with minimum shortest path cover $\mathcal{S}^*$ the query complexity can be bounded as* $\mathrm{qc}(G) \leq |\mathcal{S}^*|(2 + \log d(G))$.

By considering a particular fully connected weighted graph with an arbitrary number of paths with weights are such that all cut edges are independent of each other, we can show that this bound is tight:

Proposition 3. *For any $d, s \in \mathbb{N}$, there exists a weighted graph G with diameter $d(G) = d$ and minimum shortest path cover $\mathcal{S}^*$ of size s such that $qc(G) \geq |\mathcal{S}^*| \log d(G)$.*

For many graphs, however, we can do much better, as the labels of different paths in the cover are typically not independent. Consider for instance a k-dimensional hypergrid with edge length ℓ, that is, the Cartesian product of k paths with ℓ vertices each: More than $k^{\ell-1}/\ell$ shortest paths are needed to cover the hypergrid but $1 + \lceil \log(k\ell) \rceil$ queries suffice to identify the labelling of the graph.

This leads directly to an idea for improving the bound and deriving our algorithm for active learning geodesically convex halfspaces: after each query, we deduce additional labels using convex hulls and extensions. The algorithm first queries the vertices of a hull set until it finds two vertices with different labels or concludes that all vertices in the graph have the same label. Then it performs a binary search on a shortest path between these two vertices to identify a cut edge $\{a, b\}$. Finally, Algorithm 1 efficiently queries the remaining vertices. It initialises the sets A and B with the convex hull of rays a/b and b/a, respectively. The main loop queries any vertex v in the region $\hat{W}_{=ab} = V(G) \setminus (\sigma(a/b) \cup \sigma(b/a))$ and updates the known labels with the new vertex using convex hulls and extensions. The queries needed in the first two steps are summarised in Lemma 4.

Lemma 4. *If a weighted graph G with halfspace separable labels has at least one cut edge, it can be found with $h(G) + \lceil \log d(G) \rceil$ queries.*

Lemma 5 shows that the initialisation and updates of A and B in Algorithm 1 are indeed valid.

Lemma 5. *Let G be a weighted graph with halfspace separable labels λ and let $A \subseteq \{x \in V(G) \mid \lambda(x) = 1\}$ and $B \subseteq \{x \in V(G) \mid \lambda(x) = 0\}$. For any $v \in \sigma(A/B)$, it holds that $\lambda(v) = 1$*

We can bound the number of queried vertices of Algorithm 1 using the set $W^*_{=ab} = \hat{W}_{=ab} \setminus \{v \in \hat{W}_{=ab} \mid \exists w \in \hat{W}_{=ab}$ such that $v \in w/a \cap w/b\}$. Intuitively, it is enough to know the labels of the vertices in $W^*_{=ab}$ to deduce all labels of $\hat{W}_{=ab}$. Theorem 6 shows that Algorithm 1 is correct and summarises our main upper bound.

Algorithm 1: Halfspace querying

Input: graph G, cut edge $\{a, b\} \in E(G)$

$A := \sigma(a/b), \quad B := \sigma(b/a)$

while $A \cup B \neq V(G)$ **do**

  query any vertex $v \in V(G) \setminus (A \cup B)$

  if $\lambda(v) = \lambda(a)$ **then**

    $A := \sigma((A \cup \{v\})/B), \quad B := \sigma(B/(A \cup \{v\}))$

  else

    $A := \sigma(A/(B \cup \{v\})), \quad B := \sigma((B \cup \{v\})/A)$

Theorem 6. *Let G be a weighted or unweighted graph. Finding a cut edge with a minimum hull set and then applying Algorithm 1, results in the query complexity*

$$\mathrm{qc}(G) \leq h(G) + \log d(G) + \max_{\{a,b\} \in E(G)} |W^*_{=ab}| \,.$$

To give a more intuitive bound for the unweighted case, we relate the last term to the *treewidth* $\mathrm{tw}(G)$, a well-studied graph-theoretic parameter [Bodlaender, 1996] measuring the 'tree-likeness' of a graph: Any cut edge $\{a, b\}$ together with $W^*_{=ab}$ has a complete bipartite minor $K_{2,|W^*_{=ab}|}$ which implies $\mathrm{tw}(G) \geq {}^{|W^*_{=ab}|}/_2$ [Bodlaender et al., 1997]. This means that many real-world graphs such as molecules [Horváth and Ramon, 2010] and infrastructure-based networks [Maniu et al., 2019] which have small treewidth, also have small $W^*_{=ab}$ and can be queried efficiently by Algorithm 1. Corollary 7 summarises this result.

Corollary 7. *Let G be a unweighted graph with hull set H and treewidth $\mathrm{tw}(G)$. Then, for the query complexity it holds that $\mathrm{qc}(G) \leq h(G) + \log d(G) + 2\,\mathrm{tw}(G)$.*

3.2 Lower bounds under separation axioms

Having discussed our upper bounds on the query complexity, we now turn to a simple lower bound based on the extreme vertices $\mathrm{Ext}(G)$ of the graph, recall each of them is a halfspace by definition.

Theorem 8. *For a weighted graph G with extreme vertices $\mathrm{Ext}(G)$, it holds that $qc(G) \geq |\,\mathrm{Ext}(G)|$.*

Considering a graph consisting of paths that coincide only in their endpoints, we can show that this bound is also tight and therefore that the gap between it and our upper bound can be arbitrarily large.

Proposition 9. *For any $d, s \in \mathbb{N}$, there exists an unweighted graph G with diameter $d(G) = d$ and minimum shortest path cover $\mathcal{S}^*$ of size s, such that $\mathrm{qc}(G) \leq |\,\mathrm{Ext}(G)| = 2$.*

This shows that the lower bound in Theorem 8 is best possible for general graphs, even when $d(G)$ and $\mathcal{S}^*$ are large. The main insight of our work is that structuring the set of all graphs along separation axioms gives increasingly tighter lower bounds.

Theorem 10. *For a weighted graph G the following holds for the query complexity $\mathrm{qc}(G)$:*

- *if G is S_2, then $\mathrm{qc}(G) \geq \max(\log d(G), |\,\mathrm{Ext}(G)|)$,*
- *if G is S_3, then $\mathrm{qc}(G) \geq \max(\log d(G), h(G))$, and*
- *if G is S_4, then $\mathrm{qc}(G) \geq \max(\log d(G), h(G), r(G) - 1)$.*

Each bound is tight in the respective family and stronger axioms lead to tighter bounds.

To see the final statement, notice that all extreme vertices are by definition contained in any hull set and hence $|\,\mathrm{Ext}(G)| \leq h(G)$. The two parameters $h(G)$ and $r(G)$ are independent of each other, in the sense that there are graphs with $h(G) < r(G)$ but also graphs with $r(G) < h(G)$. Theorems 6 and 10 together imply tight bounds (Table 1) for graph families studied in metric graph theory [Bandelt and Chepoi, 2008], convexity theory, and machine learning [Seiffarth et al., 2019].

Table 1: Tight bounds for specific unweighted graph families

graph family	bound on $\mathrm{qc}(G)$	remark
trees	$\Theta(\log d(G) + \lvert \mathrm{Ext}(G) \rvert)$	$\mathrm{Ext}(G)$ are exactly the leaves
$K_{2,3}$ minor-free	$\Theta(\log d(G) + h(G))$	including outerplanar graphs
partial cubes	$\Theta(\log d(G) + h(G))$	$\mathcal{O}(\cdot)$ holds for all bipartite graphs
weakly median	$\Theta(\log d(G) + h(G) + r(G))$	$\mathcal{O}(\cdot)$ holds for meshed graphs

3.3 Computational aspects

Above, we derived lower and upper bounds on the query complexity of learning halfspaces on graphs $G = (V, E)$. We now discuss computational aspects of our algorithm and bounds.

Computing a minimum shortest path cover $\mathcal{S}^*$ is an open problem [Manuel, 2018]. We show that greedily covering the graph with longest possible shortest paths gives a logarithmic approximation:

Theorem 11. *For a given weighted graph G, we can compute a $\mathcal{O}(\log d(G))$-approximation for the minimum shortest path cover $\mathcal{S}^*$ in time $\mathcal{O}(|V|^4)$.*

Computing the Radon number of a graph is even hard to approximate within any factor sublinear in $|V|$ [Coelho et al., 2015]. However, we can bound the Radon number with the *Hadwinger number* of the graph, that is the size of the largest clique minor, which is upper bounded by the treewidth.

Proposition 12. *For any graph G, it holds that $r(G) \leq \mathrm{tw}(G) + 2$.*

Computing intervals and hulls is possible in $\mathcal{O}(|V(G)|^3)$ time by first solving the all-pairs-shortest-path problem and afterwards check the triangle inequality for each triplet of vertices. The diameter $d(G)$ is computable by a generalised Dijkstra algorithm, see supplementary. Computing the set of extreme vertices $\mathrm{Ext}(G)$ is possible in time $\mathcal{O}(|V||E|)$ by evaluating the neighbourhood of each vertex. **Computing the minimum hull set size** $h(G)$ is APX-hard [Coelho et al., 2015]. For graphs of bounded treewidth [Kanté et al., 2019] however, the problem is solvable in polynomial time and our algorithm achieves then the bound of Corollary 7 in polynomial time. For other graphs heuristic approaches can compute a small hull set H and achieve the query complexity bounds in Theorem 6 using $|H|$ queries instead of $h(G)$. **Deciding whether all separation axioms hold** is possible in time $\mathcal{O}(|V(G)|^7)$ by examining all quadruplets of vertices, so-called *Pasch* combinations [Seiffarth et al., 2019]. Such strategies for the other cases are unknown. **Deciding whether a graph has a proper halfspace**, that is, a halfspace that is not empty nor the full graph, is NP-hard [Artigas et al., 2011]. For planar graphs, however, it can be solved in polynomial time [Glantz and Meyerhenke, 2017]. Real-world graphs often contain extreme vertices which themselves are proper halfspaces.

4 Discussion

In this section, we discuss our assumption as well as related active and *passive* learning results.

Convexity in real-world datasets In some applications convexity-based assumptions are already implicitly or explicitly used. For example in gene similarity networks, it is well-known that shortest paths largely preserve functional relationships and the category in the gene ontology [Zhou et al., 2002]. Similar results were achieved on protein-protein-interaction networks, where shortest paths between known cancer-related genes are used to identify a candidate set of novel genes that are likely to be cancer-related, as well [Li et al., 2012, 2013]. Aside from protein and gene networks, Marc and Šubelj [2018] and Šubelj et al. [2019] recently found that connected subgraphs of real-world graphs like collaboration networks are often convex. To provide further evidence for convexity in real-world graphs, we performed preliminary experiments on different networks with ground truth communities. Table 2 shows the number of convex communities in six datasets from SNAP [Leskovec and Sosič, 2016]. We found that on the DBLP scientific collaboration dataset more than 85% of the 5000 communities are convex, supporting the results of Šubelj et al. [2019]. On the Amazon product network we have similar results with roughly 80%. On the Youtube social network we found that roughly 60% of the communities are convex. On the remaining three datasets less than a third of the communities are convex. Clearly convexity on graphs is application dependent.

Table 2: Number of convex communities.

dataset	convex communities	dataset	convex communities
DBLP	$4308/5000$	Amazon	$3999/5000$
Youtube	$2990/5000$	LiveJournal	$1649/5000$
Orkut	$363/5000$	Eu-core	$7/42$

Relationships to Euclidean convexity Halfspaces in Euclidean space have been at the core of various learning algorithms, like the Perceptron [Rosenblatt, 1958, Novikoff, 1963] and support vector machines [Boser et al., 1992, Cortes and Vapnik, 1995], and are still an active research area [Daniely, 2016, Diakonikolas et al., 2020, Hopkins et al., 2020]. A simple lower bound for the number of required queries to learn the labels of a finite set $X \subseteq \mathbb{R}^m$ labelled accordingly to a halfspace is $\Omega(m \log |X|)$ [Hopkins et al., 2020]. Already for points on the unit circle, $\Omega(|X|)$ queries are required to identify all labels [Dasgupta, 2005]. The query complexity can be improved through membership query synthesis [Angluin, 1988], which allows the algorithm to query additioanally any point in $\mathbb{R}^m \setminus X$. There is no direct equivalent of membership queries in the transductive graph setting. Using them, Hopkins et al. [2020] showed that in expectation $\mathcal{O}(m \log^2(m) \log |X|) + m^{1+o(1)}$ queries are enough to infer all labels, essentially matching the lower bound. Graph and other finite convexity spaces, however, are not equivalent to the Euclidean convexity in the sense that it is not always possible to embed a graph into $\mathbb{R}^m$ or the other way around while preserving the convex hull $\sigma(\cdot)$. An example can be seen in Figure 1c and further discussion can be found in the supplemantary. As furthermore Euclidean convexity spaces are always S_4 while graphs do not have to be, there is no straightforward way to transform bounds on the query complexity from one setting to the other.

Cut-based bounds Most previous bounds for active learning on graphs are linear in the size of the set of cut edges C and *cut vertices* ∂C, that are the vertices incident to the cut edges. One such bound was given by Guillory and Bilmes [2009], subsequently further improved [Guillory and Bilmes, 2011] and extended by the line of work of Cesa-Bianchi et al. [2010]. We will illustrate the bounds on the k-dimensional hypercube graph labelled according to some halfspace. In our setting, the labels can be deduced with binary search on a shortest path connecting opposing corners using $\mathcal{O}(\log k)$ queries. Let $\delta(T)$ denote the set of outgoing edges of $T \subseteq V$. Guillory and Bilmes [2009] introduced $\Psi(L) = \min_{\emptyset \neq T \subseteq (V \setminus L)} |\delta(T)|/|T|$ to bound the error of the min-cut-based prediction strategy [Blum and Chawla, 2001] as $|C|/\Psi(L)$. Guillory and Bilmes [2011] thus propose to select a set L that maximises $\Psi(L)$ to make the bound smallest possible. On the hypercube with halfspace separable labels, the cut size $|C|$ is always $V/2$. Independent of the selection strategy and the size of the set L, it can be shown that $|L| + |C|/\Psi(L) \geq |V|$ on this example, see supplementary. Afshani et al. [2007] gave a worst-case query complexity of $c + |\partial C|(1 + \log(|V|/|\partial C|))$ under balancedness assumptions where $c \geq 2$ is the number of connected components in G without the cut edges C. As $\partial C = V$ this results in a bound $> |V|$ for the hypercube. The more recent bound of Dasarathy et al. [2015], besides the number of cut vertices, depends on the *clusteredness* κ of the cut, which is at least the largest distance between any two cut vertices of the same class in the same connected component of the graph with cut edges removed. Their upper bound is at least $|\partial C|(1 + \log \kappa)$. On the hypercube $\kappa \geq k - 1$ and $|\partial C| = 2^k$ resulting again in a bound $> |V|$. This shows that discussed previous bounds can be vacuous, that is $\geq |V|$, and more than exponentially worse than our achieved bound. All three of them are particularly sensible to large cut sizes. The sole number of cut vertices $|\partial C|$ is not sufficient to measure the complexity inherent in the labelling.

Balancedness assumptions Besides to a small cut, Afshani et al. [2007] and Dasarathy et al. [2015] also assume that the labels are *balanced*, that is the relative size β of the smallest labelled class is close to $1/2$. This assumption is required to find two vertices with different labels quickly with high probability $1 - \alpha$ for $\alpha \in (0, 1)$. We do not require this assumption as we can use any hull set to achieve the same deterministically. However, if β is known, we can with high probability improve our upper bounds. Instead of spending $h(G)$ queries we can sample a set L of size $\frac{\log(\beta\alpha)}{\log(1-\beta)}$ uniformly at random to find two different labelled vertices with probability $1 - \alpha$ [Dasarathy et al., 2015].

Margin-based bounds Bressan et al. [2021] study the query complexity of identifying geodesically convex clusters in a nearest neighbour graph given by a semi-metric space using *same-cluster* queries. Even though they rely on additional assumptions, like a large margin, their upper bound is very similar to our main result. However, they only provide lower bounds on the query complexity on specific worst-case instances, while our lower bounds hold in general under the different separation axioms. Same-cluster queries, which are binary queries on a pair of points to test whether they are in the same cluster, instead of vertex label queries. Same-cluster queries and vertex queries are equivalent in terms of the query complexity up to a multiplicative factor given by the number of classes. For an $\varepsilon \in (0, 1]$, they remove any edge with weight larger than $\varepsilon d(G)$, assume that the weight of any cut edge is $> \beta\varepsilon$ for an $\beta \in (0, 1]$, and use an interval with margin $\gamma \in (0, 1]$: $I_\gamma(a, b) = \{x \in V \mid d(a, x) + d(x, b) \leq (1 + \gamma)d(a, b)\}$ to define convex sets. The regular geodesic convexity corresponds to $\gamma = 0$. Using these, they state an $\mathcal{O}\left(\log |X| + \left(\frac{6}{\beta\gamma}\right)^{\mathrm{dens}(X)}\right)$ upper bound on the query complexity for any semi-metric space X with binary labels under the assumption that two vertices with different labels are already given, where $\mathrm{dens}(X)$ is the *density dimension* of the space [Gottlieb and Krauthgamer, 2013]. Even though they make stronger assumptions to achieve this bound, there exist graph families where our bound in Theorem 6 is exponentially better. For example, take again the $2 \times k$ grid. If we allow any halfspace, the margin is $\gamma < 2/k$. This results in a bound of at least $\log(2k) + (3k)^{\mathrm{dens}(X)}$ with $\mathrm{dens}(X) \geq 1$ while our bound predicts that $2 + \log(k + 1)$ queries are enough. Even with a large margin our bound remains slightly better.

Passive learning halfspaces Passive learning halfspaces on graphs was studied by Seiffarth et al. [2019, 2020] and de Araújo et al. [2019]. Seiffarth et al. [2019] rely on the separation axiom S_4 to bound the number of convex hull computations needed to construct a halfspace. Anthony and Ratsaby [2018] study the problem of learning large-margin halfspaces on a set equipped with a *dissimilarity measure* d, which is more genral than the shortest path distance of a graph. However, their notion of halfspace, $\{x \mid d(x, p^+) \leq d(x, p^-)\}$, is based on the dissimilarity to two prototypes p^+, p^- and in general not compatible with our convexity based results. Stadtländer et al. [2021] investigate the more general problem of learning *weakly convex* sets in a metric space (X, d). Here, weakly convex sets with parameter $\theta \geq 0$ are given by the interval mapping $I_\theta(a, b) = \{x \in X \mid d(a, x) + d(x, b) = d(a, b) \leq \theta\} \cup \{a, b\}$ corresponding to our interval mapping on a pair a, b with $d(a, b) \leq \theta$. Moran and Yehudayoff [2019] established that the VC dimension of halfspaces of any convexity space is smaller than the Radon number of the space. Building on that, we can show that VC dimension is exactly determined by the Radon number in S_4 convexity spaces. This includes the classical result that in $\mathbb{R}^m$ the VC dimension of halfspaces is $m + 1$ and the Radon number is $m + 2$.

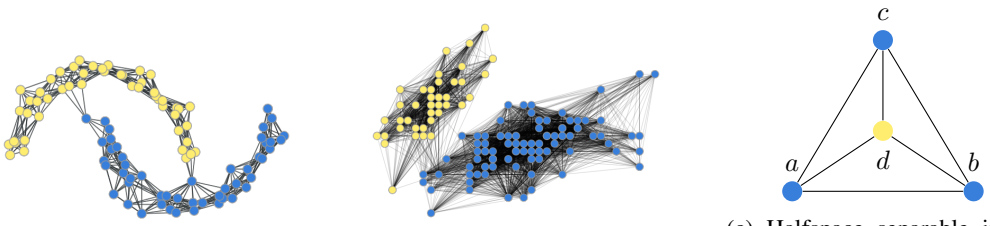
Proposition 13. *The VC dimension of the hypothesis class of halfspaces of an S_4 convexity space is exactly one less than the Radon number of the space.*

For S_4 graphs, $r(G)$ determines the VC dimension and gives a lower bound on the query complexity (Theorem 10), establishing the Radon number as a central parameter for learning geodesic halfspaces. Proposition 12 together with the above bound of Moran and Yehudayoff [2019] establishes:

Proposition 14. *The VC dimension of halfspaces in a graph G is smaller than* $\mathrm{tw}(G) + 2$.

5 Experiments

Having discussed our bounds on the query complexity and the conceptional benefits of the halfspace assumption in terms of theoretical upper bounds, we want to see whether the same holds experimentally, as well. Two practical querying strategies based on Algorithm 1 are *greedy* and *selective sampling*.The greedy strategy tries to maximise the number of known vertex labels with each query. For that, it queries the vertex v that would maximise the minimum number of known labels after the update of A and B in Algorithm 1. More precisely, after the update with a vertex v we will know the labels of the set $\sigma((A \cup \{v\})/B) \cup \sigma(B/(A \cup \{v\}))$ or the set $\sigma(A/(B \cup \{v\})) \cup \sigma((B \cup \{v\})/A)$. The greedy strategy thus selects the vertex v that maximises the size of the smaller one of these two sets, corresponding to the minimum number of labels we will know after the update. As performing the described greedy maximisation for each query is rather computationally expensive, we propose a simpler strategy based on selective sampling [Cohn et al., 1994] as an alternative. Instead of performing the actual maximisation, this strategy simply picks a vertex uniformly at random from $V(G) \setminus (A \cup B)$ in line 3 of Algorithm 1. To find the first cut edge, we use a hull set as in Lemma 4.



(a) Two moons NN-graph. (b) Iris dataset NN-graph. (c) Halfspace separable in the graph, but not in $\mathbb{R}^2$.

Figure 1: Halfspace separations on different graphs.

As discussed, computing a minimum hull set is in general not tractable. Therefore, we propose to use the selection strategy (greedy or selective sampling) also to iteratively construct a hull set until two vertices with different labels are found. For that we start with $A, B = \emptyset$ and select vertices using the selection strategy and update A and B as in Algorithm 1 accordingly, until we find two different labels. Implicitly this process will construct a (typically non-minimum) hull set H. That way the query complexity bound in Theorem 6 still holds, as it is independent of the specific selection strategy used; only the $h(G)$ is replaced with the size of the heuristically computed H. We compare these two strategies with the state-of-the-art graph-based active learning algorithm S^2 [Dasarathy et al., 2015], the classical active label propagation of Zhu et al. [2003b], and baseline non-active random sampling.

We use the ε-nearest-neighbour graphs of two moons[1] and Iris[2], see Figures 1a and 1b. The parameter ε was selected such that the graph is connected and the labels are halfspace separable. Additionally, we constructed a 20×20 grid and a 2^{10} hypercube, labelled with a random halfspace. We implemented all approaches in `python` and ran the experiments on an Ubuntu 21.04 laptop with 32GB main memory. The data, code, and instructions to reproduce the results are in the supplementary.

5.1 Query evaluation

Dasarathy et al. [2015] propsed to count the number of found cut vertices after each query as a measure to compare querying algorithms without relying on any classification method. An efficient graph-based active learner should detect these as fast as possible. The results for 20 iterative queries in the upper half of Figure 2 show that our approaches identify the cut vertices more efficiently than S^2, active label propagation, and random sampling by deducing many labels after each query. Our greedy approach requires less than 5 queries to deduce all vertex labels on all four datasets emphasising the strength of convexity-based assumptions. The selective sampling approach is only slightly worse using 2-8 queries more. S^2, active label propagation, and random sampling can find at most one cut vertex per query as it does not rely on these assumptions. As can be seen on the grid dataset, S^2 performs significantly better than active label propagation and random sampling finding a cut vertex on any new query, which is not by coincidence as it was specifically designed for this task.

5.2 Predictive performance

We also evaluated the predictive performance of the chosen queries. Our two approaches predict the labels of the computed hulls and default to the majority of known labels when they do not know the label of a vertex. The other three approaches perform label propagation [Zhu et al., 2003a] with the default Gaussian similarity and $\sigma = 1$ on top of the selected vertices.

Again we let each method iteratively select 20 vertices and checked the accuracy on the whole graph after each query. In the lower half of Figure 2, we can see that again our greedy approach identifies the vertex labels of the whole graph on all datasets within 5 queries. The predictive behaviour of the other three approaches is significantly worse, even unstable. The main reason is that label propagation favours small cuts (weighted by the similarity), and thus fails to predict halfspaces, especially on the Iris dataset and hypercube. S^2 could mitigate this problems to some extent, even though it predicts the labels also with label propagation, by removing any found cut edge from the graph. Still, it has difficulties to find the correct halfspace and then stick to these prediction.

[1] $\varepsilon = 0.14$, `make_moons(noise=0.1, random_state=0)` in `scikit-learn` [Pedregosa et al., 2011]
[2] $\varepsilon = 0.3$, first class vs the other two [Fisher, 1936]

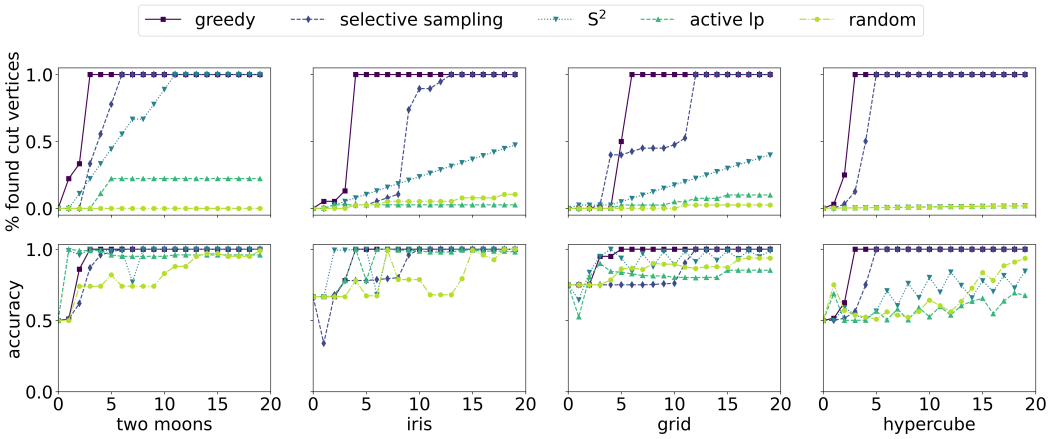


Figure 2: Found cut vertices and accuracy plotted against number of queries

6 Conclusion and Future Work

In this work, we employed convexity theoretical concepts to achieve novel bounds on the query complexity of learning halfspaces on graphs. On the one hand, we derived two general upper bounds, one based on shortest path covers and the other based on the diameter, hull sets, and the quantity $\max_{\{a,b\}\in E(G)} |\hat{W}^*_{=ab}|$, which we in turn bounded by the treewidth of the graph. On the other hand, we stated a simple general lower bound based on extreme vertices and increasingly tighter lower bounds for graph families corresponding to separation axioms. For the strongest separation axiom S_4, we achieved almost matching lower and upper bounds, where only the gap between the Radon number $r(G)$ (as a lower bound) and the treewidth $\mathrm{tw}(G)$ (as an upper bound) remains to be closed.

We compared our bounds to previous results based on the cut-size and found that the halfspace assumption enabled learning with large cuts, while previous bounds often become vacuous in this case. Comparing our results to similar bounds in Euclidean halfspace learning, we discussed that they are largelly incompatible due to inherent differences in the convexity spaces. On the practical side, we found that ground-truth communities in real-world networks such as collaboration and product networks tend to be convex. Based on our theoretical bounds, we devised two practical querying strategies, a greedy and a selective-sampling based approach. We compared them with two previous active learning strategies on halfspace separable datasets in terms of the number of queries they needed to find a cut and their predictive performance. We found that our algorithms require considerably less queries to identify the correct halfspace. The main bottleneck of the algorithms is the cubic computational complexity that is required to compute convex hulls $\sigma(\cdot)$. To scale our methods to large datasets we will investigate efficient approximations of convex hulls and distances.

Weakly convex sets, as studied by Stadtländer et al. [2021], could be used to relax the notion of halfspaces and make our results more broadly applicable. In particular, Stadtländer et al. [2021] state that any weakly convex set with respect to a parameter $\theta \geq 0$ can be decomposed into weakly convex *blocks*, separated by a distance of more than θ. The number of these blocks forms a reasonable parameter which probably can be used to adapt our bounds to weakly convex classes. Unifying the ideas of Seiffarth et al. [2019], Stadtländer et al. [2021], and Bressan et al. [2021] with ours by developing querying and prediction strategies on general interval convexity spaces is an interesting future research direction. As a first step to generalise our results to *geometric* interval spaces [van de Vel, 1993] with *base-point orders* is highly promising. It might be also interesting to consider online learning of halfspaces and other convex sets on graphs.

Broader impact statement As this is an early theoretical and algorithmic work, we do not see any specific immediate societal impact, neither positive nor negative. In the long run the developed theory might help to reduce the required number of labelled data points in graph-based learning settings and help with data annotation in general.

References

Peyman Afshani, Ehsan Chiniforooshan, Reza Dorrigiv, Arash Farzan, Mehdi Mirzazadeh, Narges Simjour, and Hamid Zarrabi-Zadeh. On the complexity of finding an unknown cut via vertex queries. In *International Computing and Combinatorics Conference*, 2007.

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

Martin Anthony and Joel Ratsaby. Large-width bounds for learning half-spaces on distance spaces. *Discrete Applied Mathematics*, 243:73–89, 2018.

Danilo Artigas, Simone Dantas, Mitre C. Dourado, and Jayme L. Szwarcfiter. Partitioning a graph into convex sets. *Discrete Mathematics*, 311(17):1968–1977, 2011.

Hans-Jürgen Bandelt. Graphs with intrinsic s3 convexities. *Journal of graph theory*, 13(2):215–228, 1989.

Hans-Jürgen Bandelt and Victor Chepoi. Metric graph theory and geometry: a survey. *Contemporary Mathematics*, 453:49–86, 2008.

Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.

Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on computing*, 25(6):1305–1317, 1996.

Hans L. Bodlaender, Jan van Leeuwen, Richard Tan, and Dimitrios M. Thilikos. On interval routing schemes and treewidth. *information and computation*, 139(1):92–109, 1997.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, 1992.

Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, and Andrea Paudice. Exact recovery of clusters in finite metric spaces using oracle queries. In *COLT*, 2021.

Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Active learning on trees and graphs. In *COLT*, 2010.

Erika M. M. Coelho, Mitre C. Dourado, and Rudini M. Sampaio. Inapproximability results for graph convexity parameters. *Theoretical Computer Science*, 600:49–58, 2015.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *STOC*, 2016.

Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In *COLT*, 2015.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, 2005.

Paulo H.M. de Araújo, Manoel Campêlo, Ricardo C. Corrêa, and Martine Labbé. The geodesic classification problem on graphs. *Electronic Notes in Theoretical Computer Science*, 346:65–76, 2019.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In *COLT*, 2020.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2):179–188, 1936.

Roland Glantz and Henning Meyerhenke. On finding convex cuts in general, bipartite and plane graphs. *Theoretical Computer Science*, 695:54–73, 2017.

Lee-Ad Gottlieb and Robert Krauthgamer. Proximity algorithms for nearly doubling spaces. *Journal on Discrete Mathematics*, 27(4):1759–1769, 2013.

Andrew Guillory and Jeff Bilmes. Label selection on graphs. In *NIPS*, 2009.

Andrew Guillory and Jeff Bilmes. Active semi-supervised learning using submodular functions. In *UAI*, 2011.

M. Hopkins, D. Kane, S. Lovett, and G. Mahajan. Point location and active learning: Learning halfspaces almost optimally. In *FOCS*, 2020.

Tamás Horváth and Jan Ramon. Efficient frequent connected subgraph mining in graphs of bounded tree-width. *Theoretical Computer Science*, 411(31-33):2784–2797, 2010.

Shizuo Kakutani. Ein Beweis des Satzes von M. Eidelheit über konvexe Mengen. *Proceedings of the Imperial Academy*, 13(4):93–94, 1937.

Mamadou Moustapha Kanté, Thiago Marcilon, and Rudini Sampaio. On the parameterized complexity of the geodesic hull number. *Theoretical Computer Science*, 1:1–18, 2019.

Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

Bi-Qing Li, Tao Huang, Lei Liu, Yu-Dong Cai, and Kuo-Chen Chou. Identification of colorectal cancer related genes with mrmr and shortest path in protein-protein interaction network. *PloS one*, 7(4):e33393, 2012.

Bi-Qing Li, Jin You, Lei Chen, Jian Zhang, Ning Zhang, Hai-Peng Li, Tao Huang, Xiang-Yin Kong, and Yu-Dong Cai. Identification of lung-cancer-related genes with the shortest path approach in a protein-protein interaction network. *BioMed research international*, 2013.

Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In *International Conference on Database Theory*, 2019.

Paul Manuel. Revisiting path-type covering and partitioning problems. *arXiv preprint arXiv:1807.10613*, 2018.

Tilen Marc and Lovro Šubelj. Convexity in complex networks. *Network Science*, 6(2):176–203, 2018.

Shay Moran and Amir Yehudayoff. On weak epsilon-nets and the Radon number. In *Symposium on Computational Geometry*, 2019.

Albert B. Novikoff. On convergence proofs for perceptrons. Technical report, Stanford Research Institute, Menlo Park, California, 1963.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ignacio M. Pelayo. *Geodesic convexity in graphs*. Springer, 2013.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Florian Seiffarth, Tamás Horváth, and Stefan Wrobel. Maximal closed set and half-space separations in finite closure systems. In *ECMLPKDD*, 2019.

Florian Seiffarth, Tamás Horváth, and Stefan Wrobel. Maximum margin separations in finite closure systems. In *ECMLPKDD*, 2020.

Eike Stadtländer, Tamás Horváth, and Stefan Wrobel. Learning weakly convex sets in metric spaces. *arXiv preprint arXiv:2105.06251*, 2021.

Lovro Šubelj, Dalibor Fiala, Tadej Ciglarič, and Luka Kronegger. Convexity in scientific collaboration networks. *Journal of Informetrics*, 13(1):10–31, 2019.

Marcel L.J. van de Vel. *Theory of convex structures*. Elsevier, 1993.

Xianghong Zhou, Ming-Chih J Kao, and Wing Hung Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences*, 99(20):12783–12788, 2002.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003a.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, 2003b.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We shortly discussed, that as this is an early theoretical work, we do not see any specific immediate societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the supplemental material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Only in the supplemental material, due to space limitations and too small plots.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Two small synthetic datasets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]