Online learning of convex sets on graphs

Maximilian Thiessen(✉) and Thomas Gärtner

Research Unit of Machine Learning, TU Wien, Vienna, Austria
maximilian.thiessen@tuwien.ac.at, thomas.gaertner@tuwien.ac.at

Abstract. We study online learning of general convex sets and halfspaces on graphs. While online learning of halfspaces in Euclidean space is a classical learning problem, the corresponding problem on graphs is understudied. In this context, a set of vertices is convex if it contains all connecting shortest paths and a halfspace is a convex set whose complement is also convex. We discuss mistake bounds based on the Halving algorithm and shortest path covers. Halving achieves near-optimal bounds but is inefficient in general. The shortest path cover based algorithm is efficient but provides optimal bounds only for restricted graph families such as trees. To mitigate the weaknesses of both approaches, we propose a novel polynomial time algorithm which achieves near-optimal bounds on graphs that are $K_{2,k}$ minor-free for some constant $k \in \mathbb{N}$. In contrast to previous mistake bounds on graphs, which typically depend on the induced cut of the labelling, our bounds only depend on the graph itself. Finally, we discuss the agnostic version of this problem and introduce an adaptive variant of Halving for k-intersections of halfspaces.

Keywords: online learning · graph convexity · node classification.

1 Introduction

We study online learning of halfspaces and general convex sets on graphs. While most previous mistake bounds in online learning on graphs are based on the *cut-size*, that is, the number of edges with differently labelled endpoints, we focus on *label-independent* bounds, which can be computed directly from the graph itself. Our approach makes small mistake bounds possible even if the cut-size is large. To achieve that, we assume that the vertices with positive labels are convex in the graph. Here convex means that the vertex set is connected and belongs to some intersection-closed hypothesis space. We will focus on the *geodesic* convexity defined by shortest paths. In the special case of halfspaces, where both the positively and negatively labelled vertex sets are convex, we prove a strong bound given solely by the *diameter* of the input graph and the size of a the largest complete bipartite graph $K_{2,m}$ that is a minor of the input graph.

While the problem of online learning halfspaces in Euclidean space is a classical machine learning problem [34,32], the graph variant of this problem has not yet been studied. Convexity on graphs is a well-studied topic outside of the field of machine learning [15,14,33]. However, only recently it was used in learning problems. Seiffarth et al. [35] initiated the study of node classification

under the assumption that both classes are geodesically convex, hence halfspaces. Stadtländer et al. [36] studied a more relaxed version of geodesic convexity allowing multiple disconnected convex regions, called *weak convexity*. In a previous work, we study the active learning version of this problem and have shown near-tight bounds on the query complexity of learning halfspaces [37]. Bressan et al. [6] developed bounds for the same problem under additional margin assumptions and stronger query oracles. While active learning of general convex sets on graphs is not possible with a sub-linear query bound, as it requires to query the whole graph already in the case of a single path, non-vacuous mistake bounds in the online setting can still be achieved, as we will show. Previously, the special case of learning monotone classes on partially ordered sets has been studied [18,31]. Monotone classes can be seen as halfspaces under the *order convexity* [38] on directed acyclic graphs, hence a special case of graph convexity spaces.

2 Background

We introduce necessary concepts in online learning and convexity spaces.

2.1 Online learning

We will focus on the *realisable online-learning* setting [28]. Given a set X and a hypothesis space $\mathcal{H} \subseteq \{h(\cdot) \mid h : X \to \{-1, 1\}\}$, our learner A_t knows $\mathcal{H}$ and its strategy might change in each round $t \in \mathbb{N}$. It plays the following iterative game against a potentially adversarial opponent. Any round t has the following steps:

1. opponent picks $x_t \in X$;
2. learner predicts $A_t(x_t) = \hat{y}_t \in \{-1, 1\}$ as the label of x_t;
3. opponent reveals the correct label $y_t \in \{-1, 1\}$ to the learner;
4. if $\hat{y}_t \neq y_t$ the learner makes a mistake;
5. A_t potentially updates its strategy.

The opponent is forced to play realisable, that is, there is an $h \in \mathcal{H}$ such that $y_t = h(x_t)$ for all $t \in \mathbb{N}$. The learner's predictions $\hat{y}_t$ are allowed to be *improper*, that is, there is not necessarily a hypothesis in $\mathcal{H}$ determining $\hat{y}_t$. Let $M_A(h)$ denote the worst-case number of mistakes an algorithm A makes on any sequence of points labelled by $h \in \mathcal{H}$. The goal is to minimise the worst-case number of mistakes over all hypotheses $M_A(\mathcal{H}) = \max_{h \in \mathcal{H}} M_A(h)$.

For any given hypothesis space $\mathcal{H}$, a lower bound on the number of mistakes for any online learning algorithm is given by the *Littlestone dimension* $\mathrm{Ldim}(\mathcal{H})$, which is the size of the largest *mistake tree* [28]. It is a combinatorial quantity similar to the VC dimension $\mathrm{VC}(\mathcal{H})$. The *Standard Optimal Algorithm* (SOA) [28] achieves the optimal mistake bound $\mathrm{Ldim}(\mathcal{H})$ for any finite hypothesis space. In general, it is intractable as it requires computing $\mathrm{Ldim}(\mathcal{H}')$ for multiple $\mathcal{H}' \subseteq \mathcal{H}$ in each step, which is known to be hard [16].

2.2 Convexity spaces

For a thorough introduction on convexity theory and graph convexity theory we refer the reader to [38] and [33].

For a set X and a family $\mathcal{C} \subseteq 2^X$ of subsets, the pair $(X, \mathcal{C})$ 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 unions of sets totally ordered by inclusion. For finite set systems, property *(iii)* always holds. The sets in $\mathcal{C}$ are called *convex*. If a set C and its complement $X \setminus C$ are convex, both are called *halfspaces*. We denote by $\mathcal{C}_H \subseteq \mathcal{C}$ the set of halfspaces of the convexity space $(X, \mathcal{C})$. Note that in general $\mathcal{C}_H$ is not intersection-closed. Two disjoint sets A, B are *halfspace separable* if there exists a halfspace C such that $A \subseteq C$ and $B \subseteq X \setminus C$. *Separation axioms* characterise the ability of a convexity space to separate sets via halfspaces.

Definition 1 (Separation axioms [38]). *A convexity space $(X, \mathcal{C})$ is:*

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

If S_1 holds the remaining axioms are increasingly stronger, that is, $S_2 \Leftarrow S_3 \Leftarrow S_4$. 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 $\sigma(A) = \bigcap\{C \mid A \subseteq C \in \mathcal{C}\}$. A set $A \subseteq X$ is convex, that is $A \in \mathcal{C}$, if and only if it is equal to its convex hull, $A = \sigma(A)$. 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 $r(\mathcal{C})$ such that any subset of X of size $r(\mathcal{C})$ has a Radon partition.

A particular type of convexity is *interval convexity*. It is given by 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 C in an interval convexity space is convex if and only if $C = I(C)$. The convex hull is given by $\sigma(A) = \bigcup_{k=1}^{\infty} I^k(A)$, where $I^1(\cdot) = I(\cdot)$ and $I^{k+1}(\cdot) = I(I^k(\cdot))$. Well-known interval convexity spaces are *metric spaces* (X, ρ). There, the interval contains all the points for which the triangle inequality holds with equality: $I_\rho(x, y) = \{z \in X \mid \rho(x, y) = \rho(x, z) + \rho(z, y)\}$. In Euclidean space this corresponds to all points on a line segment and leads to the standard notion of convex sets.

We study convexity spaces induced by graphs. For a graph $G = (V, E)$, a convexity space $(V, \mathcal{C})$ is a *graph convexity space* if all $C \in \mathcal{C}$ are connected in the graph G. Typically, convex sets in graphs are defined through a set of paths $\mathcal{P}$ in the graph G. The set $\mathcal{P}$ could for example consists of all shortest or

induced paths in G, or all paths up to a certain length. Then one can define the interval mapping $I_{\mathcal{P}}(x,y) = \bigcup\{V(P) \mid P \in \mathcal{P} \text{ has endpoints } x \text{ and } y\}$, where $V(\cdot)$ denotes the vertex set of the corresponding graph. The most commonly studied convexity on graphs is the *geodesic convexity* (or shortest path convexity) where $\mathcal{P}$ is the set of shortest paths in G. For a connected graph $G = (V, E)$ it is given by the interval mapping I_d, where $d : V^2 \to \mathbb{R}$ is the *shortest path distance* in G. Let $x, y \in V$. For unweighted graphs $d(x, y)$ is the minimum number of edges on any x-y-path and for graphs with edge weights, $w : E \to \mathbb{R}_{>0}$, it is the minimum total edge weight of any x-y-path. A set of vertices $C \subseteq V$ is, thus, geodesically convex if and only if C contains every vertex on every shortest path joining vertices in C, corresponding again to the Euclidean case.

We denote the size of the geodesic minimum hull set in G as $h(G)$ and the induced subgraph given by a vertex set $X \subseteq V(G)$ as $G[X]$. The *diameter* $d(G)$ of a weighted or unweighted graph G is the maximum number of edges in any shortest path in G. We denote the treewidth of a graph, which is a measure of *tree-likeness*, as $\mathrm{tw}(G)$ [3]. Let $\mathrm{cbm}(G)$ be the largest integer m such that the complete bipartite graph $K_{2,m}$ is a minor of G. For $n \in \mathbb{N}$, we let $[n] = \{1, \ldots, n\}$.

3 Learning halfspaces

We start with the online learning of halfspaces, corresponding to the special case where the positive class and its complement, the negative class are convex. We start by discussing near-optimal bounds based on the Halving algorithm, which most likely has no polynomial runtime. After that we show how to use *shortest path covers* to get an efficient algorithm achieving near-optimal mistake bounds only on restricted graph families. We mitigate the weak points of both approaches by a novel polynomial-time algorithm that achieves near-optimal bounds on graphs with bounded $\mathrm{cbm}(G)$. See Table 1 for an overview on our resulting bounds.

Table 1: Overview on mistake bounds

	halfspaces	k-intersection	convex sets
Halving	$\mathcal{O}(r(G)\log\lvert V(G)\rvert)$	$\mathcal{O}(k\,r(G)\log\lvert V(G)\rvert)$	$\mathcal{O}(\mathrm{VC}(\mathcal{C})\log\lvert V(G)\rvert))$
tree-based	$\mathcal{O}(\mathrm{cbm}(G)^2\log d(G))$	/	/
shortest path cover	$\mathcal{O}(\lvert\mathcal{S}^*\rvert\log d(G))$	$\mathcal{O}(\lvert\mathcal{S}^*\rvert\log d(G))$	$\mathcal{O}(\lvert\mathcal{S}^*\rvert\log d(G))$

3.1 Halving

The well-known Halving algorithm is a very simple yet near-optimal approach to online learning. Let $\mathbb{1}_{\{\cdot\}}$ be the indicator function and $\mathrm{VS}_t = \{h \in \mathcal{H} \mid \forall n \in [t-1] : h(x_n) = y_n\}$ be the *version space* at round t. The idea is to predict each

$\hat{y}_t$ using the majority vote $\mathbb{1}_{\{\bar{h}(x_t)\geq 0\}}$, where $\bar{h} = \sum_{h\in \mathrm{VS}_t} h(x_t)$. That way, on any mistake, half of the hypotheses in the version space can be discarded. If $\mathcal{H}$ is finite, we can bound the number of Halving's mistakes $M_{1/2}(\mathcal{H})$ as follows.

Proposition 2 (Angluin [1] and Littlestone [28]). *For any hypothesis space $\mathcal{H}$ it holds that*

$$\mathrm{VC}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H}) \leq M_{1/2}(\mathcal{H}) \leq \log|\mathcal{H}| \leq 2\,\mathrm{VC}(\mathcal{H})\log|X|\,.$$

The last inequality follows from the Sauer-Shela lemma. Note that Halving achieves the optimal mistake bound $\mathrm{Ldim}(\mathcal{H})$ up to the $\log|X|$ factor. For the set of halfspaces $\mathcal{C}_H$ of a graph convexity space $(V,\mathcal{C})$, it holds that [37]

$$\mathrm{VC}(\mathcal{C}_H) \leq r(\mathcal{C}) \leq 2\,\mathrm{tw}(G)+1 \leq 3\,\mathrm{tw}(G)\,, \tag{1}$$

and thus we additionally get the follwing proposition.

Proposition 3. *For the set of halfspaces $\mathcal{C}_H$ of any convexity space $(X,\mathcal{C})$ it holds that*

$$M_{1/2}(\mathcal{C}_H) \leq 2\,\mathrm{VC}(\mathcal{C}_H)\log|V| \leq 2r(\mathcal{C})\log|V| \leq 6\,\mathrm{tw}(G)\log|V|\,.$$

While Halving achieves a near-optimal mistake bound, it is unclear whether it is possible to run Halving in polynomial time. In particular, checking whether there exists any consistent geodesically convex halfspace $h\in\mathcal{C}_H$ for the given partially labelled graph is NP-hard [35]. It is well-known that for many hypothesis spaces on graphs, it is hard to compute Halving's predictions, that is, decide whether $\bar{h}\geq 0$ [8,21]. This makes an exact polynomial time implementation of Halving unlikely. However, in the discussion section we will mention possible directions based on sampling to potentially overcome this problem. If the VC dimension of the set of halfspaces is bounded, we can run Halving in polynomial time by enumerating the version space, as long as we can efficiently compute convex hulls.

Theorem 4. *For any finite S_4 convexity space $(X,\mathcal{C})$, Halving on $\mathcal{C}_H$ can be implemented in time $\mathcal{O}(|X|^{\mathrm{VC}(\mathcal{C}_H)+2}\sigma_T)$ per step, where σ_T is the time complexity to compute convex hulls in $(X,\mathcal{C})$.*

Proof. By the Sauer-Shelah lemma, the hypothesis space and hence any version space has size $\mathcal{O}(|X|^{\mathrm{VC}(\mathcal{C}_H)})$. For any given partially labelled S_4 convexity space, the question whether there exists a consistent hypothesis reduces to the question whether the convex hulls of the positively and negatively labelled points overlap. This follows directly from the definition of S_4 spaces, as in this case, we can find a halfspaces separating the two convex hulls.

A naive enumeration of the version space would result in $2^{|X|}$ such checks. To achieve an enumeration in time $\mathcal{O}(|X|^{\mathrm{VC}(\mathcal{C}_H)+2}\sigma_T)$ we have to be more careful. We first compute the *region of disagreement* D, which consists of all $x\in X$ such that there exist consistent $h,h'\in\mathcal{C}_H$ with $h(x)\neq h'(x)$ in $\mathcal{O}(|X|\sigma_T)$ time.

We will perform the following recursive enumeration. Let $x \in D$. We know that there exists hypotheses h, h' consistent with the remaining labelled points and $h(x) \neq h'(x)$. So, we branch on x by setting its label either to 1 or to -1. Recursively we recompute the disagreement region for this new set of points and continue. Any leaf in this recursion tree corresponds to a unique hypothesis consistent with the original labelled points. Also, as D shrinks in each branching step by at least one element, the path from root to leaf in the recursion tree has length at most $|X|$. In total this gives $\mathcal{O}\left(|X||X|^{\mathrm{VC}(\mathcal{C}_H)}\right)$ many recursion steps, as the number of leaves corresponding to unique hypotheses is bounded by the size of the hypothesis space. As each branching step takes $\mathcal{O}(|X|\sigma_T)$ we achieve the stated overall runtime. □

As in the geodesic convexity, convex hulls can be computed in time $\sigma_T = \mathcal{O}(|V(G)|^3)$ [37], we achieve a polynomial runtime for S_4 graphs of bounded treewidth, because $\mathrm{VC}(\mathcal{C}_H) \leq 3\,\mathrm{tw}(G)$ by Equation (1). For non S_4 graphs we can still achieve polynomial time if we can enumerate the version space in polynomial time. For example, we can enumerate all consistent hypotheses of bipartite or planar graphs, which are in general not S_4 and have unbounded treewidth, in polynomial time [19] leading to the next result.

Proposition 5. *Let G be a planar or bipartite graph and $(V(G), \mathcal{C})$ the geodesic convexity on G. Halving can be implemented in polynomial time for the hypothesis space of geodesically convex halfspaces $\mathcal{C}_H$ on G.*

3.2 Shortest path cover based approach

In contrast to the inefficient Halving algorithm, we discuss now a simple and efficient algorithm achieving optimal bounds only on specific graph families.

To derive a simple upper bound, we note that one immediate consequence of the halfspace assumption is that any shortest path P can have at most one *cut edge*, that is, an edge with differently labelled endpoints. We can follow Gärtner and Garriga [18] to perform online binary search on a path P if we already know the labels of its endpoints. In this case, we can predict for any $x_t \in V(P)$ the label of the closer endpoint. That way if we make a mistake we can deduce at least half of the path's labels. That means that we will make at most $\lceil \log d(G) \rceil$ mistakes, as the length of P is at most the diameter $|V(P)| - 1 \leq d(G)$. Here, log is the base 2 logarithm. If we do not know the endpoints' labels, we can apply the following simple strategy. First we make at most one mistake on the first point $a \in V(P)$. Then we will predict on any $b \in V(P)$ the same label as a, so that on mistake we would have two different labelled points on P. By the halfspace assumption we can infer the labels of all vertices but the a-b sub-path and hence we are back in the previous case with endpoints with known labels.

Lemma 6. *Given any shortest path P in a graph G, there exists a prediction strategy making at most $2 + \log(|V(P)| - 1)$ mistakes on P.*

We can generalise this approach to the whole graph using *shortest path covers* [37], which is a set $\mathcal{S}$ of shortest paths whose vertices cover the graph:

$\bigcup_{P \in \mathcal{S}} V(P) = V(G)$. Performing binary search on each path in $\mathcal{S}$ gives our next mistake upper bound. We call this approach the SPC algorithm SPC($\mathcal{S}$) based on a shortest path cover $\mathcal{S}$.

Theorem 7. *Let $(V, \mathcal{C})$ be the geodesic convexity space on a graph $G = (V, E)$ and $\mathcal{S}$ a shortest path cover of G. The mistake bound for the* SPC *algorithm using a shortest path cover $\mathcal{S}$ is*

$$M_{\mathrm{SPC}(\mathcal{S})}(\mathcal{C}_H) \leq |\mathcal{S}|(2 + \log d(G)) .$$

As we can compute an $\mathcal{O}(\log d(G))$-approximation $\mathcal{S}$ to the minimum shortest path cover $\mathcal{S}^*$ in polynomial time [37] we get the following result.

Theorem 8. *Let $(V, \mathcal{C})$ be the geodesic convexity space on a graph $G = (V, E)$ and $\mathcal{S}^*$ a minimum shortst path cover of G. There exists a polynomial-time online learning algorithm, which computes a shortest path cover $\mathcal{S}$ such that*

$$M_{\mathrm{SPC}(\mathcal{S})}(\mathcal{C}_H) \leq \mathcal{O}(|\mathcal{S}^*|(\log d(G))^2) .$$

There exist edge-weighted graph families where the bound of Theorem 7 is asymptotically tight. For example, we can use the same construction as in [37].

Proposition 9. *There exists a family of edge-weighted graphs $G_{k,\ell}$ with $k, \ell \in \mathbb{N}$, such that $G_{k,\ell}$ has a shortest path cover of size k and diameter ℓ and for any online algorithm A applied to the geodesically convex halfspaces $\mathcal{C}_H$ in $G_{k,\ell}$ the mistake lower bound $M_A(\mathcal{C}_H) \geq k \log \ell$ holds.*

3.3 Efficient algorithms for graphs with bounded cbm(G)

While the previously discussed SPC algorithm is tight on specific graphs, there exists graphs, where it is arbitrarily bad. For example, on the star graph, which is a tree T with $V(T) - 1$ leaves, the minimum shortest path cover $\mathcal{S}^*$ has size $|\mathcal{S}^*| \geq \frac{V(T)-1}{2}$ resulting in a mistake bound linear in $V(T)$, whereas the optimal strategy makes at most two mistakes. In this section, we mitigate the weakness of the SPC algorithm and achieve a polynomial time algorithm with a near-optimal mistake bound on graphs with bounded cbm(G).

Algorithm 1: Tree-based online halfspace learning on graphs

Input: unweighted graph G, with $n = |V(G)|$

compute a Dijkstra shortest path tree T rooted at x_1
predict $\hat{y_1} = 1$ and receive a mistake if $\hat{y_1} \neq y_1$
for $t \in [n] \setminus \{1\}$ **do**
    Let P be the root-leaf path in T containing x_t
    Predict $\hat{y_t}$ with binary search on P.

Theorem 10. *Let G be an unweighted graph. Algorithm 1 has a $\mathcal{O}(|V|^2)$ per step runtime and achieves a mistake bound of $\mathcal{O}(\mathrm{cbm}(G)^2 \log d(G))$ for the class of geodesically convex halfspaces in G.*

Proof. Without loss of generality we can assume that x_1 is positive. Correctness follows immediately by the fact that we are only applying the binary search strategy to each root-leaf shortest path in T.

By Lemma 6 we will make at most $\mathcal{O}(\log d(G))$ mistakes on each such path. Note however, that we only make mistakes on such a path if its leaf in T is negative. Otherwise we will just predict positive all the time and do not make any mistake. We will bound the number of possible leaves in T that can be negative by $\mathcal{O}(\mathrm{cbm}(G)^2)$ and hence show that only $\mathcal{O}(\mathrm{cbm}(G)^2)$ binary searches are required, while on all other paths we will not make any mistakes at all.

Let $H_- \subseteq \mathcal{C}_H$ be any negatively labelled geodesically convex halfspace with $x_1 \notin H_-$ and let $R_+ \subseteq T \setminus H_-$ be the set of vertices that are positive and have a neighbour in H_-. As H_- is convex, and hence connected, we can contract all edges in H_- such that only a single vertex h_- remains. Additionally, we contract all edges leading to the R_+ vertices from x_1 but the ones with endpoints in R_+. This constructions shows that G has a complete bipartite minor $K_{2,|R_+|}$ with x_1 and h_- on the one side and R_+ on the other. Hence, $|R_+| \leq \mathrm{cbm}(G)$.

We inspect again the original non-contracted G. Let $r \in R_+$ and let $L_- \subseteq H_-$ be the children of r in T. As r is positive and all vertices in L_- are negative, L_- must be a clique in G. Otherwise, the halfspace assumption would be violated as the shortest path between $a, b \in L_-$ would go over r. For that, we use the fact that G is unweighted. Any clique of size $f \geq 3$ contains a $K_{2,f-2}$ and hence $f \leq \mathrm{cbm}(G) + 2$. All together this gives $\mathcal{O}(\mathrm{cbm}(G)^2)$ cut edges on T. Note that there might be more cut edges in G, but the halfspaces is determined by the cut edges in T. The runtime is given by one run of Dijkstra and the repeated binary searches on each path. □

Note that for many graphs, the tree-based approach gives a significantly better bound than the shortest path cover based algorithm. On any tree the SPC mistake bound will be linear in the number of leaves, while the tree-based approach will just perform one binary search resulting in $\mathcal{O}(\log d(G))$. Also, for *outerplanar* graphs a constant number of binary searches suffice, as $\mathrm{cbm}(G) \leq 2$ [13].

3.4 Lower bounds

We will use the separation axioms to discuss general mistake lower bounds for arbitrary graph convexity spaces. Let $(V, \mathcal{C})$ be a graph convexity space. Without any further assumptions, we have the VC dimension $\mathrm{VC}(\mathcal{C}_H)$ as a lower bound on the optimal number of mistakes $\mathrm{Ldim}(\mathcal{C}_H)$ as already discussed. For S_4 convexity spaces we have $r(\mathcal{C}) - 1 = \mathrm{VC}(\mathcal{C}_H)$ [37] and hence in this case, also the Radon number is a lower bound on the optimal number of mistakes, $r(\mathcal{C}) - 1 \leq \mathrm{Ldim}(\mathcal{C})$. Interestingly the minimum hull set size $h(G)$ is not a lower bound in general even though it is a lower bound in the active setting for S_3 convexity spaces [37]. For

example, the star graph T has $h(T) = |V(T)| - 1$ but $\mathrm{Ldim}(\mathcal{C}_H) = 1$. Finally, in geodesic S_2 graph convexity spaces, we can place a cut edge arbitrarily on any shortest path, hence $\mathrm{Ldim}(\mathcal{C}_H) \geq \log d(G)$. Compared to specific worst-case graphs, as in Proposition 9, these lower bounds hold in general for any graph, as long as the graph convexity space satisfies the corresponding separation axiom.

4 Learning general convex sets

Having discussed three different approaches to learn halfspaces on graphs, we now turn to general convex sets. We discuss Halving and an adapted shortest path cover based approach for this setting. In the special case of k-intersections of halfspaces, we discuss an adaptive strategy that does not require to know k. Let us start with a standard result on intersection-closed hypothesis spaces, adapted to our graph setting.

Proposition 11 (Horváth and Turán [24]). *For any graph convexity space $(V(G), \mathcal{C})$ on a graph G it holds that*

$$\mathrm{VC}(\mathcal{C}) = \max_{C \in \mathcal{C}} h(G[C])\,.$$

This immediately shows that the minimum hull set size $h(G)$ is a lower bound on $\mathrm{VC}(\mathcal{C})$ and hence also on $\mathrm{Ldim}(\mathcal{C})$. Also, $r(\mathcal{C}) - 1$ is a lower bound on $\mathrm{VC}(\mathcal{C})$, as any set without a Radon partition can be shattered. It is unclear how to compute $\max_{C \in \mathcal{C}} h(G[C])$ efficiently, as already computing $h(G)$ is APX-hard [11]. We provide an efficiently computable upper bound. The VC dimension $\mathrm{VC}(\mathcal{C})$ of any convexity space on a graph can be bounded by the VC dimension of the set of all connected sets $\mathcal{H}_{\mathrm{con}} \supseteq \mathcal{C}$ of G. The quantity $\mathrm{VC}(\mathcal{H}_{\mathrm{con}})$ is bounded by the maximum number of leaves $\ell(G)$ in any spanning tree of G, $\ell(G) \leq \mathrm{VC}(\mathcal{H}_{\mathrm{con}}) \leq \ell(G) + 1$ [27]. Hence, we achieve the following proposition.

Proposition 12. *For any graph convexity space $(V(G), \mathcal{C})$ on a graph G it holds that*

$$\mathrm{VC}(\mathcal{C}) \leq \mathrm{VC}(\mathcal{H}_{con}) \leq \ell(G) + 1\,.$$

Computing $\ell(G)$ is also APX-hard, yet, a near-linear time 3-approximation algorithm exists [30]. The first inequality of Proposition 12 is tight for specific convexity spaces: We can take a maximum vertex set $A \subseteq V(G)$ shatterable by connected sets and define $\mathcal{C}_A = 2^A \cup V(G)$. The resulting convexity space $(V(G), \mathcal{C}_A)$ satisfies $\ell(G) \leq \mathrm{VC}(\mathcal{C}_A) = \mathrm{VC}(\mathcal{H}_{\mathrm{con}}) \leq \ell(G) + 1$.

By Proposition 2 we directly get that Halving achieves the mistake bound $\mathcal{O}(\mathrm{VC}(\mathcal{C}) \log |V(G)|) = \mathcal{O}(\ell(G) \log |V(G)|)$. The next theorem shows that we can run Halving in polynomial-time if the VC dimension is a constant and convex hull computations are efficiently possible in $(X, \mathcal{C})$.

Theorem 13. *For any finite convexity space $(X, \mathcal{C})$ Halving can be implemented in time $\mathcal{O}(|X|^{\mathrm{VC}(\mathcal{C})+1} \sigma_T)$ per step where σ_T is the time complexity to compute convex hulls on $(X, \mathcal{C})$.*

To achieve the runtime in Theorem 13, we can use the algorithm of Boley et al. [5] to enumerate the whole version space in each step. Interestingly, we can adapt our SPC algorithm to be able to handle convex sets with the same asymptotic mistake bound.

Theorem 14. *Let $(V, \mathcal{C})$ be the geodesic convexity space on a graph $G = (V, E)$ and $\mathcal{S}$ a shortst path cover of G. The mistake bound of the* SPC *algorithm using an* SPC $\mathcal{S}$ *is*

$$M_{\mathrm{SPC}(\mathcal{S})}(\mathcal{C}) \leq \mathcal{O}(|\mathcal{S}| \log d(G)) \,.$$

One can achieve the bound by performing two instead of one binary searches as soon as one point is known on any particular path. The idea is based on the same strategy on path covers [18]. We additionally remark that as on any fixed shortest path we cannot shatter three points, we get an upper bound on the VC dimension based on shortest path covers.

Proposition 15. *The VC dimension of geodesically convex sets in a graph is upper bounded by the size of the minimum shortest path cover $2|\mathcal{S}^*|$.*

Note that $|\mathcal{S}^*| \leq \ell(G)$ as any (Dijkstra-based) shortest-path tree with k leaves is a spanning tree and can be covered with k shortest paths.

4.1 Learning k-intersections of halfspaces

In Euclidean space, any convex set can be represented as an intersection of a set of halfspaces. As general convex sets in Euclidean space have infinite VC dimension [25], a common way to bound the complexity is to only look at convex sets that can be represented as the intersection of k halfspaces. The parameter k linearly determines the VC dimension [25].

In general, not all convexity spaces have the property that all convex sets are intersections of halfspaces. Take for example the geodesically convex halfspaces in the complete bipartite graph $K_{2,3}$. The graph only has the two halfspaces $(\emptyset, V(K_{2,3}))$, while each vertex and edge on its own is convex. This property is actually exactly captured by the S_3 separation axiom.

Proposition 16 (van de Vel [38]). *Convex sets in a convexity space $(X, \mathcal{C})$ can be represented as an intersection of a set of halfspaces if and only if $(X, \mathcal{C})$ is an S_3 convexity space.*

For any hypothesis space $\mathcal{H}$ define $\mathcal{H}^{k\cap} = \{h_1 \cap \cdots \cap h_k \mid h_1, \ldots, h_k \in \mathcal{H}\}$ for $k \in \mathbb{N}$. Let $(X, \mathcal{C})$ be a convexity space and denote its halfspaces as $\mathcal{C}_H$. As $X \in \mathcal{C}_H$, we have $\mathcal{C}_H^{k'\cap} \subseteq \mathcal{C}_H^{k\cap}$ for all $k' \leq k$. Note that, Proposition 16 implies that for finite S_3 convexity spaces there is some $k \in \mathbb{N}$ such that $\mathcal{C} = \mathcal{C}_H^{k\cap}$. For any hypothesis space $\mathcal{H}$, the VC dimension of $\mathcal{H}^{k\cap}$ is bounded by $\mathcal{O}(k \log k \,\mathrm{VC}(\mathcal{H}))$ [12]. Additionally for finite X we can again use the Sauer-Shelah Lemma and get

$$\mathrm{VC}(\mathcal{H}^{k\cap}) \leq \log |\, \mathcal{H}^{k\cap} \,| = k \log(|\, \mathcal{H} \,|) = \mathcal{O}(k \log(|X|^{\mathrm{VC}(\mathcal{H})})) = \mathcal{O}(k \,\mathrm{VC}(\mathcal{H}) \log |X|) \,.$$

Hence, applying Halving results in the following bound.

Proposition 17.

$$M_{1/2}\left(\mathcal{C}_H^{k\cap}\right) = \mathcal{O}(k\,\mathrm{VC}(\mathcal{C}_H)\log\min\{|X|,k\}\log|X|)$$

Applying this to k-intersections of halfspaces in graphs gives:

Theorem 18. *Let $G=(V,E)$ be a graph and $(V,\mathcal{C})$ a graph convexity space on G. Halving achieves the following mistake bound:*

$$M_{1/2}\left(\mathcal{C}_H^{k\cap}\right) = \mathcal{O}(k\,\mathrm{tw}(G)\log\min\{|X|,k\}\log|X|)\,.$$

On S_3 graphs we additionally get:

Corollary 19. *Let $G=(V,E)$ be a graph and $(V,\mathcal{C})$ an S_3 graph convexity space on G. Let $k\in\mathbb{N}$ such that $\mathcal{C}=\mathcal{C}_H^{k\cap}$. Halving achieves the following mistake bound:*

$$M_{1/2}(\mathcal{C}) = \mathcal{O}(k\,\mathrm{tw}(G)\log\min\{|X|,k\}\log|X|)\,.$$

The two previous bounds for Halving are difficult to use as to the best of our knowledge there is no obvious way to compute or upper bound k for a given S_3 graph convexity space. Also, the minimum k required to achieve $\mathcal{C}=\mathcal{C}_H^{k\cap}$ is non-trivial to compute. In this last paragraph of the section, we discuss how to make Halving *adaptive*, in the sense that if the target hypothesis C^* is in $\mathcal{C}_H^{k'\cap}$ for some $k'\in\mathbb{N}$ which we do not know, we still get a bound linear in k' instead of the globally required k. This can be achieved using the standard *doubling trick* [10]. It works by assuming that k belongs to $\{2^{i-1},\dots,2^i\}$ for $i\in[\lceil\log k\rceil]$ and iteratively applying Halving to the hypothesis space $\mathcal{H}_{2^i}$. Each time the whole hypothesis space $\mathcal{H}_{2^i}$ is not consistent anymore with the labels seen so far, the i is increased by one. We call this approach ADA-$^1/_2$, for *adaptive Halving*, and it achieves the following mistake bound.

Proposition 20. *Let $\mathcal{H}$ be a hypothesis space such that $\emptyset=\mathcal{H}_0\subseteq\mathcal{H}_1\subseteq\mathcal{H}_2\subseteq\cdots\subseteq\mathcal{H}$. ADA-$^1/_2$ achieves the following bound if the target hypothesis H is in $\mathcal{H}_k$ for some unknown $k\in\mathbb{N}$:*

$$M_{\text{ADA-}^1/_2}(\mathcal{H}) \leq \sum_{i\in[\lceil\log k\rceil]} \log|\,\mathcal{H}_{2^i}\setminus\mathcal{H}_{2^{i-1}}\,|\,.$$

Applied to k-intersections of halfspaces in graphs we achieve:

Corollary 21.

$$M_{\text{ADA-}^1/_2}(\mathcal{H}^{k\cap}) = \mathcal{O}(k\,\mathrm{tw}(G)\log\min\{|X|,k\}\log|X|)\,.$$

The additional constant factor to achieve the adaptive variant is negligible compared to standard Halving on the set of all convex sets $\mathcal{C}$, where the required number of halfspaces k could be even linear in $|X|$, for example, on a star graph.

5 Discussion

We discuss efficiency aspects of Halving and how to generalise it to the agnostic case. We compare our results to previous bounds in online learning on graphs and show how we can use the *closure algorithm* in our setting.

Efficient Halving by sampling All discussed Halving based algorithms in this paper are in general not efficient. In particular, computing the weighted majority vote is in many cases hard. One possible way around this issue is to use the randomised version of Halving. It samples a consistent hypothesis uniformly at random and uses it for prediction of the current point x_t. This simple strategy RAND-$^1/_2$ is already enough to achieve the Halving bound in expectation.

Proposition 22 (Littlestone and Warmuth [29]).

$$\mathbb{E}[M_{\text{RAND-}^1/_2}(\mathcal{H})] \leq \ln |\, \mathcal{H} \,| \,.$$

Thus, if we can sample uniformly at random from the version space we achieve Halving's bound in expectation. As a simple example, let us compare Halving and Rand-Halving on the simple learning problem of halfspaces on a path P. Standard Halving enumerates the whole version space of size $\mathcal{O}(|V(P)|)$, while Rand-Halving only needs to sample a number in $[|V(P)|]$, which can be achieved with $\mathcal{O}(\log |V(P)|)$ random binary draws; an exponential increase. However in general, sampling uniformly at random from a version space is a non-trivial task. Boley et al. [4] and Ganter [17] discuss sampling general convex sets in the context of *frequent pattern mining* and *formal concept analysis*. Nevertheless, their results are also applicable in our context. [4] shows that in general it is hard to sample a convex set uniformly at random, which corresponds to sampling a consistent convex hypothesis. Under additional assumptions they construct a Markov chain with polynomial mixing time and also discuss various practically efficient heuristics. Applying these techniques in our context is future work. A potential way to overcome the hardness might be to approximate sample, that is, only close to uniform, which still would provide an $\mathcal{O}(\log |\, \mathcal{H} \,|)$ bound.

Agnostic online learning In the *agnostic* version of the problem, we drop the realisability assumption, hence the opponent is allowed to use arbitrary labels y_t for $t \in [T]$ for some $T \in \mathbb{N}$. In this more general online learning model it is essentially hopeless to bound the number of mistakes, as the opponent can always set $y_t \neq \hat{y}_t$. Because of that, typically the *regret* is studied instead in agnostic online learning. The regret for any particular sequence $x_1, x_2, \ldots$ of an randomised algorithm A with predictions $A_t(x_t) = \hat{y}_t$ is

$$R_A(\mathcal{H}) = \mathbb{E}\left[\max_{x_1,\ldots,x_T} \sum_{t\in[T]} \mathbb{1}_{\{\hat{y}_t \neq y_t\}} - \min_{h\in\mathcal{H}} \sum_{t\in[T]} \mathbb{1}_{\{h(x_t)\neq y_t\}} \right],$$

where the expectation is taken over the random predictions of the algorithm A. Hence, we compare the performance of algorithm A with the best fixed hypothesis from a given hypothesis space $\mathcal{H}$ in hind-sight.

Ben-David and Pál [2] have proven that there exists an algorithm A achieving the optimal regret $R_A(\mathcal{H}) \leq \sqrt{1/2 \ln(|\mathcal{H}|)T}$. This means that if we let $\{x_1, \ldots, x_T\} = V$ for some graph $G = (V, E)$ we get the regret bound $R_A(\mathcal{H}) \leq \sqrt{1/2 \ln(|\mathcal{H}|)|V|}$ if each vertex x_i appears only once. Note that in the realisable case we achieved bounds $\mathcal{O}(\log |V|)$ (not considering other parameters), while here in the agnostic case we get $\mathcal{O}(\sqrt{|V|})$. If we can expect that there is some hypothesis in $\mathcal{H}$ that performs rather well, say

$$\min_{h \in \mathcal{H}} \sum_{t \in [T]} \mathbb{1}_{\{h(x_t) \neq y_t\}} \leq M^\star$$

for some known $M^\star \in \mathbb{N}$, we can significantly improve the bound to

$$R_A(\mathcal{H}) \leq \sqrt{2M^\star \operatorname{Ldim}(\mathcal{H})} + \operatorname{Ldim}(\mathcal{H}) .$$

For small $M^\star$ this asymptotically matches the realisable bound. By using the standard *doubling trick* [10] we can achieve almost the same bound without knowing $M^\star$. This bound allows learning in the following special case. Assume the target hypothesis h^* is a vertex set that is a positive convex set but with at most $M^\star$ labels flipped to negative. That is, there exists $h \in \mathcal{C}$ which predicts everywhere positive where h^* predicts positive, but can additionally predict at up to $M^\star$ many points positive, where h^* is negative. Let $\mathcal{C}_{M^\star}$ be this space of *noisy* convex sets containing each set $C \in \mathcal{C}$ with all possible at most $M^\star$ label flips. That is $|\mathcal{C}_{M^\star}| = \mathcal{O}(|\mathcal{C}||X|^{M^\star})$. Applying adaptive Halving achieves a mistake bound similar to the the regret-based analysis:

$$M_{\text{ADA -}1/2}(\mathcal{C}_{M^\star}) = \mathcal{O}((\operatorname{VC}(\mathcal{C}) + M^\star) \log |X|) .$$

5.1 Comparison to cut-based learning

Common bounds in online learning on graphs do not make any hypothesis-space-based assumption and instead depend on the *cut-size* $\Phi_G(y) = \sum_{v,w \in E(G)} \mathbb{1}_{\{y(v) \neq y(w)\}}$ of the labelling y. Let $\tilde{\mathcal{H}}_c$ be the hypothesis space of labellings with bounded cut-size $\Phi_G(y) \leq c$. The VC dimension of $\tilde{\mathcal{H}}_c$ can be bounded linearly by c, $\operatorname{VC}(\tilde{\mathcal{H}}_c) \leq 2c + 1$ [26]. By applying Halving to this hypothesis space we get the bound $\mathcal{O}(c \log |V|)$ [22]. Again, we can use the doubling trick to get the same bound without knowing the correct value of c.

Proposition 23.

$$M_{\text{ADA -}1/2}(\tilde{\mathcal{H}}_c) = \mathcal{O}(c \log |V|) .$$

Herbster et al. [21] proved that the majority vote in $\tilde{\mathcal{H}}_c$ is NP-hard, based on the fact counting label-consistent min-cuts is #P-hard. This makes the existance of an efficient and exact Halving algorithm for $\tilde{\mathcal{H}}_c$ unlikely.

Comparing with our bounds, we see that the cut-size c now has the same role as previously $r(G)$ or $\mathrm{tw}(G)$ in our bounds. Note however, that our bounds are label-independent, that is, under the halfspace or convexity assumption they hold for any labelling. Cut-size based bounds are complementary and depend on the actual labelling.

The problem of online learning on graphs was introduced by Herbster et al. [23]. They bound the number of mistakes as $4\Phi_G(y)d(G)\,\mathrm{bal}(y)$, where $\mathrm{bal}(y) = (1 - 1/|V(G)||\sum y_i|)^{-2}$ is a *balancedness* term. The efficient Pounce algorithm [20] achievesthe mistake bound $\mathcal{O}(\Phi_G(y)(\log|V(G)|)^4)$ for unweighted graphs, almost matching the near-optimal bound of Halving. In parallel to these works, Cesa-Bianchi et al. [7] first developed an efficient and optimal algorithm for online learning on trees and showed that Halving on trees actually also asymptotically achieves the optimal bound, which can be much smaller than $\mathcal{O}(c\log|V(G)|)$. The authors then generalised these ideas to general graphs [9] and achieved under mild assumptions an efficient algorithm that is optimal up to a $\log|V(G)|$ factor. We refer to [9] and [21] for an overview and in-depth discussion.

The convexity or the halfspace assumption are orthogonal to the standard assumption of small cut-size. For example, on a $2 \times k$ grid, we can have halfspaces corresponding to the two $1 \times k$ halves, that have a cut of size k. That means that the convexity or halfspace assumption can lead to strong bound in situations where the cut of the labelling might be large. However, assuming a small cut can also improve our bounds. For example, the shortest path cover based bound can be changed to $\mathcal{O}(\min\{|\mathcal{S}|, \Phi_G(y)\}\log d(G))$, as we only have to do at most $\min\{|\mathcal{S}|, \Phi_G(y)\}$ binary searches.

6 Conclusion

In this paper, we have studied online learning of halfspaces and general convex sets on graphs. On the one hand, we discussed that Littlestone's Halving algorithm achieves near-optimal bounds in general convexity spaces, yet is inefficient in general in its standard form. On the other hand, we have used shortest path covers to achieve a simple and efficient algorithm, which is however not optimal in many cases. For the special case of geodesic halfspaces on graphs with bounded $\mathrm{cbm}(G)$, we proposed an algorithm with near-optimal mistake bound and quadratic runtime. We have discussed general lower bounds and specific worst-case examples. In the case of halfspaces we argued that general, increasingly stronger lower bounds are achieved through the separation axioms $S_1, \ldots, S_4$. We looked at the special case of k-intersections of halfspaces and discussed an adaptive version of Halving using the well-known doubling trick. Finally, we compared our bounds to previous label-dependent mistake bounds and discussed potential extensions to the agnostic case. As future work, we are looking into more general efficient and near-optimal algorithms, more relaxed assumptions on the labels, and multi-class online learning on graphs.

References

1. Angluin, D.: Queries and concept learning. Machine learning **2**(4), 319–342 (1988)
2. Ben-David, S., Pál, D., Shalev-Shwartz, S.: Agnostic online learning. In: COLT (2009)
3. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on computing **25**(6), 1305–1317 (1996)
4. Boley, M., Gärtner, T., Grosskreutz, H.: Formal concept sampling for counting and threshold-free local pattern mining. In: ICDM (2010)
5. Boley, M., Horváth, T., Poigné, A., Wrobel, S.: Listing closed sets of strongly accessible set systems with applications to data mining. Theoretical Computer Science **411**(3), 691–700 (2010)
6. Bressan, M., Cesa-Bianchi, N., Lattanzi, S., Paudice, A.: Exact recovery of clusters in finite metric spaces using oracle queries. In: COLT (2021)
7. Cesa-Bianchi, N., Gentile, C., Vitale, F.: Fast and optimal prediction on a labeled tree. In: COLT (2009)
8. Cesa-Bianchi, N., Gentile, C., Vitale, F., Zappella, G.: A correlation clustering approach to link classification in signed networks. In: COLT (2012)
9. Cesa-Bianchi, N., Gentile, C., Vitale, F., Zappella, G.: Random spanning trees and the prediction of weighted graphs. JMLR **14**(1), 1251–1284 (2013)
10. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games (2006)
11. Coelho, E.M.M., Dourado, M.C., Sampaio, R.M.: Inapproximability results for graph convexity parameters. Theoretical Computer Science **600**, 49–58 (2015)
12. Csikós, M., Mustafa, N.H., Kupavskii, A.: Tight lower bounds on the VC-dimension of geometric set systems. JMLR **20**(1), 2991–2998 (2019)
13. Diestel, R.: Graph Theory. Springer, 5th edn. (2017)
14. Duchet, P.: Convex sets in graphs, II. Minimal path convexity. Journal of Combinatorial Theory, Series B **44**(3), 307–316 (1988)
15. Farber, M., Jamison, R.E.: Convexity in Graphs and Hypergraphs. SIAM Journal on Algebraic Discrete Methods **7**(3), 433–444 (1986)
16. Frances, M., Litman, A.: Optimal mistake bound learning is hard. Information and Computation **144**(1), 66–82 (1998)
17. Ganter, B.: Random extents and random closure systems. In: CLA. vol. 959, pp. 309–318 (2011)
18. Gärtner, T., Garriga, G.C.: The cost of learning directed cuts. In: ECMLPKDD (2007)
19. Glantz, R., Meyerhenke, H.: On finding convex cuts in general, bipartite and plane graphs. Theoretical Computer Science **695**, 54–73 (2017)
20. Herbster, M., Lever, G., Pontil, M.: Online prediction on large diameter graphs. NIPS (2008)
21. Herbster, M., Pasteris, S., Ghosh, S.: Online prediction at the limit of zero temperature. NIPS (2015)
22. Herbster, M., Pontil, M.: Prediction on a graph with a perceptron. NIPS (2006)
23. Herbster, M., Pontil, M., Wainer, L.: Online learning over graphs. In: ICML (2005)
24. Horváth, T., Turán, G.: Learning logic programs with structured background knowledge. Artificial Intelligence **128**(1-2), 31–97 (2001)
25. Kearns, M.J., Vazirani, U.: An introduction to computational learning theory (1994)
26. Kleinberg, J.: Detecting a network failure. Internet Mathematics **1**(1), 37–55 (2004)
27. Kranakis, E., Krizanc, D., Ruf, B., Urrutia, J., Woeginger, G.: The vc-dimension of set systems defined by graphs. Discrete Applied Mathematics **77**(3), 237–257 (1997)

28. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine learning **2**(4), 285–318 (1988)
29. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Information and computation **108**(2), 212–261 (1994)
30. Lu, H.I., Ravi, R.: Approximating maximum leaf spanning trees in almost linear time. Journal of algorithms **29**(1), 132–141 (1998)
31. Missura, O., Gärtner, T.: Predicting dynamic difficulty. In: NIPS (2011)
32. Novikoff, A.B.: On convergence proofs on perceptrons. In: Symposium on the Mathematical Theory of Automata (1962)
33. Pelayo, I.M.: Geodesic convexity in graphs. Springer (2013)
34. Rosenblatt, F.: The Perceptron: a probabilistic model for information storage and organization in the brain. Psychological review **65**(6), 386 (1958)
35. Seiffarth, F., Horváth, T., Wrobel, S.: Maximal closed set and half-space separations in finite closure systems. In: ECMLPKDD (2019)
36. Stadtländer, E., Horváth, T., Wrobel, S.: Learning weakly convex sets in metric spaces. In: ECMLPKDD (2021)
37. Thiessen, M., Gärtner, T.: Active learning of convex halfspaces on graphs. In: NeurIPS (2021)
38. van de Vel, M.L.: Theory of convex structures (1993)