

Active Learning on Graphs with Geodesically Convex Classes

Maximilian Thiessen
maximilian.thiessen@tuwien.ac.at
TU Wien
Vienna, Austria

Thomas Gärtner
thomas.gaertner@tuwien.ac.at
TU Wien
Vienna, Austria

ABSTRACT

We study the problem of actively learning the vertex labels of a graph, assuming the classes form geodesically convex subgraphs, which is related to linear separability in the Euclidean setting. The main result of this paper is a novel query-efficient active learning algorithm with label-independent upper bounds on the number of queries needed to learn all labels. For that, we use shortest path covers and provide a logarithmic approximation for the subproblem of computing a shortest path cover of minimum size. We extend the approach to arbitrarily labeled graphs using a convexity-based selection criterion. Finally, we discuss whether the convexity assumption holds on real-world data and give some first preliminary results on citation and image benchmark datasets.

KEYWORDS

active learning, graphs, geodesic convexity, node classification, multi-class, path cover

ACM Reference Format:

Maximilian Thiessen and Thomas Gärtner. 2020. Active Learning on Graphs with Geodesically Convex Classes. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

This work introduces a new, purely combinatorial approach for active node classification on graphs based on the notion of *geodesic graph convexity*. In classical Euclidean settings, learning becomes easy when we assume that the labeled classes form convex regions in the space, i.e. they are linearly separable, which allows building efficient algorithms, giving performance guarantees, and staying practically relevant, see e.g. the work of Dasgupta et al. [2009].

Recently, Seiffarth et al. [2019] and de Araújo et al. [2019] have started to explore convexity assumptions in graph-based semi-supervised settings, aiming to achieve similar benefits as in Euclidean space. Analogous to the Euclidean setting, the authors assume that the classes form (geodesically) convex subgraphs, instead of convex regions in \mathbb{R}^d . Geodesic convexity is a generalization of the regular notion of convexity in Euclidean space. A subgraph is convex if it is closed under taking shortest paths, i.e. any shortest path having endpoints in a convex subgraph does not leave the

subgraph. In many practical settings, like disease spreading and community detection tasks, it seems reasonable to suppose that a shortest path having related endpoints will traverse through a region of similar data points.

We use this assumption to get a provably query-efficient active learning algorithm and turn it into a practical method.

In the following sections, we start by introducing the necessary background in convexity theory and discuss our new active learning algorithm. Then, we turn it into a practically efficient version based on a greedy selection criterion. Finally, we discuss the practical relevance of the convexity assumption and show some preliminary first results on citation and image benchmark datasets.

Related work. The idea of using binary search for querying tasks was often studied before [Afshani et al. 2007; Dasarathy et al. 2015; Emamjomeh-Zadeh et al. 2016; Gärtner and Garriga 2007; Nowak 2009] and is a central part of our approach. Missura and Gärtner [2011] studied active learning on graphs with convex classes for the special case of directed acyclic graphs, whereas we consider general weighted graphs. For an introduction to convexity spaces and geodesic graph convexities, we refer the reader to the works of Kay and Womble [1971], van De Vel [1993], and Pelayo [2013]. Auer and Cesa-Bianchi [1998] were one of the firsts using convexity spaces in the learning context. Shortest path covers, our main algorithmic tool, were studied before in a non-learning setting by Pan and Chang [2006] and Fitzpatrick [1997].

2 QUERYING CONVEX GRAPH PARTITIONS

Let $G = (V, E)$ be a directed or undirected graph with labels $\lambda : V \rightarrow \{0, 1\}$ and possibly weighted edges $w : E \rightarrow \mathbb{R}_{\geq 0}$, where the vertices V correspond to data points. The edges E together with the weights w encode structural relationships between the vertices. The goal is to iteratively query as few vertices as possible, while at the same time predicting the labels of the remaining vertices with small error. We assume a noise-free oracle, thus querying a vertex v always returns the correct label $\lambda(v)$.

The weights w induce a *shortest path distance* on the vertices, corresponding to the weighted length of any shortest path between two vertices. The *diameter* of a graph is the maximum number of edges in any shortest path. To achieve a query-efficient algorithm we will assume, similar to *linear separability* in Euclidean space, that each of the classes corresponds to a non-overlapping convex subgraph of G . This means that the labeled subgraphs form a *convex partition* of the graph. Deciding whether a graph admits such a partition is an interesting problem on its own and NP-hard [Artigas et al. 2011]. Geodesic graph convexities and the classical Euclidean one are indeed special cases of abstract convexity spaces on general set systems [van De Vel 1993].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Intuitively, due to the lack of connecting straight lines as in \mathbb{R}^n , connecting shortest paths are used to define convexity on graphs. To formalize this, let us define the *convex interval* $I(x, y)$ of two vertices x and y using the set of all shortest paths \mathcal{P} in G :

$$I(x, y) = \bigcup_{P \in \mathcal{P}} \{V(P) \mid P \text{ has } x \text{ and } y \text{ as its endpoints}\} \cup \{x, y\},$$

where $V(P)$ denotes the vertex set of the path P . For a subset of vertices $X \subseteq V$, we use the shorthand notation $I(X) = \bigcup_{x, y \in X} I(x, y)$. We say a subgraph $G' = (V', E')$ of G is (*geodesically*) *convex*, if $I(V') = V'$.

The main benefit of assuming the classes to be convex is that it greatly reduces the number of possible labelings of a single shortest path. Either both endpoints of the path have the same label, implying the whole path must have exactly this label, or they are different and the label changes at exactly one edge.

This leads to our main algorithmic result. For that, let us define *shortest path covers*. A set $\mathcal{S} \subseteq \mathcal{P}$ of shortest paths, whose vertices jointly cover all vertices V of the graph, $\bigcup_{P \in \mathcal{S}} V(P) = V$, is a shortest path cover. Having such a cover reduces the problem of inferring the graph's labels to binary search on each path to find the edge where the labels change. This yields an upper bound on the number of needed queries to infer all the labels of the graph. This simple idea can be generalized to the multi-class setting and results in the following theorem.

THEOREM 1. *For any weighted graph $G = (V, E)$ labeled according to a convex r -partition with a shortest path cover \mathcal{S} and diameter d , $O(|\mathcal{S}|r \log d)$ queries are needed to correctly infer all vertex labels of G .*

PROOF. We go through each path in \mathcal{S} one by one. We spend 2 queries to get the labels of the endpoints of the path. If they are equal, we predict all the labels of the path with this label and proceed with the next path.

Otherwise, for $r = 2$, there is exactly one edge where the labels change. We find it with binary search using $O(\log d)$ queries, which yields the correct labels for the whole path.

For $r > 2$, we fix one endpoint and its corresponding class. Due to the convexity assumption, there is one edge partitioning the path into the fixed class and all other classes. Thus, we can treat all remaining classes as a single convex class on the path by itself. We perform the same binary search as in the $r = 2$ case to find the edge where the fixed class ends. We iterate this until all edges with label changes are found with at most $r - 1$ binary searches.

Overall, summing up the queries made on all paths in \mathcal{S} gives us the bound. \square

Given the bound of Theorem 1, we are naturally interested in a shortest path cover of minimum size tightening the bound. The hardness of computing such a cover is an open problem [Manuel 2018] and previously there were no known approximation algorithms. We achieved an $O(\log d)$ -approximation using the classical greedy algorithm for the set-cover problem [Chvatal 1979]. The main idea is to iteratively select a shortest path covering the maximum number of not yet covered vertices. The details are in the appendix.

The strength of this bound is that it does not depend on the labels, but solely on the unlabeled graph structure. This is different from most previous bounds, which for example depend on the size of the number of edges between the classes [Afshani et al. 2007; Blum and Chawla 2001]. Thus, given the convexity assumption holds, this allows to upper bound the number of needed queries before even starting the active learning scheme. Moreover, the approach described in Theorem 1 does not need to know the number r of classes in advance, as the binary search dynamically continues, as long as there could be a new class left.

2.1 Greedy budgeted approach

One drawback of the described theoretical querying scheme is that it queries one path with binary search until the end before switching to the next path. In practical situations a more adaptive and global scheme is preferable.

Indeed, always bisecting the path with the largest region of yet unlabeled vertices and stopping early after m queries gives us an exponentially decreasing upper bound on the prediction error of $O\left(e^{-\frac{m}{|\mathcal{S}|r}}\right)$. This behavior is typical for active learning-based bounds and is one of the main theoretical benefits of active approaches compared to regular passive bounds, usually decreasing with $\Omega\left(\frac{1}{m}\right)$, see for example the work of Dasgupta [2006].

To make the approach more efficient we adapt classical techniques from probabilistic active learning, like selecting the most *informative* point [Settles 2012], to our setting. Thus, we query the vertex v whose label would maximally increase the number of known labels, for example by applying the convex interval $I(\cdot)$. We do not only gain the label $\lambda(v)$ but also all labels in $I(C \cup \{v\}) \setminus I(C)$, where $C \subseteq V$ are the already queried vertices belonging to the same class as v . More generally, having queried the vertices $C_1, \dots, C_r \subseteq V$, belonging to the classes 1 to r respectively, the criterion becomes:

$$\max_{v \in V} \frac{1}{r} \sum_{i=1}^r |I(C_i \cup \{v\}) \setminus I(C_i)|, \quad (1)$$

where ties are resolved uniformly at random.

The problem with such a pure greedy approach is that we would lose our upper bound of Theorem 1. To combine the strengths, of querying in a global and exploitative manner while still keeping the performance guarantee, we perform the maximization in Equation (1) of the greedy criterion not over all vertices in the graph, but only over a certain subset of candidates. In particular, we let each path of a computed shortest path cover iteratively provide a candidate vertex. The candidates are simply mimicking the querying based on the theoretical scheme with binary search: First, select the endpoints of the path as candidates and if their labels differ also the vertices yielded by binary search, i.e. the one bisecting the region of unknown labels.

We propose to use this combined querying scheme also for arbitrarily labeled graphs and justify it empirically in the following section.

3 EXPERIMENTS

We first discuss whether the convexity assumption holds in practical situations. As in the Euclidean setting, real-world data is noisy

	Citeseer	Cora	MNIST	Coil
properties	citation, unweighted	citation, unweighted	image, 5-NN	image, 10-NN
$ V $	2110	2458	1957	1089
$ E $	3757	5209	3981	4913
classes	6	7	10	2
$ S $	667	646	269	245
convex labeled paths in S	535/667 (80%)	540/646 (84%)	223/269 (83%)	205/245 (84%)

Table 1: Overview of the used benchmark citation and image nearest neighbor (NN) datasets. For each dataset, this table contains the number of vertices, edges, classes, and the size of the shortest path cover computed with the discussed approximation method. In the last row, the number of shortest paths labeled according to a convex partition in the shortest path cover is given.

and typically does not follow the convexity assumption completely, which is the equivalent of assuming linearly separable classes. However, it seems reasonable to assume that not all but most of the shortest paths that have endpoints in one class do not leave the subgraph representing the class.

Indeed, our algorithm does not depend on the whole graph to be labeled according to a convex partition. As long as the currently queried path is by itself partitioned into convex subpaths by the classes, the algorithm will not make an error on this particular path.

3.1 Testing the convexity assumption

We evaluate the convexity assumption on four benchmark datasets and report some promising findings in Table 1. We use the citation datasets Citeseer and Cora as described by Ma et al. [2013], a random sample of MNIST [LeCun 1998] with 200 images of each digit and the binary version of the Coil image dataset by Chapelle et al. [2006]. Both citation datasets are unweighted. For the image datasets, we construct the k -nearest-neighbor similarity graph with $k = 5$ for MNIST and $k = 10$ for Coil and assign the Euclidean distance to the edges. Finally, we take the largest connected component of each of these graphs. Further information about the test graphs is summarized in Table 1.

For each graph, we compute a shortest path cover S with the discussed greedy method. The covers were constructed without knowledge of the labels and not fine-tuned. To evaluate the convexity assumption we check for each path in the covers, whether the classes form convex subpaths of the path. On all datasets, more than 80% of the paths fulfill this property. Consequently, our simple algorithm described in Theorem 1 will make perfect predictions on at least 80% of the paths, even though the convexity assumption is not fulfilled on the whole graph.

3.2 Query evaluation

Having discussed the practical relevance of the convexity assumption, we present some first preliminary results regarding query performance.¹

We compare our budgeted greedy approach to passive random sampling. Both approaches query 1 to 50 vertices in the graphs and afterwards use the multi-class extension [Bengio et al. 2006]

¹A Python implementation can be found at <https://github.com/mthiessen/convex-active-graph-learning>

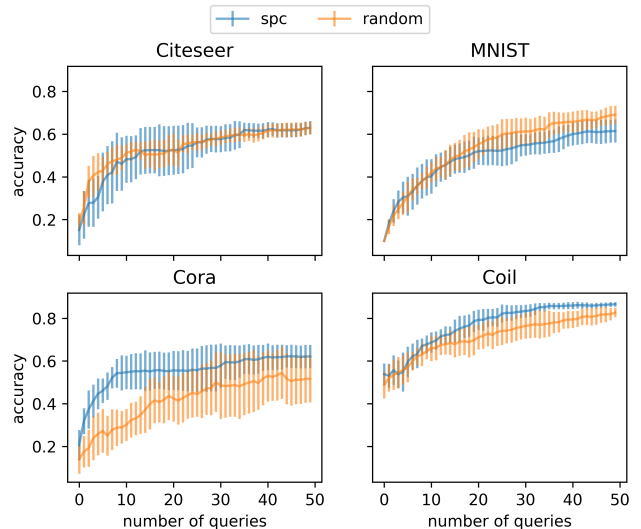


Figure 1: Accuracy vs. number of queries for shortest-path-cover-based querying (spc) and random sampling. The average accuracy over ten runs is shown with error bars representing the standard deviation.

of label propagation [Zhu and Ghahramani 2002] to predict the remaining labels. The accuracy is computed over the whole graph. We repeated each run 10 times. The results are shown in Figure 1.

On the Citeseer dataset, active querying does not improve the accuracy on average compared to random sampling. The variance is however quite high. This suggests that a more carefully chosen selection criterion than Equation (1) could stabilize and improve this behavior. The performance on MNIST is even worse, as random sampling outperforms our approach on average by roughly 5% for the larger number of queries.

We achieve the most promising results on the Cora graph. Our approach outperforms random sampling on average by roughly 25% in the beginning and still by 10% for 30-50 queries. On Coil, our approach also performs about 10% better. Additionally, our approach does not only perform on average better on the latter two but also produces significantly more stable results especially towards the end.

4 CONCLUSION

In this paper, we discussed the theoretical and practical advantages of the geodesic convexity assumption for graphs. We designed a novel query-efficient active learning algorithm and achieved a label-independent query upper bound. Building on this, we developed a practical greedy selection criterion.

We have shown that a local version of the assumption, looking at single shortest paths, is often true on various benchmark datasets. This is the reason for the promising first results, even though the convexity assumption regarding the whole graph, which was used to design our algorithms, is typically not fulfilled in practice.

Still, there are multiple drawbacks to our method. The main one is scalability, as computing convex intervals takes up to $O(|V|^3)$

and shortest path covers $O(|V|^4)$ time. To apply our methods on larger graphs, we will consider fast approximations, for example, based on landmarks [Potamias et al. 2009]. Additionally, our current assumption is too restrictive for real-world datasets. Generalizations to noisy labels and allowing each class to form multiple convex regions in the graph instead of just one will be considered as future work.

A $O(\log d)$ -APPROXIMATION FOR THE MINIMUM SHORTEST PATH COVER PROBLEM

The general set-cover problem asks for a minimum number of sets $B_1, B_2, \dots \in \mathcal{B}$ to cover a ground set A , i.e. $\bigcup_i B_i = A$. Let k be the size of the largest set in \mathcal{B} . The greedy strategy iteratively selects the set in \mathcal{B} covering the largest number of yet uncovered elements in A . It achieves a $(1 + \ln k)$ -approximation for the minimum set cover problem [Chvatal 1979].

To apply it to our setting, where the ground set A is the set of vertices V , the family of subsets \mathcal{B} is the set of shortest paths \mathcal{P} , and k is one larger than the diameter d of the graph, $k = d + 1$, we need to compute a shortest path covering the largest number of not yet covered vertices.

We have to assume that the weights w are strictly positive, because otherwise in the case of all weights w being zero, computing such a shortest path covering a maximum number of vertices is equivalent to computing a Hamiltonian path and thus NP-hard [Garey and Johnson 1979].

It is not possible to simply enumerate all shortest paths and select the one covering the largest number of vertices, as there might be exponentially many shortest paths in the graph.

Still, this can be achieved by modifying the weights $w(a, b)$ for each directed edge (a, b) in the graph to $w^*(e) = (w(e), 0)$ if b is already covered and $(w(e), -1)$ otherwise. We can assume the graph to be directed, because an undirected graph can be transformed into an equivalent directed graph by duplicating each edge $\{x, y\}$ to (x, y) and (y, x) of the same weight. Sorting the weights w^* lexicographically results in the following behavior. Two paths of the same w -weight, but with one path covering a larger number of not yet covered vertices, have different weights according to w^* , where the one covering more vertices is shorter. We can apply the generalized Dijkstra's algorithm of Sobrinho [2001] to compute a w^* -shortest path, which thus gives us a w -shortest path that covers the largest number of not yet covered vertices.

Iteratively computing such w^* -shortest paths results in a shortest path cover \mathcal{S} being at most logarithmically larger than the minimum shortest path cover \mathcal{S}^* : $|\mathcal{S}| \leq (1 + \ln(d + 1))|\mathcal{S}^*|$.

REFERENCES

- Peyman Afshani, Ehsan Chiniforooshan, Reza Dorrigiv, Arash Farzan, Mehdi Mirzazadeh, Narges Simjour, and Hamid Zarrabi-Zadeh. 2007. On the complexity of finding an unknown cut via vertex queries. In *International Computing and Combinatorics Conference*. Springer, 459–469.
- D. Artigas, S. Dantas, M. C. Dourado, and J. L. Szwarcfiter. 2011. Partitioning a graph into convex sets. *Discrete Mathematics* 311, 17 (2011), 1968–1977.
- Peter Auer and Nicolo Cesa-Bianchi. 1998. On-line learning with malicious noise and the closure algorithm. *Annals of mathematics and artificial intelligence* 23, 1-2 (1998), 83–99.
- Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, O. Chapelle, B. Schölkopf, and A. Zien (Eds.). MIT Press, Cambridge, MA, 193–216.
- Avrim Blum and Shuchi Chawla. 2001. Learning from Labeled and Unlabeled Data using Graph Mincuts. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 19–26.
- O. Chapelle, B. Schölkopf, and A. Zien (Eds.). 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.
- Vasek Chvatal. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research* 4, 3 (1979), 233–235.
- Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu. 2015. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In *Conference on Learning Theory*. 503–522.
- Sanjoy Dasgupta. 2006. Coarse sample complexity bounds for active learning. In *Advances in neural information processing systems*. 235–242.
- Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. 2009. Analysis of perceptron-based active learning. *Journal of Machine Learning Research* 10, Feb (2009), 281–299.
- Paulo Henrique Macêdo de Araújo, Manoel Campêlo, Ricardo C Corrêa, and Martine Labbé. 2019. The Geodesic Classification Problem on Graphs. *Electronic Notes in Theoretical Computer Science* 346 (2019), 65–76.
- Ehsan Emamjomeh-Zadeh, David Kempe, and Vikrant Singhal. 2016. Deterministic and probabilistic binary search in graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 519–532.
- Shannon Lesley Fitzpatrick. 1997. *Aspects of domination and dynamic domination*. Ph.D. Dissertation.
- Michael R Garey and David S Johnson. 1979. *Computers and intractability*. Vol. 174. W. H. Freeman & Co.
- Thomas Gärtner and Gemma C Garriga. 2007. The cost of learning directed cuts. In *European Conference on Machine Learning*. Springer, 152–163.
- David Kay and Eugene W Womble. 1971. Axiomatic convexity theory and relationships between the Carathéodory, Helly, and Radon numbers. *Pacific J. Math.* 38, 2 (1971), 471–485.
- Yann LeCun. 1998. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- Yifei Ma, Roman Garnett, and Jeff Schneider. 2013. Σ -optimality for active learning on Gaussian random fields. In *Advances in Neural Information Processing Systems*. 2751–2759.
- Paul Manuel. 2018. Revisiting path-type covering and partitioning problems. *arXiv preprint arXiv:1807.10613* (2018).
- Olana Missura and Thomas Gärtner. 2011. Predicting dynamic difficulty. In *Advances in neural information processing systems*. 2007–2015.
- Robert Nowak. 2009. Noisy generalized binary search. In *Advances in neural information processing systems*. 1366–1374.
- Jun-Jie Pan and Gerard J. Chang. 2006. Isometric path numbers of graphs. *Discret. Math.* 306, 17 (2006), 2091–2096.
- Ignacio M Pelayo. 2013. *Geodesic convexity in graphs*. Springer.
- Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. 2009. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 867–876.
- Florian Seiffarth, Tamás Horváth, and Stefan Wrobel. 2019. Maximal Closed Set and Half-Space Separations in Finite Closure Systems. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 21–37.
- Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.
- J. L. Sobrinho. 2001. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, Vol. 2. 727–735 vol.2.
- Marcel LJ van De Vel. 1993. *Theory of convex structures*. Elsevier.
- Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. (2002).