Abstract: The Cost of Learning Directed Cuts

Thomas Gärtner[1] and Gemma C. Garriga[2]

[1] Fraunhofer IAIS, Schloß Birlinghoven, 53754 Sankt Augustin, Germany
[2] HIIT Basic Research Unit, Helsinki University of Technology, Finland
thomas.gaertner@iais.fraunhofer.de, gemma.garriga@hut.fi

Classifying vertices in digraphs is an important machine learning setting with many applications. We consider learning problems on digraphs with three characteristic properties: (*i*) The target concept corresponds to a directed cut; (*ii*) the total cost of finding the cut has to be bounded a priori; and (*iii*) the target concept may change due to a hidden context.

For one motivating example consider classifying intermediate products in some process, e.g., for manufacturing cars or the control flow in software, as faulty or correct. The process can be represented by a digraph and the concept is monotone: Typical faults that appear in an intermediate product will also be present in later stages of the product. The concept may depend on a hidden variable as some pre-assembled parts may vary and the fault may occur only for some charges and not for others. In order to be able to trade off between the cost of having a faulty product and the costs needed to find the cause of the fault, tight performance guarantees for finding the bug are needed.

Performance guarantees proposed in machine learning literature can be distinguished into concept-dependent and concept-independent. *Concept-dependent guarantees* state that the performance of the learning algorithm depends on the *unknown* target concept's complexity. *Concept-independent guarantees* state that the performance of the learning algorithm depends on the instance space's complexity, or in a transductive setting on the given training and test sets. While both types of guarantees are interesting and challenging, learning theory has mainly concentrated on concept dependent guarantees. In contrast, for real-world applications, one often faces the question whether the costs of labelling a whole dataset exceeds a given budget or not. In this case, concept-independent, transductive, bounds are to be preferred over concept-dependent guarantees.

The first observation of this work is that for learning directed cuts we can achieve tight, concept-independent guarantees. Based on a fixed size of the minimum path cover, we establish logarithmic performance guarantees for online learning, active learning, and PAC learning. We furthermore show which algorithms and results carry over to learning intersections of monotone with anti-monotone concepts. An important contribution of this work concerns learning algorithms able to cope with concept drift due to hidden changes in the context, i.e., the concept depends on an unobservable variable that can change over time. Worst case guarantees in this setting are related to adverserial learning.

1 Learning Directed Cuts

Classification on graphs involves identifying a particular subset of the vertex set. In this paper we investigate identifying directed cuts. A *directed cut* in a digraph (V, E) is a set of vertices $U \subset V$ such that $\nexists u \in U, v \in V \setminus U : v \geq u$ where $v \geq u$ iff there is a directed walk from u to v, i.e., $\geq$ is the preorder on V induced by E. Identifying a directed cut can, equivalently, be seen as learning a monotone function $y : V \to \Omega$ where $\forall (u, v) \in E : y(v) \Rightarrow y(u)$. Without loss of generality, it is sufficient to consider learning directed cuts on directed acyclic graphs with a unique minimum and maximum (DAGs). All results on DAGs translate to general digraphs by transforming a general digraph to a DAG (contracting the strongly connected components and adding a 'super-source' and a 'super-sink') and applying a learning algorithm to the DAG.

As we are considering a transductive setting, i.e., the whole vertex set (without any labels) is input to the learning algorithm, we are interested in achieving costs logarithmic in the size of the graph. Hence, we consider a concept not learnable on a digraph (under some cost model) if we can not achieve polylogarithmic cost bounds. It can be seen that the general problem of identifying a directed cut is not learnable. Consider the DAG $([\![n+2]\!], (\{n+1\} \times [\![n]\!]) \cup ([\![n]\!] \times \{n+2\}))$ where $[\![n]\!] = \{1, \ldots, n\}$. Given $y(n+1) \wedge \neg y(n+2)$, and any information about the vertices $[\![n]\!] \setminus \{u\}$ for some $u \in [\![n]\!]$, we can not infer the label of u. This implies that—in the worst case—it is impossible to identify the correct directed cut without costs in the order of the size of the graph.

Hence, in the following we focus on parameters with respect to which directed cuts become fixed-parameter learnable. For that we could either focus on concept-dependent or concept-independent parameters. The related setting of identifying a cut in an undirected graph (see e.g. [1] for a derivation of the VC dimension of small cuts) has been found to be fixed parameter learnable where the parameter is the size of the concept cut, i.e., the number of edges between differently labelled vertices. Although we can show that directed cuts are also fixed parameter learnable if the size of the directed concept cut is considered as the parameter, the dependency of the total cost on a parameter of the concept is often not desireable. In particular, in cases where we want or need to estimate a priori the total cost of learning the cut, concept-dependent parameters such as the size of the concept cut are not sufficient.

A *path cover* of $U \subseteq V$ in a DAG $G = (V, E)$ is a set $Q \subseteq \mathcal{P}(G)$ of paths such that $U = \bigcup_{p \in Q} V(p)$. In our algorithmic analysis, we rely on a minimum path cover of $V(G)$. A minimum path cover Q^* can be found in polynomial time [2]. We assume wlog that Q^* contains only maximal paths and hence that each path starts at the source (labelled $\top$) and ends at the sink (labelled $\bot$). Table 1 summarises the results that can be obtained when the size of the minimum path cover Q^* of the DAG is fixed. In all but one of the considered learning settings, monotone concepts (directed cuts) as well as the intersection of monotone and anti-monotone concepts are fixed-parameter learnable, i.e., the total cost of learning is $O(|Q^*| \log |V|)$.

	Monotone Concepts	Intersections
Active query bound	$\leq \lvert Q^*\rvert \log \lvert V\rvert$	$= \lvert V\rvert$
Online mistake bound	$\leq \lvert Q^*\rvert \log \lvert V\rvert$	$\leq \lvert Q^*\rvert + 2\lvert Q^*\rvert \log \lvert V\rvert$
VC dimension	$= \lvert Q^*\rvert$	$\leq 2\lvert Q^*\rvert; \geq \lvert Q^*\rvert$
ϵ, δ-PAC sample complexity	$\leq \left\lceil \frac{\lvert Q^*\rvert}{\epsilon} \ln \frac{\lvert Q^*\rvert}{\delta} \right\rceil$	$\leq \left\lceil \frac{2\lvert Q^*\rvert}{\epsilon} \ln \frac{2\lvert Q^*\rvert}{\delta} \right\rceil$

Table 1. Total cost of learning monotone concepts as well as intersections of monotone and anti-monotone concepts in a DAG (V, E), for various cost models and fixed size of the minimum path cover Q^*. While '=' denotes that the result is exact for all digraphs, '$\leq$' ('$\geq$') denotes that the result is an upper (lower) bound that is obtained for some graphs but there might be special cases with smaller (larger) cost.

Notice that learning directed cuts generalises learning of monotone Boolean formulae by considering each Boolean instance as a vertex of a directed hypercube, i.e., the digraph $(2^{[\![n]\!]}, \{(U, U \cup \{v\}) \mid U \subseteq [\![n]\!] \wedge v \notin U\})$. Most results consider particular sets of functions with restricted complexity, e.g. the size of the smallest decision tree representing the concept [3]. As complexity is a concept dependent parameter, we can not use it to estimate classification costs a priori. Our results are orthogonal to these bounds, as depending on the concept class and the training and test instances, one or the other bound is tighter.

2 Learning Directed Cuts despite Changing Concepts

In many real world learning problems the target concept may change depending on a hidden context such as unobservable variations in the environment. For instance, consider again the digraph representing the assembly of a product from subparts. Learning a directed cut corresponds in this example to finding those stages of the process that introduced faults. Indeed, whether a fault is introduced at some stage or not, may depend on an unknown property, like variations in basic parts. Hence, the fault may occur only sometimes, depending on this hidden context. In order to be able to trade off between the cost of having a faulty product and the costs needed to find the cause of the fault, tight performance guarantees for learning with hidden contexts are needed.

For that, we consider a variant of active learning where the target concept $y_b : V \to \{\top, \bot\}$ also depends on a *hidden context* b that may change at any time. We can view this as having different monotone concepts C_b for each value of b and the oracle chooses a b for each query. For simplicity we restrict ourselves to Boolean contexts $b \in \{\top, \bot\}$. The aim of the learning algorithm is to discover either of the true concepts, $C_\top$ or $C_\bot$, despite the changing concepts. To obtain worst case bounds, we consider the oracle as an adversarial 'evil' oracle that chooses the hidden context to maximise the cost of learning either of the directed cuts. Unlike the setting without hidden context, here we will also encounter settings in which we can not find a true concept even by querying all vertices.

Traditional membership queries are not sufficient to learn directed cuts when a hidden context is present. To enable learnability we allow the algorithm to ask 'm-ary" queries, i.e., to query the label of m vertices at the same time, which the oracle is then forced to answer according to the same concept. In other words, during one m-ary query the context does not to change. Although 2-ary queries are more powerful than 1-ary queries, directed cuts with a hidden context are still not learnable with 2-ary queries. However, we have an algorithm that is guaranteed to find one of the true concepts for any DAG and any pair of concepts with at most $|Q^*|^2 + 2|Q^*| \log |V|$ many 3-ary membership queries.

3 Conclusions

The question whether a learning task is feasible with a given budget is an important problem in real world machine learning applications. Recent learning theoretical work has concentrated on performance guarantees that depend on the complexity of the target function or at least on strong assumptions about the target function. In this work we propose performance guarantees that only make natural assumptions about the target concept and that otherwise depend only on properties of the unlabelled training and test data. This is in contrast to related work on learning small cuts in undirected graphs where usually the size of the concept cut is taken as a (concept-dependent) learning parameter.

For learning on digraphs, which are a natural model for many technical processes, monotonicity is a very natural assumption. In this paper we proposed to use the size of the minimum path cover as a performance parameter for learning monotone concepts (directed cuts) on digraphs. On the one hand, this parameter can efficiently be computed from unlabelled training and test data only; and is, on the other hand, sufficiently powerful to make directed cuts fixed parameter learnable in the active, online, as well as PAC learning frameworks.

In real world learning problems on directed graphs, an additional problem is often that the concept that a learning algorithm tries to identify changes, depending on some hidden context. We hence extended the usual query learning model to include a hidden context that can change at any time and explored learnability with a more powerful query model. In particular, to enable learnability despite a changing concept, it is necessary (and sufficient) that the learning algorithm can query three different vertices at a time where it is ensured that the answers will be provided on the basis of the same concept.

References

1. Kleinberg, J.: Detecting a network failure. In: Foundations of Computer Science, IEEE (2000)
2. Ntafos, S., Hakimi, S.: On path cover problems in digraphs and applications to program testing. IEEE Transactions on Software Engineering (1979)
3. O'Donnell, R., Servedio, R.A.: Learning monotone decision trees in polynomial time. In: IEEE Conference on Computational Complexity, IEEE (2006)