Proceedings of Machine Learning Research vol :1–20, 2026 39th Annual Conference on Learning Theory

Active Learning and Covering Problems with Precedence

author names withheld

Editor: Under Review for COLT 2026

Abstract

In the Bayesian Active Learning a hidden hypothesis is required to be uncovered. To do so, the learner is allowed to perform tests, each of which reveals partial information about the hidden hypothesis. Upon receiving this information, the learner adaptively selects the next test to be performed. The goal is to uncover the hidden hypothesis while performing as few tests as possible in the worst or average case.

In the covering problems, we are given a set of items and a collection of subsets that cover these items. The objective is to select a sequence of subsets that covers all items, which minimizing the worst or average covering cost.

For both types of problems, a natural constraint may arise that some tests can only be performed only after certain other tests (or some subsets can only be selected after selecting certain other subsets). We model such constraints using directed acyclic graphs (DAGs) that impose precedence on the tests or subsets. This paper explores the connection of active learning and covering problems under such constraints.

We show that given any bicriteria $(O(1)\,,\alpha)$-approximation ratio for the Precedence Constrained Set Cover, we can obtain an $O(\alpha \cdot \log n)$-approximation ratio for the Worst Case Active Learning with precedence constraints, where n is the number of hypothesis. Similarly, we prove that given any $O(\beta)$-approximation ratio for the Precedence Constrained Min-Sum Set Cover, we can obtain an $O(\beta \cdot \log n)$-approximation ratio for the Average Case Active Learning with Precedence Constraints. Finally, we provide several approximation algorithms for the Set Cover and Min-Sum Set Cover problems with various types of precedence constraints.

Keywords: Bayesian active learning, Set cover, Precedence constraints, Approximation Algorithms, Decision Trees

Ogolne uwagi:

- **wszelkie uwagi pisze jako komenda ... dzieki czemu na koniec latwo sie pozbyc; nie bede raczej uwag dorzucal w mailach, aby nie zniknelo; wszystko co ponizej oczywiscie do dyskusji, a gdy bedzie zgoda i bedzie zaimplementowane, to bede usuwal artefakty.**

- **przejrzawszy sporo papierow z poprzedniego COLT, mam obserwacje, ze dobrym/typowym ukladem papieru jest: intro; our contribution; related work; outline(opcjonalnie); preliminaries; wyniki; appendix.**

- **intro zwykle nie jest zbyt rozlekle oraz prawie zawsze pozbawione lania wody. Czesto od razu definicja problemu, aby wprowadzic pojecia, aby moc szybko formalnie podac wyniki (our contribution)**

- **"front" artykulu, czyli wszystko do preliminaries to typowo 4-5 stron,**

- **front we wszystkich miejcach zawiera zwykle odnosniki do literatury a sekcja "related work" jest czesto tytulowana "other related work" lub cos w tym rodzaju**

- **czytelnik powiniem poza dowodami rozumiec baze przeczytawszy front (czyli rozumiec wyniki, widziec co papier robi) a jesli chce sie dowiedziec jak/dlaczego (dowody) to idzie dalej. Czesto recenzent jest leniwy i nie zajrzy dalej niz front... niestety.**
- **W zwiazku z powyzszym sekcja "our contribution" (potencjalnie w tytule dodamy "and techniques" jak Michal sugeruje) powinna sie pochwalic takze jakimis ciekawszymi trickami lub technikami uzytymi pozniej w dowodach.**

1. Introduction

pomysl na intro: zdefiniowac dwa glowne problemy: learning + set cover; zapowiedziec, ze sa powiazane ze soba i celem papieru jest przestudiowanie tych zaleznosci plus usyskanie konkretnych wynikow; pytanie/do sprawdzenia: czy ktores wyniki przypadkiem poprawiaja lub sa tozsame z najlepszymi znanymi bez precedensow; ewentualne inna "marketingowe" uwagi.

Consider a set $\mathcal{H}$ of n *hypotheses*, a set $\mathcal{T}$ of m *tests* and an unknown *target hypothesis* $h^* \in \mathcal{H}$ that needs to be discovered through an adaptive learning process. Each test $t \in \mathcal{T}$ is a partition of $\mathcal{H}$, that is, t consists of subsets of $\mathcal{H}$ such that $x \cap y = \emptyset$ for any $x.y \in t$ and $\bigcup t = \mathcal{H}$. As a result of executing a test $t \in \mathcal{T}$, questioner recives a *reply* that reveals $x \in t$ such that $h^* \in x$. That is, the questioner learns which subset of $\mathcal{H}$ that belongs to t contains the target. Each subsequent test is selected by questioner by taking into acocunt replies from all test to date. Without formally stating an optimization criterion we refer to the above as the *Adaptive Learing Process* (AL). (Another widely used name in the literature is the decision tree construction). The goal for the questioner is to output h^*.

Consider an arbitrary partial order $(\mathcal{T}, \preceq)$ that introduces a precedence relation between tests. This leads us to the two adaptive learning problems in which order to perform a test t, all its predecesors had to be performed previously. Hence we have the *Worst Case Adaptive Learing with Precedences* (WCALP) in which the goal is to compute the AL that respects the precedence constraints and outputs the target h^* by performing the minimum number of tests in the worst case. Similarly, in the *Average Case Adaptive Learing with Precedences* (ACALP) the optimization criterion changes to minimizing the number of queries done on average.

In this work we study connections between adaptive lerning with precedences and the covering problems defined as follows. We are given a set $\mathcal{U}$ of n items, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$, such that $\bigcup \mathcal{S} = \mathcal{U}$, an arbitrary partial order $(\mathcal{S}, \preceq)$ on these subsets and an integer k. We say that a subfamily $\mathcal{C} \subseteq \mathcal{S}$ *covers* at least k items from $\mathcal{U}$ if $|\bigcup \mathcal{C}| \geq k$. We ask for a $\mathcal{C} \subseteq \mathcal{S}$ that covers at least k items from $\mathcal{U}$ and for each $x \in \mathcal{C}$ and each $y \in \mathcal{S}$ such that $y \preceq x$ it holds $y \in \mathcal{C}$. In the *Precedence Constrained Set Cover* (PCSC) the goal is to minimize $|\mathcal{C}|$. A permutation $C_1, \ldots, C_k)$ of the elements in $\mathcal{C}$ is *consistent* with the partial order $(\mathcal{S}, \preceq)$ if for any C_i and C_j such that $C_i \preceq C_j$ it holds $i < j$. The *coverage time* of a $x \in \bigcup \mathcal{C}$ is the minimum index i such that $x \in C_i$. In the *Precedence Constrained Min-Sum Set Cover* (PCMSSC) the goal is to find a sequence $(C_1, \ldots, C_k)$ that minimizes the total coverage time of all items in $C_1 \cup \cdots \cup C_k$.

1.1. Our contribution

Pomysl na rozdzial:

- **zajawka, że wprpwadzimy nowe inne problemy (raz - jak pomocnicze; dwa - jako dopelnienie obrazu roznych rzeczy z litearatury)**
- **zdefiniowac pozostałe problemy z "diagramu" zaleznosci miedzy nimi**
- **diagram**
- **najwazniejsze twierdzenia**
- **tabelka na podsumowanie**

Consider following problems:

- The *Precedence Constrained Bayesian Active Learning Problem* consists a set of $\mathcal{H}$ of n hypothesis, a set $\mathcal{T}$ of m tests and a DAG (directed acyclic graph) $\mathcal{F} = \{\mathcal{T}, \preceq\}$ encoding the precedence constraints between available tests. Among $\mathcal{H}$ a hidden hypothesis is required to be encovered. To do so, the learner is allowed to perform tests, each of which reveals partial information about the hidden hypothesis. Upon receiving this information, the learner adaptively selects the next test to be performed. Importantly, in order to perform such test the learner needs to perform all of its predecesors in $\mathcal{F}$ first. The goal is to uncover the hidden hypothesis while performing as few tests as possible. Depending on the chosen criterion we distinguish between the *Precedence Constrained Worst Case Active Learning* (PCWCAL) and *Precedence Constrained Average Case Active Learning* (PCACAL) problems.

- The *Precedence Constrained Covering Problem* consists of a set of n items $\mathcal{U}$, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$ that cover these items, and a DAG $\mathcal{F} = \{\mathcal{S}, \preceq\}$ encoding the precedence constraints between available subsets. The goal is to select a sequence of tests that covers at least K items. Depending on the chosen criterion we distinguish between the *Precedence Constrained Set Cover* (PCSC) and *Precedence Constrained Min-Sum Set Cover* (PCMSSC) problems. In the first we are only interested in minimizing the number of selected subsets, while in the second we want to minimize the average time it takes to cover an item.

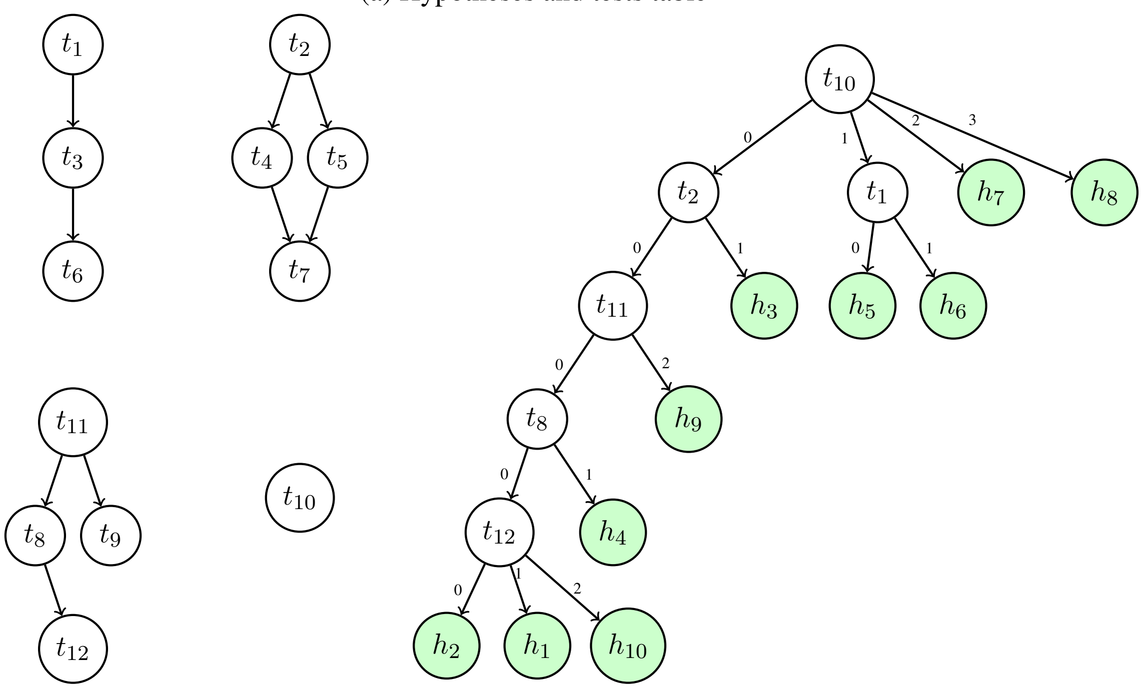
1.2. Our results and techniques

precedence/problem	**PCSC**	**PCMSSC**	**PCWCAL**	**PCACAL**
none	$O(\log n)$	4	$O(\log n)$	$O(\log n)$
inforest	$O(\log n)$*	4	$O(\log n)$*	$O(\log n)$*
outforest	$O(\log^2 n)$**	$O(\log n)$**	$O(\log^2 n)$*	$O(\log^2 n)$*
general	$O(\sqrt{n}\log n)$*	$O(\sqrt{n})$	$O(\sqrt{n}\log n)$*	$O(\sqrt{n}\log n)$*

Table 1: Approximation algorithms for various covering and active learning problems under different precedence constraints. (* denotes new results, ** denotes previously unmentioned corollaries of known results)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
h_1	0	0	0	0	0	0	2	0	0	0	0	1
h_2	0	0	0	0	0	1	0	0	0	0	0	0
h_3	0	1	0	1	0	0	0	0	0	0	0	0
h_4	0	0	0	0	1	0	0	1	0	0	0	0
h_5	0	0	0	0	0	0	1	0	0	1	1	0
h_6	1	0	0	2	0	0	0	0	0	1	0	0
h_7	0	0	0	0	0	2	0	0	1	2	0	0
h_8	0	0	1	0	0	0	0	0	0	3	0	0
h_9	0	0	0	0	0	0	0	1	1	0	2	0
h_{10}	0	0	0	1	0	0	0	0	0	0	0	2

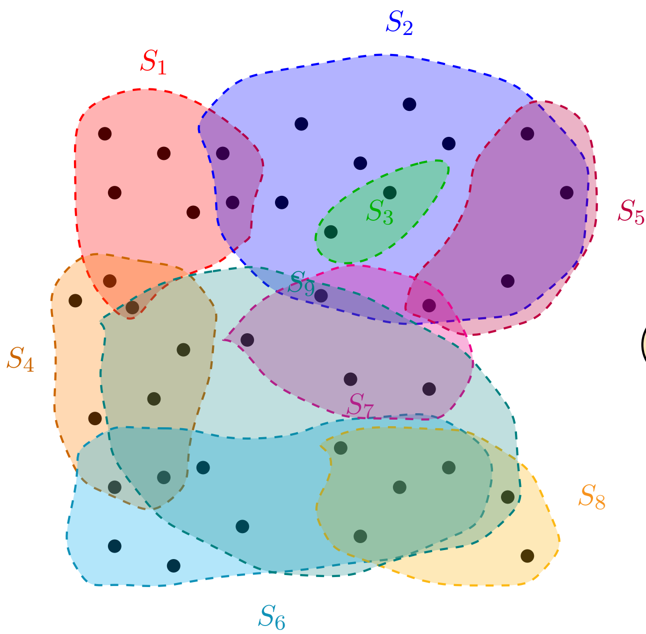
(a) Hypotheses and tests table



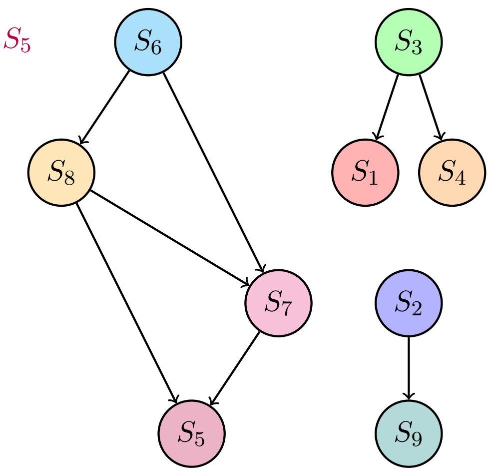
(b) Precedence DAG

(c) Valid decision tree

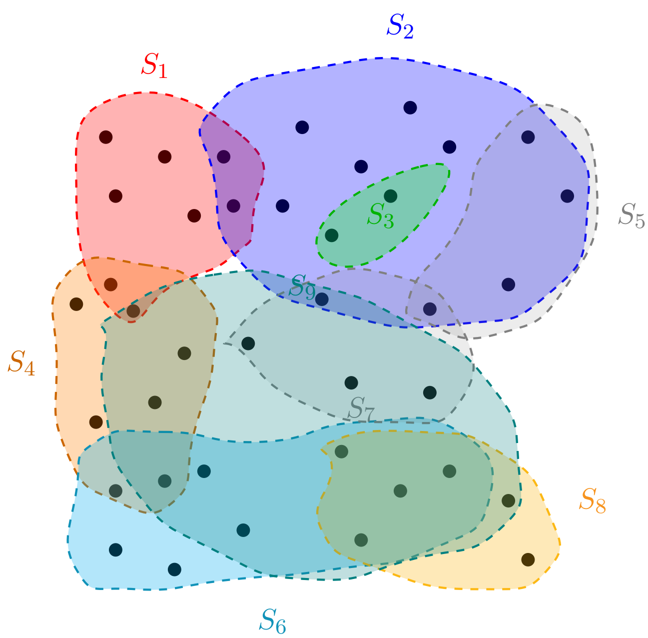
Figure 1: Example of a PCAL instance with 10 hypotheses and 12 tests. (a) Hypotheses-tests table. (b) Precedence DAG with four components. (c) A valid decision tree solution respecting precedence constraints.



(a) Universe and covering sets



(b) Precedence DAG



(c) Solution: selected sets $\{S_1, S_2, S_6, S_8, S_9\}$ (in color)

Figure 2: Example of a PCCP instance with 39 elements and 9 covering sets. (a) Universe with covering sets. (b) Precedence DAG with three components. (c) Solution using 7 selected sets (colored).

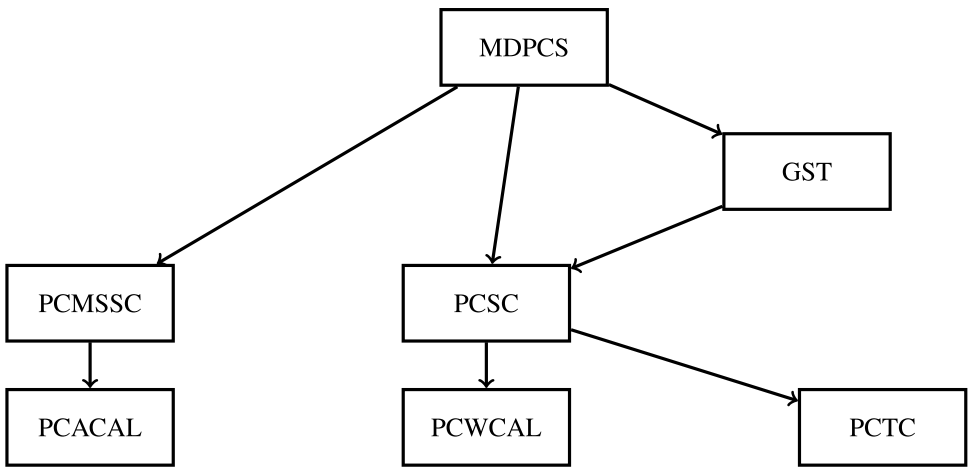


Figure 3: Relationships between covering and active learning problems, $\Pi_1 \rightarrow \Pi_2$ denotes that an approximation algorithm for problem Π_1 implies an approximation algorithm for problem Π_2.

2. Preliminaries

Definition 1 (Precedence constrained set cover (PCSC)) *Given a universe $\mathcal{U}$ of n items, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$, a DAG $\mathcal{F} = \{\mathcal{S}, \preceq\}$ encoding precedence constraints, and a coverage requirement K, find a precedence-closed subfamily $\mathcal{C} \subseteq \mathcal{S}$ that covers at least K items while minimizing $|\mathcal{C}|$.*

Definition 2 (Precedence constrained min-sum set cover (PCMSSC)) *Given a universe $\mathcal{U}$ of n items, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$, a DAG $\mathcal{F} = \{\mathcal{S}, \preceq\}$ encoding precedence constraints, and a coverage requirement K, find a precedence-closed sequence of sets that covers at least K items while minimizing the average time (position in sequence) at which items are covered.*

Definition 3 (Precedence constrained test cover (PCTC)) *Given a set $\mathcal{H}$ of n hypotheses, a set $\mathcal{T}$ of m tests, and a DAG $\mathcal{F} = \{\mathcal{T}, \preceq\}$ encoding precedence constraints, find a precedence-closed subfamily of tests that distinguishes all pairs of hypotheses.*

In the active learning problems, we are given a set of hypotheses $\mathcal{H}$ and a set of tests $\mathcal{T}$. Each test t is an arbitrary partition of $\mathcal{H}$ into disjoint subsets $U_{t,1}, U_{t,2}, \ldots, U_{t,r_t}$, where r_t is the number of possible responses to test t. When a test t is performed, the response indicates which subset $U_{t,j}$ contains the hidden hypothesis $h^* \in \mathcal{H}$. The tests are subject to precedence constraints encoded by a DAG $\mathcal{F} = \{\mathcal{T}, \preceq\}$, meaning that a test t can be performed only if all its predecessors in $\mathcal{F}$ have already been performed.

The goal is to design a *strategy* of learning which is an adaptive algorithm, that based on the previous responses provides the searcher with the next test to be performed. We represent this strategy as a rooted tree called a *decision tree* D, where each internal node represents a test to be performed and each leaf represents a hypothesis. We demand that each edge outgoing from the root r of D is associated with a unique response to a test r and that the same holds for all decision subtrees of $D - r$ (which are not leafs). It should be remarked that it is possible for a test to appear multiple times in the decision tree. Therefore, by $\mathcal{T}_D$ we denote subset of $\mathcal{T}$ that appear in decision tree D and by T_D we denote the set of inner nodes of D, so that each usage of a test in D corresponds to a unique element of T_D.

Let $T_{D,h}$ denote the sequence of tests performed when the hidden hypothesis is h and the learner follows the strategy represented by decision tree D. The cost of identifying hypothesis h using decision tree D is defined as $\texttt{COST}(D, h) = |T_{D,h}|$. We consider two cost measures for decision trees: the worst-case cost $\texttt{COST}_W(D) = \max_{h\in\mathcal{H}} \texttt{COST}(D, h)$ and the average-case cost $\texttt{COST}_A(D) = \sum_{h\in\mathcal{H}} \texttt{COST}(D, h)$ (up to a multiplicative factor of $1/|\mathcal{H}|$).

Definition 4 (Precedence constrained worst case active learning (PCWCAL)) *Given a set $\mathcal{H}$ of n hypotheses, a set $\mathcal{T}$ of m tests, and a DAG $\mathcal{F} = \{\mathcal{T}, \preceq\}$ encoding precedence constraints, construct a decision tree respecting precedence constraints that identifies any hypothesis from $\mathcal{H}$ while minimizing the worst-case depth of the tree.*

Definition 5 (Precedence constrained average case active learning (PCACAL)) *Given a set $\mathcal{H}$ of n hypotheses, a set $\mathcal{T}$ of m tests and a DAG $\mathcal{F} = \{\mathcal{T}, \preceq\}$ encoding precedence constraints, construct a decision tree respecting precedence constraints that identifies any hypothesis from $\mathcal{H}$ while minimizing the expected depth (average case cost).*

Let $\mathcal{H}_t$ denote the set of hypotheses that are not yet distinguished by the tests selected before test $t \in T_D$ in the decision tree. Then we immediately obtain the following simple observation:

Observation 1 *Let D be any decision tree for $\mathcal{I} = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then we have that:*

$$\mathit{COST}_A(D) = \sum_{t \in T_D} |\mathcal{H}_t|$$

Definition 6 (Group Steiner Tree (GST)) *Given an undirected graph $G = (V, E)$ with edge costs, a root vertex $r \in V$, and groups $g_1, \ldots, g_k \subseteq V$, find a minimum-cost tree T rooted at r that contains at least one vertex from each group g_i.*

For a subfamily $\mathcal{A} \subseteq \mathcal{F}$, we define the coverage as:

$$\operatorname{cov}(\mathcal{A}) \equiv \bigcup_{A \in \mathcal{A}} A$$

For a subset X of the universe, the coverage on X is:

$$\operatorname{cov}(\mathcal{A}, X) = \operatorname{cov}(\mathcal{A}) \cap X$$

The density Δ of a nonempty subfamily $\mathcal{A}$ on subset X is:

$$\Delta(\mathcal{A}, X) \equiv \frac{|\operatorname{cov}(\mathcal{A}, X)|}{|\mathcal{A}|}$$

For convenience, we define $\Delta(\emptyset, X) < 0$.

Definition 7 (Max-Density Precedence-Closed Subfamily (MDPCS)) *Given a family of m sets $\mathcal{G}$, a precedence relation $\prec$, and a set of n items to be covered $R \subseteq \operatorname{cov}(\mathcal{G})$, the MDPCS problem asks to find a precedence-closed subfamily $\mathcal{A} \subseteq \mathcal{G}$ that maximizes $\Delta(\mathcal{A}, R)$.*

For $S \in \mathcal{G}$, let $P[S]$ denote the minimal precedence-closed subfamily of $\mathcal{G}$ containing S (i.e., the ancestors of S including S itself).

3. Active Learning via Covering Problems

We begin with the following folklore lemma concerning both worst and average case learning.

Lemma 8 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any PCAL instance. Let $\mathcal{H}' \subseteq \mathcal{H}$. Then $\texttt{OPT}(\mathcal{H}', \mathcal{T}, \mathcal{F}) \leq \texttt{OPT}(I)$.*

3.1. Worst Case

Definition 9 (Coversep) *Let D be any decision tree for $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. We define a sequence of tests P_D called* coversep *as follows. Initially, P_D is empty and $\mathcal{H}' = \mathcal{H}$. While $|\mathcal{H}'| > |\mathcal{H}|/2$, we append to P_D the test $r(D_{\mathcal{H}'})$ and update $\mathcal{H}'$ to be the set of hypotheses corresponding to the child of $D_{\mathcal{H}'}$ that contains the most hypotheses. If $\texttt{COST}_W(D) = \texttt{OPT}_W(I)$, then we denote $P^*(I) = P_D$ (ties broken arbitrarily).*

It should be remarked that P_D is well-defined, as each test in P_D can have at most one child associated with more than half of the hypotheses in $\mathcal{H}'$. Since P_D is a subpath of D, we also have the following simple observation.

We will say that a test in $P^*(I)$ *sepcovers* an element $u \in \mathcal{H}$ in C if after applying the test t in C, u belongs to a response $\mathcal{H}'$ of size at most $|\mathcal{H}|/2$.

Observation 2 *Let I be any instance of PCWCAL. Then $|P^*(I)| \leq \texttt{OPT}_W(I)$.*

We will use $|P^*(I)|$ as a lower bound on $\texttt{OPT}_W(I)$ in the analysis of the approximation algorithm for PCWCAL. We have the following lemma:

Lemma 10 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any PCWCAL instance. Let S^* be the optimal solution for the PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ where $f = 1/4$ and a test t covers $h \in \mathcal{H}$ if $\{U_t(u)\} \leq \frac{3}{4} \cdot \{\mathcal{U}\}$. Then, $|S^*| \leq |P^*(I)|$.*

Proof We show that $P^*(I)$ is a feasible solution for the PCSC instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, K)$. Assume towards a contradiction that this is not the case, i. e. less than $|\mathcal{H}|/4$ are covered by $P^*(I)$. Therefore there exists $t \in P^*(I)$ and a response $\mathcal{H}'$ of size $|\mathcal{H}'| \leq |\mathcal{H}|/2$ such that hypotheses in $\mathcal{H}'$ are not covered by $P^*(I)$, otherwise the claim holds trivially, since all hypotheses are covered. Let $\mathcal{H}' \subseteq U_{t,j}$ (since $\mathcal{H}'$ is a response to a test t, such $U_{t,j}$ always exists). By assumption, we have that $|U_{t,j} - \mathcal{H}'| < |\mathcal{H}|/4$. Therefore, we have that $|U_{t,j}| = |\mathcal{H}'| + |U_{t,j} - \mathcal{H}'| < 3/4 \cdot |\mathcal{H}|$ which by definition means that h is covered by $P^*(I)$, a contradiction. ■

Theorem 11 *If there is an (γ, α)-bicriteria approximation algorithm for PCSC then there is an $O\left(\frac{\alpha}{\log\left(\frac{2\gamma}{2\gamma-1}\right)} \cdot \log n\right)$-approximation algorithm for PCWCAL. In particular when $\gamma = O(1)$, the approximation is $O(\alpha \cdot \log n)$.*

Proof The algorithm 1 is recursive and works as follows: Given an instance $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$, if $|\mathcal{H}| = 1$ we return the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$. Otherwise, we run the (γ, α)-approximation algorithm for PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$, where a test t covers element $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$. Let S be the returned set of tests. We build a decision tree D_S on tests from S closed under $\mathcal{F}$. For each $\mathcal{H}' \in \mathcal{H} - S$, we recursively call WORSTDECISIONTREE on instance $(\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S)$ and attach the returned decision tree to the leaf of D_S corresponding to $\mathcal{H}'$. Finally, we return the constructed decision tree D. The following observation follows by Lemmas 8 and 10.:

Algorithm 1: The $O(\alpha \cdot \log n)$-approximation algorithm for the PCWCAL

procedure WORSTDECISIONTREE$(\mathcal{H}, \mathcal{T}, \mathcal{F})$
- **if** $|\mathcal{H}| = 1$ **then**
  - **return** the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$.
- **end**
- **foreach** $t \in \mathcal{T}$ **do**
  - Set t to cover $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$.
- **end**
- $S \leftarrow$ Run the (γ, α)-approximation algorithm for PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$.
- $D \leftarrow D_S \leftarrow$ any decision tree built on tests from S respecting the precedence constraints $\mathcal{F}$.
- **foreach** $\mathcal{H}' \in \mathcal{H} - S$ **do**
  - $D' \leftarrow$ WORSTDECISIONTREE$(\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S)$.
  - Attach D' to the leaf of D corresponding to $\mathcal{H}'$.
- **end**
- **return** D.

Observation 3 *Let D_S be the decision tree built on tests from S respecting the precedence constraints $\mathcal{F}$. Then, $\texttt{COST}_W(D_S) \leq \alpha \cdot |P^*(I)|$.*

We are now ready to prove the theorem.

Lemma 12 *Let D be the decision tree returned by* WORSTDECISIONTREE *on input $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then, $\texttt{COST}_W(D, I) \leq \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma - 1}\right)} \cdot \log n \cdot \texttt{OPT}_W(I)$.*

Proof We prove the lemma by induction on n. The base case when $n = 1$ is trivial since the the cost of the decision tree is 0. Assume by induction that for every $I' = (\mathcal{H}', \mathcal{T}, \mathcal{F})$ such that $\mathcal{H}' \in \mathcal{H} - S$ and $n' = |\mathcal{H}'|$ we have $\texttt{COST}_W(D', I') \leq \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n' \cdot \texttt{OPT}_W(I')$, where D' is the decision tree returned by WORSTDECISIONTREE on input I'. We have that:

$$
\begin{aligned}
\texttt{COST}_W(D, I) \leq & \texttt{COST}_W(D_S, I) + \max_{\mathcal{H}' \in \mathcal{H} - S} \texttt{COST}_W\left(D', I'\right) \\
\leq & \alpha \cdot |S^*| + \max_{\mathcal{H}' \in \mathcal{H} - S} \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n' \cdot \texttt{OPT}_W\left(I'\right) \\
\leq & \alpha \cdot |P^*(I)| + \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log\left(\frac{(4\gamma - 1) \cdot n}{4\gamma}\right) \cdot \texttt{OPT}_W(I) \\
= & \alpha \cdot \texttt{OPT}_W(I) + \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n \cdot \texttt{OPT}_W(I) - \alpha \cdot \texttt{OPT}_W(I) \\
= & \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n \cdot \texttt{OPT}_W(I)
\end{aligned}
$$

where the second inequality follows by the induction hypothesis, the third inequality follows by Lemma 8 and the fact that $|\mathcal{H}'| \leq \frac{(4\gamma-1)}{4\gamma} \cdot n$ for every $\mathcal{H}' \in \mathcal{H} - S$ and the equalities follow by rearranging terms. This concludes the proof of the lemma. ■

■

3.2. Average Case

We follow a similar idea, however we use the connection to PCMSSC instead of PCSC. In order to lower bound the cost of the optimal decision tree, we will need the following notion: Let S be any sequence of tests in a decision tree D. Then let:

$$\texttt{COST}_A(S, I) = \sum_{t \in S} |\mathcal{H}_t| \,.$$

We have the following observations:

Observation 4 *Let D be any decision tree for $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ and let S be any sequence of tests in D. Then,*

$$\texttt{COST}_A(D, I) = \texttt{COST}_A(S, I) + \sum_{D' \in D - S} \texttt{COST}_A(D', I) \,.$$

As an immediate corollary for $D = D^*(I)$ and $S = P^*(I)$ we have:

Observation 5 *Let I be any instance of PCACAL. Then $\texttt{COST}_A(P^*(I), I) \leq \texttt{OPT}_A(I)$.*

This allows to use $\texttt{COST}_A(P^*(I), I)$ as a lower bound on $\texttt{OPT}_A(I)$ in the analysis of the approximation algorithm for PCACAL. We have the following lemma, analogous to Lemma 10.

Lemma 13 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any PCACAL instance. Let S^* be the optimal solution for the PCMSSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$, where a test t covers element $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$. Then, $\texttt{COST}_A(S^*) \leq \texttt{COST}_A(P^*(I))$.*

Proof Assume towards a contradiction that $\texttt{COST}_A(S^*) > \texttt{COST}_A(P^*(I))$. We will show that in such case there exists a cover $\sigma \subseteq P^*(I)$ such that $c_A(\sigma) < \texttt{COST}_A(P^*(I))$, a contradiction with the optimality of S^*. Let σ be the smallest subsequence of $P^*(I)$ which covers at least $|\mathcal{H}|/4$ elements. Such a subsequence exists, by repeating the argument used in the proof of Lemma 10. Let $h \in \mathcal{H}$. There are two cases to consider:

- σ covers h. Consider the first test t that sepcovered h in $P^*(I)$. By definition, tests previous to t in σ cover at most $|\mathcal{H}|/4$ elements. Since at the moment of sepcovering, h belonged to a response of size at most $|\mathcal{H}|/2$, we know that t also covers h in σ. This means that the contribution of h to $c_A(\sigma)$ is at most its contribution to $\texttt{COST}_A(P^*(I))$.

- σ does not cover h. In such case h is sepcovered by some test t in $P^*(I)$ but not in σ. therefore, the contribution of h to $c_A(\sigma)$ is $|\sigma|$ and its contribution to $\texttt{COST}_A(P^*(I))$ is at least $|\sigma|$.

Thus, we have that $c_A(\sigma) < \texttt{COST}_A(P^*(I))$, a contradiction. ■

Theorem 14 *If there is a β- approximation algorithm for PCMSSC then there is an $O(\beta \cdot \log n)$-approximation algorithm for PCACAL.*

Proof The idea behind Algorithm 2 is the same as for the worst case version of the problem except the fact that we use a solution to PCMSSC instead of PCSC.

Algorithm 2: The $O(\beta \cdot \log n)$-approximation algorithm for the PCACAL

procedure AVERAGEDECISIONTREE($\mathcal{H}, \mathcal{T}, \mathcal{F}$)

- **if** $|\mathcal{H}| = 1$ **then**
  - **return** the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$.
- **end**
- **foreach** $t \in \mathcal{T}$ **do**
  - Set t to cover $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$.
- **end**
- $S \leftarrow$ Run the β-approximation algorithm for PCMSSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$.
- $D \leftarrow D_S \leftarrow$ decision tree which consists of sequence of tests S.
- **foreach** $\mathcal{H}' \in \mathcal{H} - S$ **do**
  - $D' \leftarrow$ AVERAGEDECISIONTREE($\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S$).
  - Attach D' to the leaf of D corresponding to $\mathcal{H}'$.
- **end**
- **return** D.

Observation 6 *Let D_S be the decision tree consisting of the sequence of tests S. Then, $\mathit{COST}_A(D_S, I) \leq \beta \cdot \mathit{COST}_A(P^*(I), I)$.*

We are now ready to prove the theorem:

Lemma 15 *Let D be the decision tree returned by* AVERAGEDECISIONTREE *on input $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then, $\mathit{COST}_A(D, I) \leq \beta \cdot \log_{4/3} n \cdot \mathit{OPT}_A(I)$.*

Proof We prove the lemma by induction on n. The base case when $n = 1$ is trivial since the the cost of the decision tree is 0. Assume by induction that for every $I' = (\mathcal{H}', \mathcal{T}, \mathcal{F})$ such that $\mathcal{H}' \in \mathcal{H} - S$ and $n' = |\mathcal{H}'|$ we have $\mathtt{COST}_A(D', I') \leq \beta \cdot \log_{4/3} n' \cdot \mathtt{OPT}_A(I')$, where D' is the decision tree returned by AVERAGEDECISIONTREE on input I'. We have that:

$$
\begin{aligned}
\mathtt{COST}_A(D, I) \leq & \mathtt{COST}_A(D_S, I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \mathtt{COST}_A\big(D', I'\big) \\
\leq & \beta \cdot \mathtt{COST}_A(P^*(I), I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \beta \cdot \log_{4/3} n' \cdot \mathtt{OPT}_A\big(I'\big) \\
\leq & \beta \cdot \mathtt{COST}_A(P^*(I), I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \beta \cdot \log_{4/3}\left(\frac{3}{4} \cdot n\right) \cdot \mathtt{OPT}_A(I) \\
= & \beta \cdot \mathtt{COST}_A(P^*(I), I) + \beta \cdot \Big(\log_{4/3} n - 1\Big) \cdot \mathtt{OPT}_A(I) \\
\leq & \beta \cdot \log_{4/3} n \cdot \mathtt{OPT}_A(I)
\end{aligned}
$$

where the second inequality follows by the induction hypothesis, the third inequality follows by Lemma 8 and the fact that $|\mathcal{H}'| \leq \frac{3}{4} \cdot n$ for every $\mathcal{H}' \in \mathcal{H} - S$ and the equalities follow by rearranging terms. This concludes the proof of the lemma. ■

■

4. Binary search with precedence

In this section, we consider the special case of the PCWCAL and PCACAL where the instance $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ is an instance of the binary search problem with precedence constraints. In this setup we are given a linearly ordered set of n elements $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ with $h_1 \prec h_2 \prec \ldots \prec h_n$ and a set of tests $t_{i,j} \in \mathcal{T}$ corresponding to performing a comparison operation informing the learner whether the target element is less than or equal to h_j or greater than h_j. This problem is NP-hard (see Section **??**) however it is possible to derive an $O(\log n)$-approximation algorithm for both the PCWCAL and PCACAL.

The algorithm for a worst case is simple. We use the equivalence between binary searching in an ordered set and the edge ranking coloring of a path. An edge ranking of a path is a coloring of its edges such that any path between two edges of the same color contains an edge of a lower color **?**. Intuitevely, the color corresponds to the level of the decision tree where the test corresponding to the edge is performed. Let the input path be P. For any test $t \in \mathcal{T}$ define its *depth* as $d(t) = \max_{\tau \in \mathcal{T}, P_{\tau,t} \in F}\{d(\tau, t)\} + 1$. Let the height of $\mathcal{F}$ be defined as $h(\mathcal{F}) = \max_{t \in \mathcal{T}}\{d(t)\}$. The algorithm starts by partitioning $\mathcal{T}$ into $h(\mathcal{F})$ sets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{h(\mathcal{F})}$ such that for any $t \in \mathcal{T}_i$, $d(t) = i$. Then, the algorithm builds a decision tree processing layer one by one. Let $\mathcal{T}_i$ be such layer. We concatenate the (possibly disjoint) edges in T_i into a path P_i. Then, we compute an optimal edge ranking coloring of P_i using $\log n$ colors starting from color $(i-1) \cdot \log n$. Finally, we build a decision tree for the tests in $\mathcal{T}_i$ according to the edge ranking coloring. Observe that the resulting coloring is a valid edge ranking of P and that each edge has a color greater than all its predecessors in $\mathcal{F}$. Thus, the precedence constraints are respected. The final decision tree D is built recursively picking the root of the decision tree to be edge in P with the smallest color. it is easy to see that $\texttt{COST}_W(D, P) \leq h(\mathcal{F}) \cdot \log n$. Since the optimal decision tree has depth at least $h(\mathcal{F})$, the algorithm is an $O(\log n)$-approximation.

5. Set covering with constraints

5.1. Max-Density Precedence-Closed Subfamily (MDPCS)

The key to solce PCSC and PCMSSC is to solve the MDPCS problem. An approximation algorithm for MDPCS can be used as an essential subroutine in our algorithms for PCSC and PCMSSC. By **?**, the following greedy algorithm achieves an $O(\sqrt{m})$-approximation for MDPCS:

Algorithm 3: The greedy algorithm for MDPCS

```
procedure MDPCS-GREEDY(𝒢, ≺, R)
    𝒜 ← 𝒢
    foreach S ∈ 𝒢 do
        if Δ(P[S], R) > Δ(𝒜, R) then
            𝒜 ← P[S]
        end
    end
    return 𝒜
```

Let $\delta = \max_{S \in \mathcal{G}} \Delta(P[S], R)$. When $\delta \geq 1$, then the approximation factor of the greedy can also be bounded by $O(\sqrt{n})$. We show that if we enforce a certain condition on the input called *ϵ-shallow ancestry*, then for $\epsilon < 1$ the greedy algorithm achieves an $O(n^{\epsilon})$-approximation .

For each $S \in \mathcal{G}$, let $p(S) = |P[S]|$ and $c(S, R) = |\mathrm{cov}(P[S], R)|$.

Theorem 16 *Suppose there exists a constant $C > 0$ and $\epsilon \in (0, 1)$ such that for all $S \in \mathcal{G}$, $p(S) \leq C \cdot c(S)^{\epsilon}$ (ϵ-shallow ancestry). Then MDPCS-Greedy provides an $O(n^{\epsilon})$-approximation.*

Proof Let $\mathcal{A}^*$ be an optimal solution consisting of sets $S_1, \ldots, S_k$. There are two cases:

1. If $\delta \geq n^{1-\epsilon}$, then, we observe that $\Delta(\mathcal{A}, R) = \delta \geq n^{1-\epsilon}$. Since we can cover at most n elements with at least one set, we have $\Delta(\mathcal{A}^*, R) \leq n$. Therefore:
$$\frac{\Delta(\mathcal{A}^*, R)}{\Delta(\mathcal{A}, R)} \leq \frac{n}{n^{1-\epsilon}} = n^{\epsilon}$$
2. Else, if $\delta \leq n^{1-\epsilon}$, we proceed as follows: By definition of density, for any $S \in \mathcal{G}$, $c(S) \leq \delta \cdot p(S)$. Combining this with the ϵ-shallow ancestry condition, we have that for all $S \in \mathcal{G}$, $c(S) \leq \delta \cdot C \cdot c(S)^{\epsilon}$. Rearraning this inequality, we get that $c(S) \leq (\delta \cdot C)^{\frac{1}{1-\epsilon}}$. We have that:
$$|\mathrm{cov}(\mathcal{A}^*)| = \left| \bigcup_{j=1}^{k} \mathrm{cov}(S_j, R) \right| \leq \sum_{j=1}^{k} c(S_j) \leq k \cdot (\delta \cdot C)^{\frac{1}{1-\epsilon}}$$
Therefore:
$$\Delta(\mathcal{A}^*, R) = \frac{|\mathrm{cov}(\mathcal{A}^*, R)|}{k} \leq (\delta \cdot C)^{\frac{1}{1-\epsilon}}$$
By the greedy choice, $\Delta(\mathcal{A}, R) \geq \delta$ and by assumption $\delta \leq n^{1-\epsilon}$. Thus:
$$\frac{\Delta(\mathcal{A}^*, R)}{\Delta(\mathcal{A}, R)} \leq \frac{(\delta \cdot C)^{\frac{1}{1-\epsilon}}}{\delta} = C^{\frac{1}{1-\epsilon}} \cdot \delta^{\frac{\epsilon}{1-\epsilon}} \leq C^{\frac{1}{1-\epsilon}} \cdot (n^{1-\epsilon})^{\frac{\epsilon}{1-\epsilon}} = C^{\frac{1}{1-\epsilon}} \cdot n^{\epsilon}$$

Since C is constant, the theorem follows. ∎

5.2. Precedence constrained set cover

Tu poniżej miałem jakaś próbe pisania tego, ale sie pokomplikowało wiec ten pseuudokod jest niekompletny. To jest do zmiany wszystko:

We show the following:

Theorem 17 *If there exists an γ approximation algorithm for the MDPCS problem, then there exists an $(H_K + 1) \cdot \gamma$ - approximate algorithm for the PCSC problem.*

Proof ∎

Algorithm 4: The γ-greedy algorithm for PCSC

```
procedure PCSC(U, S, F, K)
    C ← ∅
    while |cov(C,U)| < K do
        A ← Run the γ-approx. algorithm for MDPCS on (U − C, S − C, F − C, m)
        if |cov(C ∪ A,U)| ≥ K then
            Find the minimum budget B ∈ [|A|], such that the γ-approx. algorithm for MDPCS on
            (U − C, S − C, F − C, B) returns a set B with |cov(B,U − C)| ≥ (K−cov(C,U))/α
            while |cov(C,U)| < K do
                B ← Run the γ-approx. algorithm for MDPCS on (U − C, S − C, F − C, B)
                C ← C ∪ B
            end
            return C
        end
        foreach u ∈ cov(A,U − C) do
            c(u) ← Δ(A,U − C)
        end
        C ← C ∪ A
    end
```

Theorem 18 *If the precedence constraints form an outforest, then there exists an bicriteria $(4, O(\log n))$-approximation algorithm for PCSC which be converted to an $O(\log^2 n)$ approximation algorithm.*

5.3. Precedence constrained min sum set cover

By **?**:

Theorem 19 *If there exists an γ approximation algorithm for the MDPCS problem, then there exists an $4 \cdot \gamma$ - approximate algorithm for the PCMSSC problem.*

Theorem 20 *If the precedence constraints form an outforest, then there exists an $(O(\log n))$-approximation algorithm for PCMSSC.*

6. Hardness

Theorem 21 *PCWCAL with outforest precedence constraints is NP-hard to approximate within a factor of* $O(\log^2 n)$ *unless P = NP.*

7. Conclusions and Future Work

Appendix A. My Proof of Theorem 1

This is a boring technical proof.

Appendix B. My Proof of Theorem 2

This is a complete version of a proof sketched in the main text.