Simultaneous Learning and Covering with Adversarial Noise

Andrew Guillory GUILLORY@CS.WASHINGTON.EDU
Jeff Bilmes BILMES@EE.WASHINGTON.EDU
University of Washington, Seattle, WA 98195, USA

Abstract

We study simultaneous learning and covering problems: submodular set cover problems that depend on the solution to an active (query) learning problem. The goal is to jointly minimize the cost of both learning and covering. We extend recent work in this setting to allow for a limited amount of adversarial noise. Certain noisy query learning problems are a special case of our problem. Crucial to our analysis is a lemma showing the logical OR of two submodular cover constraints can be reduced to a single submodular set cover constraint. Combined with known results, this new lemma allows for arbitrary monotone circuits of submodular cover constraints to be reduced to a single constraint. As an example practical application, we present a movie recommendation website that minimizes the total cost of learning what the user wants to watch and recommending a set of movies.

1. Background

Consider a movie recommendation problem where we want to recommend to a user a small set of movies to watch. Assume first that we already have some model of the user's taste in movies (for example learned from the user's ratings history or stated genre preferences). In this case, we can pose the recommendation problem as an optimization problem: using the model, we can design an objective function $F(S)$ which measures the quality of a set of movie recommendations $S \subseteq V$. Our goal is then to maximize $F(S)$ subject to a constraint on the size or cost of S (e.g. $|S| \leq k$). Alternatively we can minimize the size or cost of S subject to a constraint on $F(S)$ (e.g. $F(S) \geq \alpha$).

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

Without making any assumptions about F, both of these problems are intractable. A popular assumption that makes it possible to approximately solve these problems is the assumption that F is *submodular*. A set function $F : 2^V \to \mathbb{R}$ is submodular iff for every $A \subseteq B \subseteq (V \setminus \{s\})$

$$F(A + s) - F(A) \geq F(B + s) - F(B)$$

In other words, adding an item s to a smaller set A results in a larger gain than adding it to a larger set B. In our movie recommendation application, this means that the value of recommending a particular movie only decreases as we recommend other movies. For submodular objectives, maximizing $F(S)$ subject to $|S| \leq k$ is called *submodular function maximization*, and minimizing the cost of S subject to $F(S) \geq \alpha$ is called *submodular set cover*. $F(S)$ is called monotone non-decreasing if $F(A) \leq F(B)$ for $A \subseteq B$. F is called modular (additive, linear) if $F(A+s) = F(A)+F(\{s\})$ and normalized if $F(\emptyset) = 0$. A classic result is that for integer α and integer valued, monotone objectives a simple greedy algorithm for submodular set cover gives a solution within $\ln \alpha$ times the optimal solution (Wolsey, 1982). Many applications like document summarization (Lin & Bilmes, 2011), sensor placement (Krause et al., 2008), and viral marketing in social networks (Kempe et al., 2003) can be solved via submodular maximization or set cover.

In this work we consider a version of the movie recommendation problem in which we *do not* know the user's taste preferences and must actively acquire this information. We must both identify the user's preferences through feedback (*learn*) and recommend a small set of movies according to these preferences (*cover*). Our goal is to minimize the joint cost of both learning and covering. User feedback can take many forms. For example, we could assume that after recommending a movie to a user we are given feedback in the form of a movie rating. We also consider variations with separate learning actions and recommendation actions.

Other applications outside of movie recommendation can also be phrased in terms of simultaneous learning and covering. For example we may want to find a set of documents that summarizes all documents that a user is interested in; if we do not initially know which documents a user is interested in then we must both identify the documents of interest (learn) and also summarize them (cover). Similarly we may want to advertise to an initially unknown target group of users or place sensors in an initially unknown environment.

In previous work (Guillory & Bilmes, 2010), we introduced and analyzed a problem called *interactive submodular set cover*, a direct generalization of both exact active learning with a finite hypothesis class (i.e. query learning) and submodular set cover. Many simultaneous learning and covering type problems can be formally posed as interactive submodular set cover. Interactive submodular set cover can be seen as unifying submodular set cover (Wolsey, 1982) and query learning (Hanneke, 2006; Balcázar et al., 2007). Figure 1 shows the relationship between these problems.

Interactive Submodular Set Cover
Given:

- Hypothesis class H containing unknown h^*
- Query set Q and response set R with known $q(h) \subseteq R$ for $q \in Q$, $h \in H$
- Modular query cost function c defined over Q
- Submodular, monotone objective functions $F_h : 2^{Q \times R} \to \mathbb{R}_{\geq 0}$ for $h \in H$
- Objective threshold α

Protocol: Ask a question $\hat{q}_i \in Q$ and receive a response $\hat{r}_i \in \hat{q}_i(h^*)$.
Goal: Using minimal cost $\sum_{i=1}^{T} c(\hat{q}_i)$, guarantee $F_{h^*}(\hat{S}) \geq \alpha$ where $\hat{S} = \bigcup_{i=1}^{T}\{(\hat{q}_i, \hat{r}_i)\}$

In the movie recommendation example described above, the hypothesis class H corresponds to all possible models of the user's preferences, and the target hypothesis $h^* \in H$ is the (initially unknown) true model of the user's preferences. Queries correspond to actions available to the learning and covering algorithm; for example, we can have an action for each movie corresponding to recommending that movie and an action for each genre corresponding to asking the user if they are interested in that genre. Responses correspond to feedback from actions, and $q(h) \subseteq R$ is the set of allowable responses for question q if hypothetically h were the target hypothesis. For example, if q asks

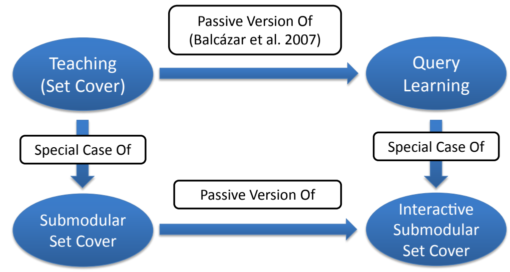

Figure 1. Related problems

the user if they are interested in horror movies then $q(h) = \{Yes\}$ if according to h the user likes horror movies and $q(h) = \{No\}$ if according to h the user does not. $q(h) = \{Yes, No\}$ if the model h is uninformative about this query. We assume $q(h) \neq \emptyset$ and that $q(h)$ is known (and available to an algorithm).

Finally, the objective function $F_h(\hat{S})$ for hypothesis h measures the quality of the recommendations given by $\hat{S}$ if hypothetically h were the target hypothesis. We assume F_h is submodular, monotone non-decreasing and known. The goal is to ensure $F_{h^*}(\hat{S}) \geq \alpha$ for a fixed α using minimal cost. Note that we assume that F_h is known for every h; what we do not know is h^*, which of the hypotheses is the target hypothesis. In general it is not necessary to identify h^* in order to ensure $F_{h^*}(\hat{S}) \geq \alpha$, but learning about h^* may be helpful. Define the *version space* $V(\hat{S})$ to be the subset of H consistent with $\hat{S}$.

$$V(\hat{S}) \triangleq \{h \in H : \forall (q, r) \in \hat{S},\ r \in q(h)\}$$

For worst case choice of h^*, in order to ensure that $F_{h^*}(\hat{S}) \geq \alpha$ it is both necessary and sufficient to ensure that $F_h(\hat{S}) \geq \alpha$ for every $h \in V(\hat{S})$.

Interactive submodular set cover makes the limiting, simplifying assumption that the target hypothesis h^* is in H. In terms of the recommendation example, interactive submodular set cover assumes that user's taste matches exactly one of a finite set of models we know in advance and that there is no noise in our feedback. In almost every real world learning application, this is an unrealistic assumption; typically, no single hypothesis will exactly match up with the observed question-response pairs because of noise or because we have an incorrect hypothesis class.

2. Noisy Interactive Set Cover

In this paper we propose a generalization of interactive submodular set cover that relaxes the assumption that

$h^* \in H$. With h^* not necessarily in H, it no longer makes sense to require $F_{h^*}(\hat{S}) \geq \alpha$; we only know objective functions F_h for $h \in H$ so if $h^* \notin H$ then we have no way of testing if $F_{h^*}(\hat{S}) \geq \alpha$. Instead we require that the covering constraint $F_h(\hat{S}) \geq \alpha$ is satisfied for all h that are sufficiently "close" to h^* (as measured through observed responses).

We propose defining closeness in terms of an additional submodular, monotone non-decreasing function $G_h(\hat{S})$ defined over question-response pairs. In particular, we require that the covering constraint $F_h(\hat{S}) \geq \alpha$ is satisfied for all h such that $G_h(S^*) < \kappa$ where κ is a known constant and S^* is the unknown set of all question-response pairs induced by h^*, $S^* \triangleq \bigcup_{q \in Q, r \in q(h^*)} \{(q, r)\}$. S^* is unknown so we cannot directly compute $G_h(S^*)$; however, the question-response pairs we observe $\hat{S}$ are a subset of S^* with which we can reason about $G_h(S^*)$. Intuitively, G_h can be thought of as computing the distance from h to h^* in terms of the question-response pairs allowed by h^*. We make the assumption that for any h, $\hat{S}$, and (q, r) such that $r \in q(h)$, $G_h(\hat{S} + (q, r)) - G_h(\hat{S}) = 0$. In other words, if we observe a question-response pair that agrees with h, then G_h does not increase. κ determines which hypotheses are close enough.

Noisy Interactive Set Cover
Given:

- Hypothesis class H (doesn't contain h^*)
- Query set Q and response set R with known $q(h) \subseteq R$ for $q \in Q$, $h \in H$ and unknown $q(h^*) \subseteq R$ for $q \in Q$
- Modular query cost function c defined over Q
- Submodular, monotone objective functions $F_h : 2^{Q \times R} \to \mathbb{R}_{\geq 0}$ for $h \in H$
- Submodular, monotone objective functions $G_h : 2^{Q \times R} \to \mathbb{R}_{\geq 0}$ for $h \in H$ with $G_h(S + (q, r)) - G_h(S) = 0$ for any S if $r \in q(h)$
- Objective threshold α, closeness threshold κ

Protocol: Ask a question $\hat{q}_i \in Q$ and receive a response $\hat{r}_i \in \hat{q}_i(h^*)$

Goal: Using minimal cost $\sum_{i=1}^{T} c(\hat{q}_i)$, guarantee $F_h(\hat{S}) \geq \alpha$ for all h such that $G_h(S^*) < \kappa$. Here $\hat{S} = \bigcup_{i=1}^{T} \{(\hat{q}_i, \hat{r}_i)\}$ and $S^* \triangleq \bigcup_{q \in Q, r \in q(h^*)} \{(q, r)\}$.

By setting $\kappa = 1$ and using $G_h(\hat{S}) \triangleq I(h \notin V(\hat{S}))$ (here I is the indicator function) we recover a variation of the original interactive submodular set cover problem. In this case satisfying the covering constraint for all h that agree with h^* corresponds exactly to satisfying the covering constraint for all h such that $G_h(S^*) < \kappa$. This is equivalent to the interactive submodular set cover problem if we include the optional assumption $G_h(S^*) < \kappa$ for at least one h (i.e. h^* agrees with some $h \in H$). We will show the algorithm we propose is approximately optimal regardless of whether this assumption is made.

We can interpret this more general problem as using an extended notion of version space such that a hypothesis h is no longer immediately eliminated as soon as a question-response pair $(\hat{q}_i, \hat{r}_i)$ with $\hat{r}_i \notin \hat{q}_i(h)$ is observed. Instead, a hypothesis h is only eliminated from consideration when $G_h(\hat{S}) \geq \kappa$. Different κ and G_h correspond to different notions of version space.

Lemma 1. *For an algorithm to ensure for any h^* that $F_h(\hat{S}) \geq \alpha$ for all h such that $G_h(S^*) < \kappa$ it is both necessary and sufficient to ensure that $F_h(\hat{S}) \geq \alpha$ for all h such that $G_h(\hat{S}) < \kappa$. The condition remains sufficient and necessary if the algorithm assumes $G_h(S^*) < \kappa$ for some h.*

Proof. To see this condition is sufficient note that $G_h(\hat{S}) \leq G_h(S^*)$. To see this condition is necessary note that for any particular h this inequality holds with equality for some h^* (the h^* which agrees with h on queries in $S^* \setminus \hat{S}$). We cannot therefore eliminate any h for which $G_h(\hat{S}) < \kappa$ as for some choice of h^* this hypothesis also satisfies $G_h(S^*) < \kappa$. This h^* satisfies the assumption $G_h(S^*) < \kappa$ for some h so the condition remains necessary with this assumption. □

We call a question asking strategy correct if it satisfies this condition. We now state our main result, the proof of which is given in Section 4: as for submodular set cover, a simple greedy algorithm is approximately optimal for noisy interactive set cover

Theorem 1. *For integer α and κ and integer monotone, normalized, submodular F_h and G_h, a greedy algorithm gives worst case cost within $1 + \ln(\kappa\alpha|H|)$ of that of any other correct question asking strategy. This also holds if we assume $G_h(S^*) < \kappa$ for some h.*

3. Application to Noisy Query Learning

Like interactive submodular set cover, noisy interactive set cover is related to query learning. In particular, we recover a version of query learning by using

$$F_h(\hat{S}) \triangleq |H \setminus V(\hat{S})| \qquad G_h(\hat{S}) \triangleq I(h \notin V(\hat{S}))$$

with $\kappa = 1$ and $\alpha = |H| - 1$. For these objectives, the goal of noisy interactive set cover is to eliminate $|H|-1$ hypotheses from the version space. This is equivalent to standard query learning if we make the additional assumptions that (1) there is an $h \in H$ that agrees with h^* on every query and (2) for each $h, h' \in H$ there is some $q \in Q$ with $q(h) \cap q(h') = \emptyset$. With these two assumptions, it is always possible to eliminate $|H| - 1$ hypotheses from the version space and the remaining hypothesis is the one hypothesis that agrees with h^*.

Noisy interactive set cover can also be used for query learning with noise. Define $\text{err}(h, \hat{S}) \triangleq \sum_{(q,r)\in\hat{S}} I(r \notin q(h))$ and $d(h, h') = |\{q : q(h) \cap q(h') = \emptyset\}|$.

Corollary 1. *Assume (1) some* $\hat{h} \in H$ *has* $\text{err}(\hat{h}, S^*) < \kappa$ *and (2) for every* $h, h' \in H$ $d(h, h') \geq 2\kappa - 1$*. Identifying* $\hat{h}$ *can be reduced to noisy interactive set cover with approximation ratio* $O(\ln(\kappa|H|))$

Proof. Using $\alpha = \kappa(|H| - 1)$, define

$$F_h(\hat{S}) \triangleq \sum_{h'\neq h} \min(\text{err}(h', \hat{S}), \kappa) \qquad G_h(\hat{S}) \triangleq \text{err}(h, \hat{S})$$

The goal of noisy interactive set cover is then to identify $|H| - 1$ hypotheses that make at least κ mistakes with respect to h^*. With assumptions (1) and (2) this is always possible and sufficient to identify $\hat{h}$. It is also necessary to identify $|H| - 1$ hypotheses that make at least κ mistakes in order to to identify $\hat{h}$; otherwise any hypothesis with less than κ mistakes may be $\hat{h}$. □

We show our algorithm is approximately optimal whether or not we assume (1). Also, assuming (2) has no effect on the analysis since it is an assumption about the hypothesis class H and our analysis holds for any H. Assumption (2) may hold, for example, when H is constructed with a clustering algorithm (Dasgupta et al., 2003) (e.g. by clustering users of a collaborative filtering system). As an alternative to these assumptions, we also show the following.

Corollary 2. *Consider the problem of identifying a set* $\hat{H} \subseteq H$ *such that* $h \in \hat{H}$ *for all* h *with* $\text{err}(h, S^*) < \kappa$ *and* $d(h, h') < 2\kappa - 1$ *for all* $h, h' \in \hat{H}$ *. This problem can be reduced to noisy interactive set cover with approximation ratio* $O(\ln(\kappa|H|))$

Proof. Using $\alpha = \kappa(|H| - 1)$, let $G_h(\hat{S}) \triangleq \text{err}(h, \hat{S})$,

$$F_h(\hat{S}) \triangleq \sum_{h':d(h,h')\geq 2\kappa-1} \min(\text{err}(h', \hat{S}), \kappa) + c_h$$

where $c_h \triangleq \kappa|\{h' : h' \neq h,\ d(h, h') < 2\kappa - 1\}|$. A set of query-response pairs solves noisy interactive set cover with these objectives iff in the final version space all remaining h are within $2\kappa - 2$ of each other and all eliminated h make at least κ mistakes. □

Dasgupta et al. (2003) study a similar query learning setting with adversarial noise. Under the hypothesis class assumptions of Corollary 1, their algorithm gives an $O(\ln |H|)$ approximation. We suspect our added dependence on κ is due to the increased flexibility of G_h in the more general problem (G_h does not need to be defined in terms of an additive or metric loss function and can be hypothesis dependant). We also note that Dasgupta et al. (2003) assume $|q(h)| = 1$ and uniform query costs, so our results are novel despite the added dependence on κ. Assuming only that H contains one or more h with $\text{err}(h, S^*) < \kappa$, Dasgupta et al. give an $O(\ln |H|)$ multiplicative approximation for finding one such h, but in this case the algorithm also has an additional additive approximation term which depends on κ and the number of hypotheses near any h. The analysis uses two phases, with the first phase similar to the problem solved in Corollary 2. Recently Golovin et al. (2010) gave approximation results for an average-case noisy query learning setting. Bellala et al. (2010) consider related average-case problems.

4. Greedy Algorithm and Analysis

In previous work (Guillory & Bilmes, 2010), we showed that a simple greedy algorithm is approximately optimal for interactive submodular set cover. The analysis proceeds in two steps. First, the optimization problem over many objectives F_h is reduced to a simpler problem over a single objective $\bar{F}$. Second, a greedy algorithm for this simpler problem is proposed and analyzed. In this section we give an analysis for noisy interactive set cover with this structure. The key insight is that noisy interactive set cover can also be reduced to a problem over a (different) single objective $\bar{F}$.

Define $F_{h,\alpha}(\hat{S}) \triangleq \min(F_h(\hat{S}), \alpha)$ to be version of F_h truncated at α. Define the following objective

$$\bar{F}(\hat{S}) \triangleq \frac{1}{\kappa|H|} \sum_{h\in H} ((\kappa - G_{h,\kappa}(\hat{S}))F_{h,\alpha}(\hat{S}) + G_{h,\kappa}(\hat{S})\alpha)$$

$\bar{F}(\hat{S}) \geq \alpha$ iff this set of query response pairs solves the noisy interactive set cover problem. That is, $\bar{F}(\hat{S}) \geq \alpha$ iff $F_h(\hat{S}) \geq \alpha$ for all h such that $G_h(\hat{S}) < \kappa$. From Lemma 1 this is sufficient and necessary to ensure $F_h(\hat{S}) \geq \alpha$ for all h such that $G_h(S^*) < \kappa$ for worst case h^*. We show $\bar{F}(S)$ is submodular.

Lemma 2. *Let* $F(S)$ *be a monotone non-decreasing submodular function ranging between* 0 *and* α*. Let* $G(S)$ *be a monotone non-decreasing submodular function ranging between* 0 *and* κ*. Then*

$$\bar{F}(S) \triangleq (\kappa - G(S))F(S) + G(S)\alpha$$

is a monotone non-decreasing submodular function.

Algorithm 1 Worst Case Greedy

$\hat{S} \Leftarrow \emptyset$
while $\bar{F}(\hat{S}) < \alpha$ **do**
  $\hat{q} \Leftarrow \text{argmax}_{q_i \in Q} \min_{r_i \in R}$
    $(\bar{F}(\hat{S} + (q_i, r_i)) - \bar{F}(\hat{S}))/c(q_i)$
  Ask $\hat{q}$ and receive response $\hat{r}$
  $\hat{S} \Leftarrow \hat{S} + (\hat{q}, \hat{r})$
end while

Proof. As a short hand we use $\delta_S(F, x) \triangleq F(S + x) - F(S)$. We show $\bar{F}(S)$ is monotone non-decreasing by showing that for any A and x, $\delta_A(\bar{F}, x) \geq 0$.

$$\begin{aligned} &\delta_A(\bar{F}, x) \\ &= \kappa\delta_A(F, x) + \delta_A(G, x)\alpha \\ &\quad + G(A)F(A) - G(A + x)F(A + x) \\ &= (\kappa - G(A))\delta_A(F, x) + \delta_A(G, x)(\alpha - F(A + x)) \end{aligned}$$

We see that each of these terms is positive so long as F and G are monotone non-decreasing and range between 0 and α and 0 and κ respectively. We now show $\bar{F}(S)$ is submodular by showing, for any $A \subseteq B$ and x, $\delta(\bar{F}, A, x) \geq \delta(\bar{F}, B, x)$. As before we have

$$\delta_B(\bar{F}, x) = (\kappa - G(B))\delta_B(F, x) + \delta_B(G, x)(\alpha - F(B + x))$$

Each term in this expression is less than the corresponding term in $\delta_A(\bar{F}, x)$. □

We note Lemma 2 does not follow trivially from any of the standard results for combining submodular functions of which we are aware. In particular, the term $(\kappa - G(S))F(S)$ is not by itself submodular so the result doesn't follow from the submodularity of the sum of two submodular functions.

Corollary 3. *$\bar{F}(S)$ is submodular monotone non-decreasing whenever F_h and G_h are submodular monotone non-decreasing for all $h \in H$.*

Proof. The result follows from Lemma 2 and the following known results. (1) When a function $F(S)$ is submodular and monotone non-decreasing, so is $F_\alpha(S) = \min(F(S), \alpha)$ for a constant α. (2) When $F_h(S)$ is submodular and monotone non-decreasing for every $h \in H$, $\bar{F}(S) = \sum_{h \in H} F_h(S)$ is also submodular and monotone non-decreasing. □

Having established the submodularity of $\bar{F}$, we now define a greedy algorithm for solving the simplified problem over $\bar{F}$. See Algorithm 1. At each time step the algorithm performs the query $q_i \in Q$ that maximizes

$$\min_{r_i \in R} (\bar{F}(\hat{S} + (q_i, r_i)) - \bar{F}(\hat{S}))/c(q_i)$$

This algorithm is the same as the worst case greedy algorithm for interactive submodular set cover with two exceptions: (1) the objective function $\bar{F}$ is different and (2) the response for a question q_i is no longer required to match the response set $q_i(h)$ for some $h \in V(\hat{S})$. This first difference does not alter the analysis. The second difference does although not as dramatically as one might expect. In fact optimality follows by noting the this simplified problem is an interactive submodular set cover problem over a single objective $F_h = \bar{F}$ with $q(h) = R$ for every q. We give a more direct analysis however with which we also show approximate optimality when we assume $G_h(S^*) < \kappa$ for some h (restricting h^*). Recall this assumption was useful for noisy query learning.

Define oracles $T \in R^Q$ to be functions mapping questions to responses and $T(\hat{Q}) \triangleq \bigcup_{\hat{q}_i \in \hat{Q}} \{(\hat{q}_i, T(\hat{q}_i))\}$. $T(\hat{Q})$ is the set of question-response pairs given by T for the set of questions $\hat{Q}$. Define General Cover Cost

$$GCC \triangleq \max_{T \in R^Q} \Big(\min_{\hat{Q}: \bar{F}(T(\hat{Q})) \geq \alpha} c(\hat{Q}) \Big)$$

GCC depends on Q, R, α, c, and $\bar{F}$ but this dependence is suppressed for simplicity of notation. GCC is the cost of satisfying $\bar{F}(T(\hat{Q})) \geq \alpha$ for worst case choice of T where the choice of T is given to the algorithm selecting $\hat{Q}$. GCC generalizes many complexity terms used in analyses of query learning (e.g. Teaching Dimension) (Hanneke, 2006; Balcázar et al., 2007).

The same GCC quantity is also used in the analysis of interactive submodular set cover (Guillory & Bilmes, 2010). The analysis proceeds by showing that GCC lower bounds the optimal worst case cost and approximately upper bounds the worst case cost of the greedy algorithm. What changes subtly in our analysis is the lower bound portion of the argument. In our problem, the optimal worst case cost is no longer the same since we now allow $h^* \notin H$. We show that GCC still lower bounds the cost of satisfying $\bar{F}(\hat{S}) \geq \alpha$ for worst case h^*. We've already argued this is necessary and sufficient for solving noisy interactive set cover, and therefore this shows GCC lower bounds worst case cost.

Lemma 3. *If there is a question asking strategy for satisfying $\bar{F}(\hat{S}) \geq \alpha$ with worst case cost C^* then $GCC \leq C^*$.*

Proof. Assume to show a contradiction that there is a question asking strategy which satisfies $\bar{F}(\hat{S}) \geq \alpha$ using worst case cost $C^* < GCC$. By definition of GCC, there is an oracle T^* with

$$\min_{\hat{Q}: \bar{F}(T^*(\hat{Q})) \geq \alpha} c(\hat{Q}) = GCC > C^*$$

We can then construct a corresponding target hypothesis h^* by setting $q(h^*) = \{T(q)\}$. When this h^* is the target hypothesis, any sequence of questions $\hat{Q}$ with cost less than or equal to C^* must have $\bar{F}(\hat{S}) < \alpha$. This contradicts our assumption. □

We show that in fact $GCC \leq C^*$ even if we make the additional assumption that some for some h, $G_h(S^*) < \kappa$. This is a stronger result since C^* is smaller in this case as we have placed an additional restriction on h^*.

Lemma 4. *Assume for some h $G_h(S^*) < \kappa$. If there is a question asking strategy for satisfying $\bar{F}(\hat{S}) \geq \alpha$ with worst case cost C^* then $GCC \leq C^*$.*

Proof. Assume again to show a contradiction that there is a question asking strategy which satisfies $\bar{F}(\hat{S}) \geq \alpha$ using worst case cost $C^* < GCC$. As before there is a T^* such that any sequence of questions $\hat{Q}$ with total cost less than or equal to C^* must have $\bar{F}(\hat{S}) < \alpha$. If $\bar{F}(\hat{S}) < \alpha$ then there must be some remaining h with $G_h(\hat{S}) < \kappa$ and $F_h(\hat{S}) < \alpha$. For any such $\hat{Q}$, we can then construct an h^* such that (1) $G_h(S^*) < \kappa$ for some h and (2) it is possible to answer questions consistently with h^* so that $G_h(\hat{S}) < \kappa$ and $F_h(\hat{S}) < \alpha$ for some h. In particular, use $q(h^*) = \{T^*(q)\}$ for $q \in \hat{Q}$ and $q(h^*) = q(h)$ for $q \notin \hat{Q}$ where h is the hypothesis with $G_h(\hat{S}) < \kappa$ and $F_h(\hat{S}) < \alpha$ when T^* is used to answer $\hat{Q}$. □

The upper bound follows that for the noise-free case (Guillory & Bilmes, 2010) giving Theorem 1.

5. Monotone Circuits of Constraints

One interpretation of noisy interactive set cover is in terms of a boolean circuit of submodular cover constraints. For every h we want either $F_h(\hat{S}) \geq \alpha$ or $G_h(\hat{S}) \geq \kappa$. Figure 2 shows the boolean circuit encoding this. This circuit is monotone (i.e. it only involves AND and OR gates). A key part of our analysis is showing that this can be reduced to a single constraint $\bar{F}(\hat{S}) \geq \alpha$. In this section we show that, in fact, Lemma 2 can be used in combination with known results (Krause et al., 2008) to reduce *any* monotone boolean circuit of constraints to a single constraint.

Lemma 5. *Given $i = 1 \ldots n$ constraints $F_i(S) \geq \alpha_i$ for normalized, monotone submodular F_i and any monotone boolean circuit over these constraints, there is a normalized, monotone submodular $\bar{F}(S)$ and $\bar{\alpha}$ such that $\bar{F}(S) \geq \bar{\alpha}$ iff the circuit evaluates to true.*

Proof. We first convert the constraints $F_i(S) \geq \alpha_i$ into constraints $\hat{F}_i(S) = \alpha_i$ where $\hat{F}_i(S) = \min(F_i(S), \alpha_i)$.

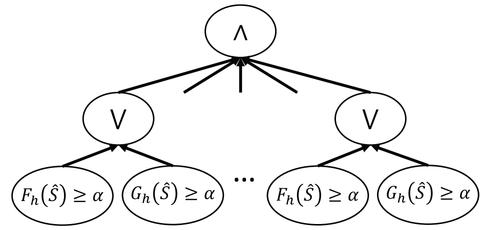

Figure 2. Noisy interactive set cover as a circuit.

$\min(F_i(S), \alpha_i)$ is submodular and ranges between 0 and α_i.

We now show how to reduce the OR of two constraints $(\hat{F}_i(S) = \alpha_i) \vee (\hat{F}_j(S) = \alpha_j)$ to a single constraint $\bar{F}(S) = \bar{\alpha}$ where $\bar{F}$ ranges between 0 and $\bar{\alpha}$. Define

$$\bar{F}(S) = (\alpha_i - \hat{F}_i(S))\hat{F}_j(S) + \hat{F}_i(S)\alpha_j$$

It is not hard to see that indeed $\bar{F}(S) = \alpha_i\alpha_j$ iff $(\hat{F}_i(S) = \alpha_i) \vee (\hat{F}_j(S) = \alpha_j)$. Lemma 2 shows that $\bar{F}$ is submodular and monotone.

We now show how to reduce the AND of two constraints $(\hat{F}_i(S) = \alpha_i) \wedge (\hat{F}_j(S) = \alpha_j)$. Define

$$\bar{F}(S) = \hat{F}_i(S) + \hat{F}_j(S)$$

as in (Krause et al., 2008). $\bar{F}(S) = \alpha_i + \alpha_j$ iff $(\hat{F}_i(S) = \alpha_i) \wedge (\hat{F}_j(S) = \alpha_j)$. $\bar{F}$ is submodular and monotone since it is the sum of monotone, submodular functions.

In both cases $\bar{F}(S) \geq \bar{\alpha} \iff \bar{F}(S) = \bar{\alpha}$ since $\bar{F}$ ranges from 0 to $\bar{\alpha}$. Furthermore, AND and OR combinations are all we need, since any monotone circuit can be written in terms of these operators. □

With this lemma we can solve (interactive) submodular set cover problems with constraints given by arbitrary monotone circuits. The $\bar{F}$ and $\bar{\alpha}$ derived are integer valued when all F_i and α_i are integer valued. We also need for $\bar{\alpha}$ to be small for the reduction to be useful. Assume for simplicity that $\alpha_i = \alpha$ for all i and the circuit is written in disjunctive normal form (this is always possible). In this case, $\bar{\alpha} \leq c_1\alpha^{c_2}$ where c_1 is the number of clauses in the circuit and c_2 the size of the largest clause in the circuit (the OR nodes multiply and the AND nodes add). The approximation ratio given by the greedy algorithm is then $\ln \bar{\alpha} \leq c_2 \ln c_1\alpha$. We therefore expect this reduction to be useful if c_2 is small (in our problem it is 2). This general reduction may be of interest outside of our problem. For example, in an advertising application we can require that each of several demographics is influenced by at least one of several advertising campaigns.

6. Movie Recommendation

As an example real world application of our theoretical results, we present a website we have developed for movie recommendation which minimizes the total cost of learning and recommending. The standard approach to movie recommendation uses collaborative filtering (Breese et al., 1998). We take a different, complimentary approach. Instead of learning from a user's fixed rating history, we directly asking the user questions. Furthermore, instead of trying to learn a comprehensive model of the user's taste, we try to learn what the user wants to watch *right now*. We think this approach is well suited for streaming services.

6.1. System Design

We use the Netflix API to retrieve a catalog of all movies and TV shows available through Netflix's Watch Instantly service (approximately 11000 titles). We set our hypothesis class H to be this set of available titles. With this H, assuming $h^* \in H$ corresponds to assuming that the user wants to watch a single title.

In our query set Q we use four types of questions: (1) genre questions such as "Do you want to watch something from the Comedy genre?", (2) release year questions such as "Do you want to watch something from the 90s?", (3) runtime questions such as "Do you want to watch something under an hour long?", and (4) cast and director questions such as "Do you want to watch a movie featuring Tom Hanks?". In addition, we include in Q a recommendation action for each title. We assume these actions have no feedback and assign them cost .1 that of a question.

We experimented with three different versions of our website which use different combinations of objective functions F_h and G_h. Each version of the website is created by solving the corresponding noisy interactive set cover problem using the worst-case greedy algorithm. In our current implementation, we solve the problem for all possible sequences of responses and use the resulting decision tree to generate a static web page. With implementation tricks (Minoux, 1978) for speeding up the greedy algorithm, we've found that writing the web page to disk usually dominates the run time. Each question is presented to the user on a separate page along with a set of 6 positive examples (displayed as box art thumbnail images) taken from the current version space. Recommendations are presented in lists with box art and a plot synopsis.

Our first, simplest version uses $\kappa = 1$ and $G_h(\hat{S}) \triangleq I(h \notin V(\hat{S}))$. For the covering objective, this first version uses $\alpha = 1$ and $F_h(\hat{S}) = 1$ iff $\hat{S}$ includes the recommendation action corresponding to h. This very simple problem does not use the full power of noisy interactive set cover and, in fact, is equivalent to query learning with membership and equivalence queries.

Our second version uses $\kappa = 2$ and $G_h(\hat{S}) \triangleq \text{err}(h, \hat{S})$. As compared to the first version, the second version does not eliminate a title from consideration until the user's responses have disagreed with this title twice.

Our third version again uses $\kappa = 2$ but uses a more complicated modular function for G_h which encodes domain knowledge about the problem. For example, if the user responds "Yes" to a genre question then our $G_h(\hat{S})$ increases by 2 for h that are not in that genre or any related genres but only by 1 for h that are not in the genre but are in related genres. We use similar heuristics for release year and runtime questions.

The third version also uses a more complicated covering objective F_h. We use $F_h(\hat{S}) = 1$ if $\hat{S}$ contains either the recommendation action corresponding to h or a recommendation action corresponding to a title h' which *covers* h. We say a title h' covers h if: (1) h is in the list of titles similar to h' available from the Netflix API, (2) h' is tagged with all genres h is tagged with, and (3) the average user rating for h' is greater than or equal to that for h. This essentially defines a directed graph over titles. We also use this same directed graph to create a small list of related titles displayed with each recommendation. In this way, titles which are relevant but covered by another recommendation are often still shown to the user but with less emphasis. For consistency we display these same related titles in all three versions of the website.

6.2. User Study Results

We conducted a small user study comparing the three different versions of our website. We asked each user to try each version of the website (presented in a different random order for each user) and then fill out a short survey. For each website, the survey asked users how strongly they agree with four statements. 1. The website was useful for finding things to watch. 2. The website's questions were easy to answer. 3. The website's recommendations matched my responses to the questions that were asked. 4. The website asked the right number of questions before recommending things to watch. We also asked the users to describe the differences between the sites and give suggestions.

We received 59 survey responses. Table 1 shows the average responses. The variance was relatively high, but there are some large significant differences in particular Website 1 vs Website 3 on Statements 1 and

Table 1. Average survey responses. 5 is Strongly Agree, 1 is Strongly Disagree. Std. dev. shown in parentheses.

Statement	Website 1	Website 2	Website 3
1	3.86 (0.90)	3.52 (1.08)	3.05 (1.02)
2	4.14 (0.80)	4.00 (0.81)	4.05 (0.78)
3	3.95 (0.97)	3.58 (1.00)	3.21 (1.07)
4	3.44 (1.15)	2.83 (1.13)	2.90 (1.16)

3. Surprisingly, there is an average preference for the simplest site, Website 1, which assumes $h^* \in H$. 45/59 participants rated Website 1 as being more useful or as useful compared to the other websites. 35/59 rated Website 2 as being more or as useful and only 23/59 rated Website 3 as being more or as useful.

Several users felt that Website 2 asked too many questions. One user commented that Website 2 asked questions that were too similar to previously asked questions. These were issues we hoped to address with the more complicated G_h and F_h objectives used in Website 3, and this partially worked; users did not think that Website 3 asked too many questions.

In fact, because its F_h objective assigns larger gains to certain titles, Website 3 often recommended a few titles early and followed up with further questions and recommendations. A few users specifically mentioned that they liked this. More users however commented they received poor or unexpected recommendations too early and gave Website 3 lower scores because of this. Some users did not seem to realize that by clicking "More" they would be asked further questions and given more recommendations. To remedy this, about midway through the study we changed the UI to make this link more apparent, but this change did not seem to improve perception of Website 3, and it was still not clear how many users chose to continue after the first recommendation. Space prevents reporting results for before and after, so we present the average. More results are included in the supplementary material.

Since users tended to prefer the simpler website, one could interpret these results as evidence against the practical utility of our more flexible theoretical results. However, in light of the user comments we received explaining their issues with the more complex websites, we think these results are more indicative of problems designing objective and cost functions by hand. One very promising direction for future work would be to learn the objectives. The user feedback also suggests some specific ways our theory may be useful in future systems. For example, our results suggest that, if any noise is allowed, a more complex model than additive zero-one error is necessary as this simple model resulted in the system asking too many questions. Several users also reported they wanted the ability to respond "Maybe". Incorporating such a response would be problematic in a noise free query learning setting: if we include "Maybe" in the set of allowed responses $q(h)$ for every q and h then then the worst-case gain of every learning action is 0. However, with our approach we can treat these responses as noise.

Supplementary material including the three websites and survey is available at http://ssli.ee.washington.edu/~guillory/icml11/.

References

Balcázar, J.L., Castro, J., Guijarro, D., Köbler, J., and Lindner, W. A general dimension for query learning. *Journal of Computer and System Sciences*, 73(6):924–940, 2007.

Bellala, G., Bhavnani, S., and Scott, C. Extensions of generalized binary search to group identification and exponential costs. In *Advances in Neural Information Processing Systems 23*, 2010.

Breese, J.S., Heckerman, D., Kadie, C., et al. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

Dasgupta, S., Lee, W.S., and Long, P.M. A theoretical analysis of query selection for collaborative filtering. *Machine Learning*, 51(3), 2003.

Golovin, D., Krause, A., and Ray, D. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems 23*, 2010.

Guillory, A. and Bilmes, J. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

Hanneke, S. The cost complexity of interactive learning, 2006. Unpublished.

Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knolwedge Discovery and Data Mining*, 2003.

Krause, A., McMahan, H.B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801, 2008.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, 7: 234–243, 1978.

Wolsey, L.A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4): 385–393, 1982.