Adaptive Submodularity with Varying Query Sets: An Application to Active Multi-label Learning

Alan Fern ALAN.FERN@OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University
Robby Goetschalckx ROBBY.GOETSCHALCKX@GMAIL.COM
Eduworks Corporation, Corvallis, Oregon, US
Mandana Hamidi-Haines HAMIDIM@OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University
Prasad Tadepalli TADEPALL@OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University

Editors: Steve Hanneke and Lev Reyzin

Abstract

Adaptive submodular optimization, where a sequence of items is selected adaptively to optimize a submodular function, has been found to have many applications from sensor placement to active learning. In the current paper, we extend this work to the setting of multiple queries at each time step, where the set of available queries is randomly constrained. A primary contribution of this paper is to prove the first near optimal approximation bound for a greedy policy in this setting. A natural application of this framework is to crowd-sourced active learning problem where the set of available experts and examples might vary randomly. We instantiate the new framework for multi-label learning and evaluate it in multiple benchmark domains with promising results.

1. Introduction

Adaptive submodularity has found many applications ranging from sensor placement to active learning. The goal is to optimize a submodular function, such as the expected information gathered by a set of sensors or a set of training examples, with a limited budget. Unlike the static setting of submodular optimization, the agent is allowed to be *adaptive* in the sense that each new selection could depend on the stochastic outcomes of previous selections.

The current approaches to adaptive submodular optimization (Golovin and Krause, 2011) assume that the set of possible choices is fixed for all of time. They provide a greedy algorithm with a near optimal approximation guarantee that exploits the adaptive submodularity of the objective function. However, in real-world applications such as crowd sourcing, all selections may not always be available. We formalize this problem through a set of constraints, where the constraints restrict the set of choices available to the agent. The constraints themselves can change randomly at each step. We address this problem of adaptive submodularity with varying queries by generalizing the framework of adaptive submodular optimization and provide a greedy algorithm with a near-optimal approximation guarantee.

An interesting application that motivates our research is active multi-label learning where an example has multiple labels, and each expert can only label a subset of them. A common scenario is crowd sourcing, where different workers are experts at labeling different classes, e.g., identifying

different species of plants or animals in pictures. Moreover, only some experts are available at any time. This motivates the following problem. Given a set of unlabeled instances, and a set of experts who randomly become available at a given time, how best to choose the next example to be labeled and by which expert?

We formalize this problem by generalizing the framework of adaptive submodular optimization in two orthogonal directions. First, following Chen and Krause (2013), we allow a *set* of queries to be asked at each time step, where the sets themselves are constrained. Second, we allow the constraints to vary at each step according to a stationary distribution. In multi-label learning, the query sets correspond to the set of labels that can be labeled by a single expert. At every time step, the system chooses an available query set and gets the answers back. The system refines its model based on the answers, and the cycle repeats. Importantly, the set of query sets available varies randomly according to a stationary probability distribution and is avaialble to the system only just in time for the query. The goal is to maximize the information obtained with a fixed query budget.

Our main contribution is to reduce this new setting to the standard adaptive submodularity setting, implying that a natural greedy adaptive policy is near-optimal. The key idea behind the proof is to view the greedy adaptive policy as an efficient implementation of a permutation policy, which orders all possible queries by a permutation, and then selects the first query in the ordering available at the current time step. The permutation policies are without loss of generality when the query set distribution is stationary and independent of the label distribution.

We empirically evaluate this framework on the active multi-label learning scenario descried above. There are two main lines of prior work on active multi-label learning. In the first line of work, all labels of a selected example are queried at once. Several heuristics for query selection have been explored, including uncertainty sampling (Brinker, 2005; Singh et al., 2009), minimizing the smallest margin of several one-vs-all classifiers (Tong and Koller, 2002), Max Loss and Mean Max Loss (MML) (Li et al., 2004), Maximum Loss Reduction with Maximum confidence (MMC) (Yang et al., 2009), and others. In the second line of work, each query consists of an example-label pair. Query selection heuristics include those based on uncertainty and diversity (Huang and Zhou, 2013; Wu et al., 2014), MML based on relationship between labels and samples (Li et al., 2004), and chi-square statistics (Ye et al., 2015). Our framework generalizes both of these lines of work by allowing queries over arbitrary subsets of labels, where the allowed subsets may vary randomly across time.

While most of the previous approaches are driven by heuristics with no performance guarantees, Vasisht et al. (2014) describes an approach with near-optimality guarantees on selecting batches of examples, for which all labels of those examples are queried. Importantly, this near-optimality guarantee is non-adpative and is closely tied to a gaussian process regression model. It also does not apply to our more general scenario where queries can be over arbitrary subsets of labels. To the best of our knowledge, our work is the first to provide such adaptive near-optimality guarantees in any multi-label setting.

2. Adaptive Submodular Optimization with Varying Query Sets

In this section we introduce a general framework for adaptive submodular optimization with varying queries, where we have a set of items in unknown states. The states can be queried subject to some exogenously chosen constraints that only appear at the time of the query. The goal is to find an adaptive query policy that opimizes a submodular objective function with a budget on the total

number of queries. Our framework naturally captures active multi-label learning via crowd sourcing, where each worker is an expert on a subset of the labels and is randomly available. The set of items correspond to the crross product of instances and labels. Each item can be in one of two states, *yes* and *no*. The query constraints C specify which sets of items can be queried at a given time. For example, we can allow different workers to be experts on different label (or instance) subsets, where the available experts varies randomly. Our analysis assumes that the subsets C at each step are drawn i.i.d. from an unknown distribution P_c, although the approach is applicable in other cases.

In this paper, we borrow the framework of adaptive submodular optimization (Golovin and Krause, 2011) and extend it to randomly varying queries and query sets. We then develop a greedy algorithm which has a near-optimal approximation guarantee by reducing it to the standard adaptive submodular optimization setting. The rest of this section reviews the previous work on adaptive submodularity and extends it to varying queries and query sets.

2.1. Adaptive Submodularity

We are given a finite set of items E and a finite set of observable states O. Each item $e \in E$ is associated with a state $o \in O$ through a function $\phi_o : E \rightarrow O$, which is called a *realization*. It is assumed that the realization ϕ_o is a random variable with known distribution $\mathbb{P}_o[\phi_o]$. Initially we are unaware of realization of the items, but they will be revealed to us via queries. Let ψ_o denote a *partial realization*, i.e., a subset of items and their observations, where $\mathrm{dom}(\psi_o)$ represents the items in ψ_o that have been observed. Moreover, we are given a utility function $f : 2^E \times O^E \rightarrow \Re \geq 0$, which maps a set of items and their observations to a real value.

The goal is to come up with a *query policy* π, or more simply a *policy*, that maximizes the expected utility under a total query budget. A query policy is a mapping from ψ_o to items in E, which specifies the element to query next given the history of observations. The policy π is executed by iteratively selecting the element suggested by π given the history of observations, and obtaining the observation for the selected element from the true realization.

Definition 1 *The* expected utility *of a policy in a partial realization ψ_o for a horizon (query budget) of l is $f_{avg}(\pi, \psi_o, l) \stackrel{def}{=} \mathbb{E}[f(E(\pi, \psi_o, l, \phi_o), \phi_o)]$, where $E(\pi, \psi_o, l, \phi_o)$ denotes the set of elements selected by π for l steps starting with the partial realization ψ_o when the true realization is ϕ_o. Let $f_{avg}(\pi, l) \stackrel{def}{=} f_{avg}(\pi, \{\}, l)$, where $\{\}$ is an empty partial realization.*

An l-horizon policy π^* is optimal if $f_{ave}(\pi^*, \psi_o, l)$ is at least as high as that of any other policy for all partial realizations $\psi_o \in \Psi$. The theory of Markov Decision Processes implies that optimal l-horizon policies always exist, although they depend on l and the distribution of ϕ (Puterman, 1994).

In general, the problem of computing an optimal policy is NP-hard. However, we can efficiently compute a near-optimal greedy policy when the utility function f satisfies a diminishing returns property. To make these notions more precise, we now summarize the adaptive submodular optimization framework of (Golovin and Krause, 2011).

Definition 2 *(Golovin and Krause, 2011) Given a partial realization ψ_o, the expected marginal benefit of an item e, $\Delta_f(e|\psi_o) \stackrel{def}{=} \mathbb{E}_{\phi_o \sim \psi_o}[f(\mathrm{dom}(\psi_o) \cup \{e\}, \phi_o) - f(\mathrm{dom}(\psi_o), \phi_o)]$ where ϕ_o is a random variable, $\phi_o \sim P[\phi_o|\psi_o]$.*

The expected marginal benefit $\Delta_f(e|\psi_o)$ gives the additional expected utility of an item e given the current partial realization ψ_o. Adaptive monotonicity requires that it is always non-negative.

Definition 3 *(Golovin and Krause, 2011) A function f is adaptive monotonic with respect to distribution $P_o[\phi_o]$ if the conditional expected marginal benefit of any query is nonnegative, i.e., for all ψ_o with $P_o[\psi_o]$ and all $e \in E$ we have $\Delta_f(e|\psi_o) \geq 0$.*

A partial realization ψ_o is said to be a *subrealization* of ψ'_o, written as $\psi_o \leq \psi'_o$, if its observed elements are a subset of those of ψ'_o, i.e., $dom(\psi_o) \subseteq dom(\psi'_o)$. Adaptive submodularity captures the diminishing returns property that the expected marginal benefit of a query never increases with more prior observations.

Definition 4 *(Golovin and Krause, 2011) A function f is adaptive submodular w.r.t. $P_o[\phi_o]$ if for all ψ_o and ψ'_o such that $\psi_o \leq \psi'_o$ and for all $e \in E$, we have $\Delta_f(e|\psi'_o) \leq \Delta_f(e|\psi_o)$.*

An approximate greedy policy picks the element which maximizes the expected marginal benefit of an item modulo a multiplicative approximation factor α.

Definition 5 *(Golovin and Krause, 2011) An* α-approximate greedy policy *for $\alpha \geq 1$ w.r.t. to the utility function f is a policy π which, for any partial realization ψ_o, picks an item $\pi(\psi_o)$ whose marginal expected benefit is $\geq \Delta_f(e|\psi_o)/\alpha$ for any other item e.*

The following theorem from (Golovin and Krause, 2011) shows that when the utility function is adaptive monotonic and adaptive submodular, an approximate greedy policy with a query budget l has a multiplicative approximation bound relative to an optimal policy with a query budget k. When $l = k$ and $\alpha = 1$, the greedy policy is $(1 - e^{-1})$-optimal.

Theorem 6 *((Golovin and Krause, 2011) Theorem 5.2.) If f is adaptive monotonic and adaptive submodular, then for any $\alpha-$approximate greedy policy π for $\alpha \geq 1$, optimal policies π^*, and positive integers l, k, $f_{avg}(\pi, l) \geq (1 - e^{\frac{-l}{\alpha k}}) f_{avg}(\pi^*, k)$.*

The next two sections generalize this model in two orthogonal directions. Section 2.2 extends it by randomly constraining the set of items to be selected. Section 2.3 allows multiple items to be queried at the same time. Finally, Section 2.4 combines the two extensions and allows querying sets of items, where the query sets are constrained randomly.

2.2. Extension to Varying Item Sets

We now introduce an extension of adaptive submodular optimization to a setting, where at each step the learner is constrained to select an item from an exogenously chosen random subset of items $C \subseteq E$. This naturally models problems such as crowd-sourcing where different workers have different expertise and not all of them are available all the time. We assume that the availability of items does not depend on their state distribution.

Assumption 1 The item set $C \in 2^E$ is i.i.d. according to a fixed but unknown distribution P_c, which is independent of P_ϕ.

Definition 7 *A valid policy π maps a partial realization ψ and item set C to an item in C.*

Definition 8 *The value of a valid policy in partial realization ψ for horizon l, $f_{c,ave}(\pi, \psi, l) \stackrel{def}{=} E_{\boldsymbol{C}} E_{\phi_o \sim \psi}[f(M(\pi, \phi_o, \boldsymbol{C}), \phi_o)]$ where $\boldsymbol{C} = C_1, \ldots, C_l$ is the vector of item sets available at steps $1, \ldots, l$, and $C_i \sim P_c$ i.i.d. and $M(\pi, \phi_o, \boldsymbol{C})$ represents the set of all items chosen until step l.*

An optimal l-horizon policy maximizes $f_{c,\psi,avg}$ for a query budget of l for all partial realizations ψ.

We seek to reduce this setting to the standard adaptive submodularity setting. The major difference in our setting is that the policy cannot choose an item until the available set of items C is known; so we cannot directly appeal to the results of the standard setting. To get around this limitation, instead of items we let the policies choose permutations over items. Given a set of allowed items, the first item in the set according to the permutation will be chosen. While there are an exponentially large number of permutations to consider, we will later show that such permutation policies can be implemented efficiently.

Let $U = \mathrm{Perm}(E) = \{\mu_1, \ldots, \mu_P\}$ represent all permutations over E. We implement the constraints via a random variable ϕ_c, which represents a mapping from permutations to subsets of E. $\phi_c : U \to 2^E$. The reason to make the constraints a function of the permutation is merely for book keeping and avoiding time indices. In fact we assume that they are independent of the permutation.

Assumption 2 The constraints are independent of μ, i.e., $Pr(\phi_c(\mu) = C) = P_c(C)$.

Definition 9 *A permutation policy is a mapping from the set of partial realizations Ψ to permutation set U. Given a permutation policy $\pi_{\mathrm{Perm}} : \Psi \to U$, the query policy it implements $\pi : \Psi \times 2^E \to E$ is such that $\pi(\psi, C) = \sigma(\pi_{\mathrm{Perm}}(\psi), C)$, where $\sigma(\mu, C)$ is the first element from μ to be in C.*

The following lemma shows that permutation policies do not lose optimality.

Lemma 10 *For every query policy there is a permutation policy that implements a query policy that is at least as good.*

Proof Let π^* be an optimal policy, which is not implementable by a permutation policy. This implies that there is a partial realization ψ, and two constraint sets C, C' with corresponding distinct optimal items e, e'. If $e \notin C'$, ordering e before e' gives a permutation policy that implements π^*. Similarly if $e' \notin C$, we can order it before e. If they are both in $C \bigcap C'$, since e is preferred by π^* when e' is also available in C, it results in at least as good a value as e'. Importantly, C or C' does not influence what occurs after e is chosen. Hence e is also optimal for C' and we can order e before e' for ψ. ■

The following defines an adaptive submodular and monotonic function V_f over the permutation sets, which represents the utility of applying all permutation policies in the set to select items. Lemma 12 then shows V_f is adaptive monotonic and adaptive submodular.

Definition 11 *For all $S_u \subseteq U$, $\sigma(S_u, \phi_c) \stackrel{\mathrm{def}}{=} \bigcup_{\mu \in S_u} \sigma(\mu, \phi_c(\mu))$; $V_f(S_u, \phi_o, \phi_c) \stackrel{\mathrm{def}}{=} f(\sigma(S_u, \phi_c), \phi_o)$; and $V_{f,ave}(\pi, l) \stackrel{\mathrm{def}}{=} f_{c,ave}(\pi, \{\}, l)$.*

Lemma 12 *If function f is adaptive monotonic and adaptive submodular with respect to P_o, then so is the function V_f with respect to P_o, P_c.*

Proof V_f is adaptive monotonic if $\Delta_{V_f}(\mu|\psi) \geq 0$. Let $\mathrm{dom}_U(\psi)$ and $\mathrm{dom}(\psi)$ represent respectively the set of permutations and the set of items queried in ψ. Let ψ be the composition of ψ_c and ψ_o, which represent the partial realizations over item sets and labels, respectively. We write $\psi = (\psi_c, \psi_o)$.

$$\begin{aligned}
\Delta_{V_f}(\mu|\psi_c,\psi_o) &= E_{\phi_c\sim\psi_c}E_{\phi_o\sim\psi_o}[V_f(\mathrm{dom}_U(\psi)\cup\{\mu\},\phi_o,\phi_c) - V_f(\mathrm{dom}_U(\psi),\phi_o,\phi_c)] && (1)\\
&= E_{\phi_c\sim\psi_c}E_{\phi_o\sim\psi_o}[f(\mathrm{dom}(\psi)+\sigma(\mu,\phi_c(\mu)),\phi_o) - f(\mathrm{dom}(\psi),\phi_o)] && (2)\\
&= E_{\phi_c\sim\psi_c}[\Delta_f(\sigma(\mu,\phi_c(\mu))|\psi_o)] && (3)\\
&= \sum_C Pr(\phi_c(\mu)=C|\psi_c)\Delta_f(\sigma(\mu,C)|\psi_o) && (4)\\
&= \sum_C P_c(C)\Delta_f(\sigma(\mu,C)|\psi_o) && (5)\\
&\geq 0 && (6)
\end{aligned}$$

Equation 2 employs the definition of V_f. Equation 3 follows from the definition of the marginal utility Δ_f and Assumption 1. Equation 5 follows from Assumption 2 and Equation 6 from adaptive monotonicity of f. Let $\psi = (\psi_c,\psi_o) \leq \psi' = (\psi'_c,\psi'_o)$. To show V_f is adaptive submodular we argue $\Delta_{V_f}(\mu|\psi) - \Delta_{V_f}(\mu|\psi') \geq 0$.

$$\begin{aligned}
\Delta_{V_f}(\mu|\psi) - \Delta_{V_f}(\mu|\psi') &= \sum_C P_c(C)\Delta_f(\sigma(\mu,c)|\psi_o) - \sum_C P_c(C)\Delta_f(\sigma(\mu,c)|\psi'_o) && (7)\\
&= \sum_C P_c(C)[(\Delta_f(\sigma(\mu,c)|\psi_o) - \Delta_f(\sigma(\mu,c)|\psi'_o)] && (8)\\
&\geq 0 && (9)
\end{aligned}$$

Equation 7 follows from Equation 5. Equation 9 follows from the adaptive submodularity of f since $\psi_o \leq \psi'_o$. ■

A direct adaptation of Theorem 6 to the varying items setting requires us to find a greedy permutation policy with respect to Δ_{v_f}. However, the space of all permutations is too big to search even for a greedy policy. Fortunately that is not necessary. Rather than first finding the greedy permutation μ for ψ and then selecting an item from C using $\sigma(\mu,C)$, we can greedily select the item in C with the most marginal utility. Since, by definition, no item in C has more marginal utility than the greedy choice, we can assert the following.

Lemma 13 *The marginal utility of the item selected by the greedy permutation policy from C is never more than that of any greedy query policy that selects an item with the most marginal utility among the items in C.*

We now state and prove an approximation result for submodular optimization with varying item sets.

Theorem 14 *If f is adaptive monotonic and adaptive submodular, then for any $\alpha-$approximate greedy query policy π for $\alpha \geq 1$, optimal query policy π^*, and positive integers l, k, $V_{f,avg}(\pi,l) \geq (1-e^{\frac{-l}{\alpha k}})V_{f,avg}(\pi^*,k)$.*

Proof From Lemma 12, if f is adaptive monotonic and adaptive submodular, so is V_f. From Lemma 13 the marginal utility of π is at least as high as that of any query policy implemented by a greedy permulation policy. From Theorem 6, the result follows when π^* is the query policy implemented by the best permutation policy. From Lemma 10, permutation policies can implement optimal policies without any loss. Hence the result follows. ■

2.3. Extension to Query Sets

We now extend the standard adaptive submodular optimization problem of Section 2.1 to a setting where multiple items are queried in each step. This is a straightforward adaptation of batch mode active learning setting of (Chen and Krause, 2013), where a small batch of k unlabeled examples are selected at each step. All labels of the selected batch are received in parallel, followed by the next batch of examples. In the current work, we let the queries be chosen from an arbitrary subset Q of 2^E rather than batches of size k. We seek a near optimal policy for picking those sets. We introduce a new function f^* that extends f to subsets of Q. $f^* : 2^Q \times O^E \rightarrow \Re \geq 0$ is defined as

$$\forall S \subseteq Q, f^*(S, \phi_o) = f(\bigcup_{q \in S} q, \phi_o). \tag{10}$$

Now, we define the expected marginal of a query as the expected improvement in the value of f^*.

$$\Delta_{f^*}(q|\psi_o) = \mathbb{E}_{\phi_o \sim \psi_o}[f^*(S \cup \{q\}, \phi_o) - f^*(S, \phi_o)]$$

where $S = dom(\psi_o)$. We similarly extend the definition of f_{avg} of a policy to define f^*_{avg} of a l-horizon policy π as the expected value of f^* when choosing the queries according to π with a query budget of l. The following lemma is a straightforward extension of Theorem 1 of (Chen and Krause, 2013) which was restricted to batch mode active learning. The proof is included in the Appendix for completeness.

Lemma 15 *If the function f is adaptive monotonic and adaptive submodular with respect to P_o, then so is the extended function f^*.*

Lemma 15 and Theorem 6 imply the following bound.

Theorem 16 *If f is adaptive monotonic and adaptive submodular, then for any $\alpha-$approximate greedy query set policy π for $\alpha \geq 1$, optimal query set policy π^*, and positive integers l, k, $f^*_{avg}(\pi, l) \geq (1 - e^{\frac{-l}{\alpha k}}) f^*_{avg}(\pi^*, k)$.*

The implication of the above theorem is that the greedy algorithm with respect to the marginal utility of query sets is approximately optimal.

2.4. Extension to Varying Query Sets

We now combine the extensions of Sections 2.2 and 2.3 into a new setting of Adaptive Submodularity with Varying Query Sets. This combines the idea of randomly constraining the available queries with the idea of simultaneously querying multiple items. Thus, the learner is now constrained to choose from an exogenously picked random subset of query sets $C \subseteq Q \subseteq 2^E$.

Given the base utility function f, we define a new utility function V_{f^*} as the expected utility of a permutation policy over query sets, which are constrained randomly according to the distribution P_c.

Lemma 17 *If function f is adaptive monotonic and adaptive submodular with respect to P_o, then so is the function V_{f^*} with respect to P_o, P_c.*

Proof From Lemma 15, since f is adaptive monotonic and adaptive submodular, so is f^*. From Lemma 12, since f^* is adaptive monotonic and adaptive submodular, so is V_{f^*}. ■

We now state our main theorem which gives an approximation bound on the performance of a greedy policy over query sets.

Theorem 18 *If f is adaptive monotonic and adaptive submodular, then for any $\alpha-$approximate greedy query policy π for $\alpha \geq 1$, optimal query policy π^*, and positive integers l, k, $V_{f^*,avg}(\pi, l) \geq (1 - e^{\frac{-l}{\alpha k}})V_{f^*,avg}(\pi^*, k)$.*

Proof The proof follows directly by composing Theorem 14 and Theorem 16. ■

An important thing to note is that although the greedy heuristic does not assume the knowledge of the constraint distribution P_c, it is competitive with the optimal query policy π^* that *does have* this knowledge. While this might appear counter-intuitive, the reason that it works is that P_c is a stationary distribution where the constraints are i.i.d. Thus, the optimal policy cannot afford to sacrifice current gain in selecting a good example in the hope that it can plan for it in the future. The diminishing returns property of the utility function implies that there is a price to pay in delaying the reward, which plays a central role in the proof of Theorem 6 and is inherited by the other theorems.

We close this section by describing a generic greedy algorithm for adaptive sub-modular optimization with varying query sets. The pseudo-code is presented in Algorithm 1. The input of the algorithm is a set of all possible queries Q, adaptive submodular function f, and budget l. It initializes the probabilistic model with a prior. It then repeats for l iterations, where in each iteration, it first observes the set of available queries C_t at time t (Step 4). The algorithm uses an evaluation function that estimates the incremental value of the query defined by the function Δ_f (Step 5). The query results in a set of observations which are added to the partial realization (Step 7) and they are used to update the model in Step 8, and the cycle repeats.

Algorithm 1 Adaptive Submodular Optimization with Varying Query Sets

1: **Input:** Query set Q ; function f; query budget l.
2: Initialize the model, and set $\psi \leftarrow \emptyset$
3: **for** $t = 1$ to l **do**
4: Let C_t be the set of available queries
5: $q^* = \underset{q \in C}{argmax}\, [\Delta_f(q|\psi)]$
6: $\Phi(q*)$ = Observe the states of the set of items of q^*
7: $\psi \leftarrow \psi \cup \{\langle q^*, \Phi(q*)\}$
8: Update the model given ψ
9: **end for**

3. Application: Active Multi-Label Learning with Varying Experts

Here we discuss the active multi-label learning application that motivates our research. We have a set of unlabeled instances X, where each instance can have multiple labels form the set of possible labels Y. We have a set of experts each of whom can only annotate a subset of labels for instances.

Moreover, we have a set of all possible queries $Q \subseteq X \times 2^Y$ and a set of available queries $C_t \subseteq Q$ at a given time step. Each query corresponds to an example-expert pair, where an expert is identified with a subset of the labels, $\mathbf{y} \subseteq Y$. Each query $(x, \mathbf{y}) \in C_t$ results in determining which of the labels in $\mathbf{y}$ are present for x. The goal of multi-label learning algorithm is to adaptively select a sequence of l queries that optimizes some objective that measures the information obtained from the experts.

We now describe the greedy algorithm for active multi-label learning with varying queries. The inputs to the algorithm are a set of unlabeled examples and a set of experts, each of whom is indicated by a subset of the labels. In this algorithm we need a model to compute the joint posterior probability distribution of the labels which is intimately involved in computing the marginal utility function. In our experiments we assumed that the labels are independent and used logistic regression to model the conditional posterior distributions of each label given the instance.

The algorithm initializes m logistic regression weight vectors, i.e., one W_y for each label $y \in Y$ (line 2 of Algorithm 2). These weights are used for computing posterior probability of each classifier $P_\psi(O_{x,y}) = 1/(1 + e^{-W_y^T f(x)})$, where $f(x)$ is the feature vector of x, and $O_{x,y}$ is the observation of label y for instance x. At each time step, the available example-expert pairs C_t is given (line 4). The algorithm computes the marginal benefit of an example-expert pairs $(x, \mathbf{y}) \in C_t$ and picks a pair $(x^*, \mathbf{y}^*)$, whose marginal benefit is the highest (line 5). We applied maximum *Gibbs error* criterion, explained in Section 3.2, to implement the greedy policy. Next, the algorithm asks the expert to annotate the vector of the subset of labels $\mathbf{y}^*$ for x, which is denoted by $\mathbf{O}_{x^*,\mathbf{y}^*}$ (line 6). Finally, the algorithm updates its partial realization ψ (line 7) and the set of classifiers based on new data (line 8).

Algorithm 2 Active Multi-Label Learning with Varying Experts

1: **Input:** unlabeled data X; utility function f
2: Initialize the logistic regression weight vector W_y for each label y, and set $\psi \leftarrow \emptyset$
3: **for** $t = 1$ to l **do**
4: Let C_t be the set of available example-query set pairs
5: $\langle x^*, \mathbf{y}^* \rangle = \underset{\langle x,\mathbf{y}\rangle \in C_t}{argmax} \left[\Delta_f(\mathbf{O}_{x,\mathbf{y}}|\psi)\right]$
6: $\mathbf{O}_{x^*,\mathbf{y}^*}$ = The observations for $\langle x^*, \mathbf{y}^* \rangle$
7: $\psi \leftarrow \psi \cup \{\langle x^*, \mathbf{O}_{x^*,\mathbf{y}^*} \rangle\}$
8: Update the weights W_y for each label y given ψ
9: **end for**

The following two sections analyze two popular greedy heuristics for active learning, namely, the maximum entropy criterion and the maximum Gibbs error criterion. Although these two criteria are quite different in general, and only the maximum Gibbs error criterion is provably near-optimal, they are equivalent for binary label classification and behave identically (Cuong et al., 2013, 2014).

3.1. Maximum Entropy Criterion

The maximum entropy criterion selects the next example whose posterior label distribution has the maximum Shannon entropy (Dagan and Engelson, 1995). Although it is a popular heuristic, as we show below, it does not satisfy adaptive submodularity. Our counterexample is based on Cuong's thesis (Cuong, 2015) (Cuong et al., 2014) where it is shown that the maximum entropy heuristic does not yield a multiplicative approximation bound.

Example. Consider that there are $n+3$ items including three special items one *id* and two *decoys*. The other items are labeled $X_1, \ldots, X_n$, one of which is an unknown *target*. All items $X_1, \ldots, X_n$ except the target have the same label 0. The target has a label of $\log m$ bits. The label of *id* has $\log n$ bits and represents the index of the target item. The labels of the two decoy items are $\log(n+1)$ bits each and are independent of each other and every other label. We denote partial realizations as sets of item-label pairs, with $\{\}$ representing the initial empty partial realization.

$$\Delta_f(id|\{\}) = H(id) = \log n \tag{11}$$
$$\Delta_f(decoy|\{\}) = H(decoy) = \log(n+1) \tag{12}$$
$$\Delta_f(X_p|\{\}) = H(X_p|\{\}) = \frac{1}{n}\log mn + \frac{n-1}{n}\log\frac{n}{n-1} \tag{13}$$
$$= \frac{1}{n}\log m + \log n - \frac{n-1}{n}\log(n-1) \tag{14}$$
$$\Delta_f(X_p|\{(id,q)\}, p \neq q) = 0 \tag{15}$$
$$\Delta_f(X_p|\{(id,p)\}) = H(X_p|\{(id,p)\}) = \log m \tag{16}$$

Equation 14 is the entropy of the items X_p. Equation 15 follows because the non-target items do not have any information. Finally Equation 16 is true because there are $\log m$ bits revealed from the target. We note that our definitions of adaptive monotonicity is satisfied for all items. To satisfy adaptive submodularity, we need:

$$\Delta_f(X_p|\{\}) \geq \Delta_f(X_p|\{(id,p)\}) \Leftrightarrow \frac{1}{n}\log m + \log n - \frac{n-1}{n}\log(n-1) \geq \log m \tag{17}$$
$$\Leftrightarrow \frac{n}{n-1}\log n - \log(n-1) \geq \log m \tag{18}$$

As n approaches ∞ the left hand side of this equation tends to 0, and fails to satisfy the above equation when $m \geq 2$.

3.2. Maximum Gibbs Error Criterion

This criterion selects the next query q^* whose posterior label distribution has the maximum Gibbs error or minimal negative Gibbs error: $q^* = \text{argmin}_q \sum_{o \in O} p_\psi(y = o|q)^2$. Gibbs error GE is the expected error of the Gibbs classifier, which samples a label from the posterior label distribution, p_ψ, and uses it for prediction. $GE(y) = 1 - \sum_{o \in O} p_\psi(y = o)^2$ where $o \in O$ is the set of all possible values (states) for label y.

Policy Gibbs error is the expected error rate of a Gibbs classifier on the set adaptively selected by the policy. In (Cuong et al., 2014), it is shown that the policy Gibbs error corresponds to the expected reduction in the volume of the version space . Since the expected version space reduction is adaptive monotonic and adaptive submodular (Golovin and Krause, 2011), choosing an item that maximzies the reduction in the expected policy Gibbs error would lead to a near-optimal policy.

In our active multi-label learning algorithm each query, q, consist of a set of k labels, $y_1, \ldots, y_k$. Thus, we compute the maximum Gibbs error with respect to the joint posterior distribution of a vector of labels in the query. So we have $q^* = \text{argmin}_q \sum_{o_1,\ldots,o_k \in O} p_\psi(y_1 = o_1, ..., y_k = o_k|q))^2$. While the above computation requires summing exponentially many terms in general,it simplifies to a polynomial-time computation when the different predictors are indepedent as in our multi-label

learning experiments. Hence, in this case, the next example q^* can be computed efficiently as:

$$q^* = \operatorname*{argmin}_q \prod_{i=1}^{k} \sum_{o_i \in O} p_\psi(y_i = o_i|q)^2.$$

4. Empirical Evaluation

We now describe an experimental evaluation of our algorithm in the context of multi-label learning. We simulate the crowd-sourcing scenario by assigning each expert a subset of the labels. We compare 3 versions of our algorithm: selecting an example and an expert according to Gibbs error criterion, selecting the example using Gibbs error criterion and the expert randomly among those available, and selecting both the example and the expert randomly. We evaluate the methods on six benchmark datasets (see Table 1), selected these dataset based on diversity of domains and their popularity within the multi-label learning community [1].

Table 1: Characteristics of the datasets.

Data set	Domain	Instance	Features	Labels
Emotions	music	593	72	6
Scene	image	2407	294	6
Flags	image	194	19	7
Yeast	biology	2417	103	14
Mediamill	video	43907	120	101
CAL500	music	502	68	174

4.1. Experimental Setup

We assume that all examples are always available, but only a subset of experts are available at a time. Each expert is identified with a subset of labels. Thus each query, i.e., example-expert pair, consists of an example and a set of labels that the expert can label. The number of labels L for each domain represents the importance of expert selection and is a relavant parameter. For each experiment the sizes of the label sets of all experts are equal. We call this the *query size* s, which is another important control parameter. To represent the expertise of different experts, we repeatedly generate random label sets of a given query size and add them to the pool until all labels are covered, rejecting a label set if it has been previously generated. We sample the labels with replacement, so the label sets for experts can overlap. We compare the following 3 schemes for query selection.

1) ***Random:*** Randomly selects an available example-expert pair, such that there is at least one new label that has not been previously queried on that example.

2) ***GibbsEx:*** Randomly selects an available expert and then chooses an example that has the maximum Gibbs error for that expert.

1. `http://mulan.sourceforge.net/datasets.html`

3) *GibbsExExp:* Selects an example-expert pair which has the maximum Gibbs error among all available pairs.

We evaluate the algorithms by their *accuracy* on the test data. We normalized the datasets to make the values of each feature have zero-mean and unit-variance. We used the binary-class logistic regression classifiers with $L2$ regularization and tuned the regularization parameter α via 5-fold cross validation. We computed the averaged result over 10 runs of 5-fold cross validation.

4.2. Experimental Results

Figure 1 shows the results for all the domains for different query sizes. Each row of plots represents a different domain. The number of labels increase as we go down from top to bottom. For each domain, the query size increases as we go from left to right. Each plot shows the accuracy on the test data as a function of the number of training queries.

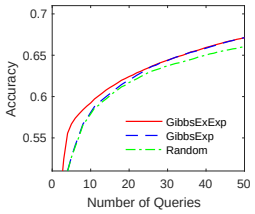
We can make a number of observations from these results. First, as one would expect, increasing the query size (i.e. compare Figure 1(m) against Figure 1(o)), would increase the accuracies of all the algorithms as the learners get more information from each query. Second, the differences between *GibbsExExp* and the other methods are more prominent as the number of labels L is increased, i.e., as we go down from top to bottom in the plots. This is especially true when the query size s is small because many experts are typically needed to label each example in these cases and it becomes important to select experts smartly. Third, when query size is small, *GibbsExExp* performs comparably and often better than *GibbsEx* and *Random*. In experiments with small query sizes, more experts are needed to label the data. Hence, *GibbsExExp* which has a better heuristic in selecting experts, has a higher accuracy than *Random* and *GibbsEx*, which select experts randomly. By increasing the query size the gap between *GibbsExExp* and *GibbsEx* reduces, while the gap between *GibbsEx* and *Random* increases. With increased query size, expert selection becomes less important than example selection because each expert can now label many labels. This behavior can be observed in all datasets, especially in datasets with large label sizes (e.g. *Mediamil* and *CAL500*).

5. Conclusions

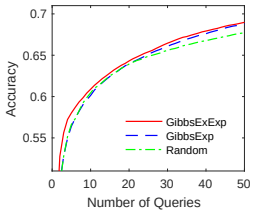
In this work we extended the framework of adpative submodular optimization to a setting where the available queries are randomly constrained and gave a simple greedy algorithm with a near-optimal performance bound. We applied the new framework to the problem of multi-label learning and showed promising results based on the Gibbs error heuristic. One new problem that arises in the crowd sourcing setting is to simultaneously learn the expertise of different workers. It would also be interesting to consider further generalizations of our framework and other potential applications.

6. Acknowledgements

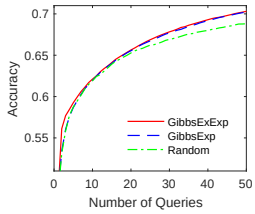
The authors acknowledge the support of grants from NSF (grant no. IIS-1619433), ONR (grant no. N00014-11-1-0106), and DARPA (grant no. DARPA N66001-17-2-4030).



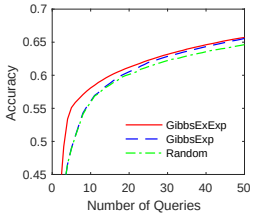
(a) Emotions, $s = 2$, $L = 6$



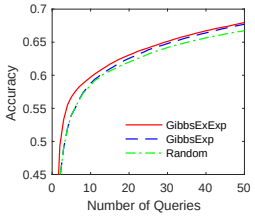
(b) Emotions, $s = 3$, $L = 6$



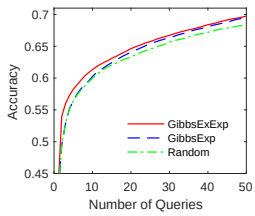
(c) Emotions, $s = 4$, $L = 6$



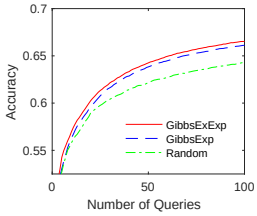
(d) Scene, $s = 2$, $L = 6$



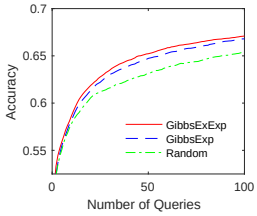
(e) Scene, $s = 3$, $L = 6$



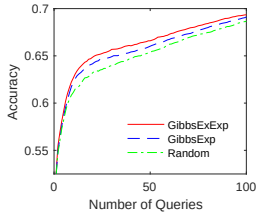
(f) Scene, $s = 4$, $L = 6$



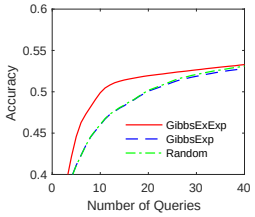
(g) Flags, $s = 2$, $L = 7$



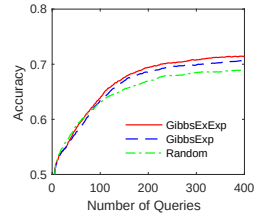
(h) Flags, $s = 3$, $L = 7$



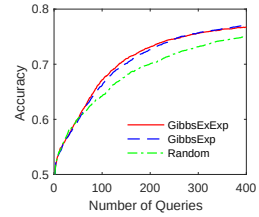
(i) Flags, $s = 6$, $L = 7$



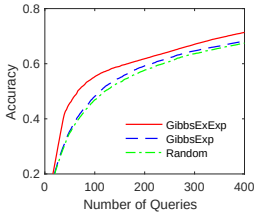
(j) Yeast, $s = 2$, $L = 14$



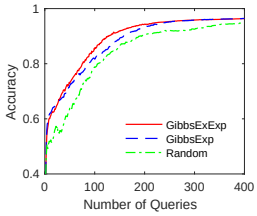
(k) Yeast, $s = 7$, $L = 14$



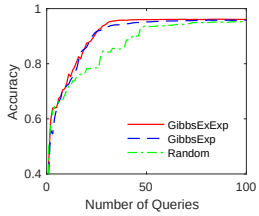
(l) Yeast, $s = 11$, $L = 14$



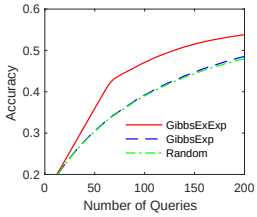
(m) Mediamill, $s = 2$, $L = 101$



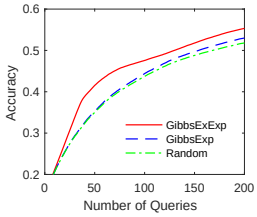
(n) Mediamill, $s = 50$, $L = 101$



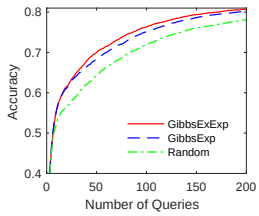
(o) Mediamill, $s = 80$, $L = 101$



(p) CAL500, $s = 2$, $L = 174$



(q) CAL500, $s = 3$, $L = 174$



(r) CAL500, $s = 50$, $L = 174$

Figure 1: The average results over 10 runs on six datasets with query size s and label set size L.

Appendix A. Proof of Lemma 15

Proof Let $S = \mathrm{dom}(\psi_o)$, $q_i = \{e_1, \ldots, e_n\}$, $R = \bigcup_{q \in S} q$, and $\psi_o \leq \psi'_o$. Then, adaptive monotonicity follows from:

$$
\begin{aligned}
\Delta_{f^*}(q|\psi_o) &= \mathbb{E}_{\phi_o \sim \psi_o}[f^*(S \cup \{q\}, \phi_o) - f^*(S, \phi_o)] && (19)\\
&= \mathbb{E}_{\phi_o \sim \psi_o}[f(\bigcup_{q \in S} q \cup \{q_i\}, \phi_o) - f(\bigcup_{q \in S} q, \phi_o)] && (20)\\
&= \mathbb{E}_{\phi_o \sim \psi_o}[f(R \cup \{e_1, \ldots, e_n\}, \phi_o) - f(R, \phi_o)] && (21)\\
&= \mathbb{E}_{\phi_o \sim \psi_o}[f(R \cup \{e_1, \ldots, e_n\}, \phi_o) - f(R \cup \{e_2, \ldots, e_n\}, \phi_o)\\
&\quad + f(R \cup \{e_2, \ldots, e_n\}, \phi_o) - f(R \cup \{e_3, \ldots, e_n\}, \phi_o)\\
&\quad + \ldots + f(R \cup \{e_n\}, \phi_o) - f(R, \phi_o)] && (22)\\
&= \mathbb{E}_{\phi_o \sim \psi_o}[f(R \cup \{e_1, \ldots, e_n\}, \phi_o) - f(R \cup \{e_2, \ldots, e_n\}, \phi_o)]\\
&\quad + \ldots + \mathbb{E}_{\phi_o \sim \psi_o}[f(R \cup \{e_n\}, \psi_o) - f(R.\phi_o)] && (23)\\
&= \Delta_f(e_1|\psi_o \cup R \cup \{e_2, \ldots, e_n\}) + \Delta_f(e_2|\psi_o \cup R \cup \{e_3, \ldots, e_n\})\\
&\quad + \ldots + \Delta_f(e_n|\psi_o)\\
&\geq 0 && (24)
\end{aligned}
$$

Equations 22, 23 and 24 follow from telescoping, the additivity of expectations, and the adaptive monotonicity of f respectively. Adaptive submodularity is shown below, where S= dom (ψ), S' = dom(ψ'), $R = \bigcup_{q \in S} q$, and $R' = \bigcup_{q \in S'} q$.

$$
\begin{aligned}
\Delta_{f^*}(q_i|\psi) - \Delta_{f^*}(q_i|\psi') &= \mathbb{E}_{\phi \sim \psi}[f^*(S \cup \{q_i\}, \phi) - f^*(S, \phi)] - \mathbb{E}_{\phi' \sim \psi'}[f^*(S' \cup \{q_i\}, \phi') - f^*(S', \phi')]\\
&= \mathbb{E}_{\phi \sim \psi}[f(\bigcup_{q \in S} q \cup \{e_1, \ldots, e_n\}, \phi) - f(\bigcup_{q \in S} q, \phi)]\\
&\quad - \mathbb{E}_{\phi' \sim \psi'}[f(\bigcup_{q \in S'} q \cup \{e_1, \ldots, e_n\}, \phi') - f(\bigcup_{q \in S'} q, \phi')] && (25)\\
&= \mathbb{E}_{\phi \sim \psi}[f(R \cup \{e_1, \ldots, e_n\}, \phi) - f(R, \phi)]\\
&\quad - \mathbb{E}_{\phi' \sim \psi'}[f(R' \cup \{e_1, \ldots, e_n\}, \phi') - f(R', \phi')] && (26)\\
&= \mathbb{E}_{\phi \sim \psi}[f(R \cup \{e_1, \ldots, e_n\}, \phi) - f(R \cup \{e_2, \ldots e_n\}, \phi)\\
&\quad + f(R \cup \{e_2, \ldots e_n\}, \phi) - f(R \cup \{e_3, \ldots e_n\}, \phi)\\
&\quad + \ldots +\\
&\quad + f(R \cup \{e_n\}, \phi) - f(R, \phi)]\\
&\quad - \mathbb{E}_{\phi' \sim \psi'}[f(R' \cup \{e_1, \ldots, e_n\}, \phi') - f(R' \cup \{e_2, \ldots e_n\}, \phi')\\
&\quad + f(R' \cup \{e_2, \ldots e_n\}, \phi') - f(R' \cup \{e_3, \ldots e_n\}, \phi')\\
&\quad + \ldots +\\
&\quad + f(R' \cup \{e_n\}, \phi') - f(R', \phi')] && (27)
\end{aligned}
$$

Equation 25 expands the definition of f^*, Equation 26 uses the definitions of R and R', and Equations 27 follows from telescoping the terms. We now rearrange the terms on the right hand side.

$$
\begin{aligned}
\Delta_{f^*}(q_i|\psi) - \Delta_{f^*}(q_i|\psi') &= \mathbb{E}_{\phi\sim\psi}[f(R\cup\{e_1,\ldots,e_n\},\phi) - f(R\cup\{e_2,\ldots e_n\},\phi)] \\
&\quad -\mathbb{E}_{\phi'\sim\psi'}[f(R'\cup\{e_1,\ldots,e_n\},\phi') - f(R'\cup\{e_2,\ldots e_n\},\phi')] \\
&\quad +\mathbb{E}_{\phi\sim\psi}[f(R\cup\{e_2,\ldots e_n\},\phi) - f(R\cup\{e_3,\ldots e_n\},\phi)] \\
&\quad -\mathbb{E}_{\phi'\sim\psi'}[f(R'\cup\{e_2,\ldots,e_n\},\phi') - f(R'\cup\{e_3,\ldots e_n\},\phi')] \\
&\quad +\ldots+ \\
&\quad +\mathbb{E}_{\phi\sim\psi}[f(R\cup\{e_n\},\phi) - f(R,\phi)] \\
&\quad -\mathbb{E}_{\phi'\sim\psi'}[f(R'\cup\{e_n\},\phi') - f(R',\phi')] \qquad (28) \\
&= \Delta_f(e_1|R\cup\{e_2,\ldots,e_n\}|\psi) - \Delta_f(e_1|R'\cup\{e_2,\ldots,e_n\}|\psi') \\
&\quad +\Delta_f(e_2|R\cup\{e_3,\ldots,e_n\}|\psi) - \Delta_f(e_2|R'\cup\{e_3,\ldots,e_n\}|\psi') \\
&\quad +\ldots+ \\
&\quad +\Delta_f(e_n|\psi) - \Delta_f(e_n|\psi') \qquad (29) \\
&\geq 0 \qquad (30)
\end{aligned}
$$

Equations 29 follows from the definition of Δ_f and Equation 30 from the submodularity of f. ■

References

K. Brinker. On active learning in multi-label classification. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nrnberger, and W. Gaul, editors, *In From Data and Information Analysis to Knowledge Engineering*, pages 206–213. Springer Berlin Heidelberg, 2005.

Y. Chen and A. Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 160–168. JMLR Workshop and Conference Proceedings, 2013.

N. V. Cuong. *Near-optimality and robustness of greedy algorithms for Bayesian pool-based active learning*. PhD dissertation, National University of Singapore, 2015.

N. V. Cuong, W. Sun Lee, N. Ye, K. Ming Adam Chai, and H. L. Chieu. Active learning for probabilistic hypotheses using the maximum gibbs error criterion. In *Advances in Neural Information Processing Systems.*, pages 1457–1465, 2013.

N. V. Cuong, W. Sun Lee, and N. Ye. Near-optimal adaptive pool-based active learning with general loss. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 122–131, 2014.

I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann, 1995.

D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research (JAIR)*, 42:427–486, 2011.

S.J. Huang and Z.H. Zhou. Active query driven by uncertainty and diversity for incremental multi-label learning. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1079–1084, 2013.

X. Li, L. Wang, and E. Sung. Multilabel svm active learning for image classification. In *2004 International Conference on Image Processing (ICIP '04)*, volume 4, pages 2207–2210, 2004.

M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.

M. Singh, E. Curran, and P. Cunningham. Active learning for multi-label image annotation. Technical report, 2009.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002.

D. Vasisht, A. Damianou, M. Varma, and A. Kapoor. Active learning for sparse bayesian multilabel classification. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 472–481. ACM Association for Computing Machinery, 2014.

J. Wu, V.S. Sheng, J. Zhang, P. Zhao, and Z. Cui. Multi-label active learning for image classification. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5227–5231, 2014.

B. Yang, J. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 917–926, New York, NY, USA, 2009. ACM.

C. Ye, J. Wu, V. S. Sheng, S. Zhao, P. Zhao, and Z. Cui. Multi-label active learning with chi-square statistics for image classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 583–586, 2015.