Near Optimal Bayesian Active Learning for Decision Making

Shervin Javdani
Carnegie Mellon University

Yuxin Chen
ETH Zürich

Amin Karbasi
ETH Zürich

Andreas Krause
ETH Zürich

J. Andrew Bagnell
Carnegie Mellon University

Siddhartha Srinivasa
Carnegie Mellon University

Abstract

How should we gather information to make effective decisions? We address Bayesian active learning and experimental design problems, where we sequentially select tests to reduce uncertainty about a set of hypotheses. Instead of minimizing uncertainty per se, we consider a set of overlapping *decision regions* of these hypotheses. Our goal is to drive uncertainty into a single decision region as quickly as possible.

We identify necessary and sufficient conditions for correctly identifying a decision region that contains all hypotheses consistent with observations. We develop a novel *Hyperedge Cutting* (HEC) algorithm for this problem, and prove that is competitive with the intractable optimal policy. Our efficient implementation of the algorithm relies on computing subsets of the complete homogeneous symmetric polynomials. Finally, we demonstrate its effectiveness on two practical applications: approximate comparison-based learning and active localization using a robot manipulator.

1 Introduction

Bayesian active learning addresses the problem of selecting a sequence of experiments, or *tests*, to determine a hypothesis consistent with observations. This fundamental problem arises in a wide range of applications such as medical procedures, content search, and robotics. It has been studied in several domains, including machine learning (Dasgupta, 2004; Balcan et al., 2006; Nowak, 2009), statistics (Lindley, 1956; Chaloner and Verdinelli, 1995), decision theory (Howard, 1966), and others.

For instance, in *automated medical diagnosis* (Kononenko, 2001) we are presented with hypotheses about the state of a patient, and select medical tests to infer their illness. In *comparison-based learning* (Goyal et al., 2008; Karbasi et al., 2012), we infer a target in a database by sequentially presenting a user with pairs of candidates, and having the user select which is closer. In robotic *active localization*, the robot attempts to identify its own or an object's location by probing, e.g., with touch or vision (Fox et al., 1998; Kollar and Roy, 2008; Hsiao et al., 2008; Javdani et al., 2013). In general, the goal is to gather the necessary information while minimizing test cost.

In this paper, we develop a general framework for addressing these problems. Instead of indiscriminately minimizing uncertainty about hypotheses directly, we aim to reduce uncertainty in a structured way to facilitate *decision making*. We suppose the hypothesis space is covered by a set of decision regions: Each region identifies the set of hypotheses for which it would succeed. Our goal is to select tests that quickly concentrate all consistent hypotheses in a single decision region.

Special cases of this general problem have been studied. In the so called *Optimal Decision Tree* (ODT) problem, each decision region corresponds to a single hypothesis. In this case, a greedy algorithm called *Generalized Binary Search* (GBS) is known to perform near optimally, i.e., the expected number of observations is $O(\log n)$ more than the optimum policy where n indicates the number of hypotheses (Dasgupta, 2004; Guillory and Bilmes, 2009; Kosaraju and Borgstrom, 1999). GBS greedily selects tests in expectation over the test outcomes to maximize the probability mass of eliminated hypotheses. Another special instance of our setting is the *Equivalence Class Determination* (ECD) problem (Golovin et al., 2010) where the set of hypotheses is (disjointly) partitioned– that is, decision

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

regions do not overlap and collectively cover the set of hypotheses. In this case, it is known that GBS performs poorly while greedily optimizing a more informative objective known as EC2 exhibits an $O(\log n)$ approximation guarantee (Golovin et al., 2010).

In both aforementioned settings, decision regions are *disjoint*. In this paper, we tackle the general case of *overlapping decision regions*, a problem that is less understood. We develop a novel surrogate objective function, which we call *Hyperedge Cutting* (HEC), and prove that the policy which greedily maximizes this objective has strong theoretical guarantees. It relies on the fact that our proposed objective function satisfies *adaptive submodularity* (Golovin and Krause, 2011), a natural diminishing returns property that generalizes the classical notion of submodularity to policies.

We empirically evaluate our algorithm on two applications: approximate comparison-based learning (Karbasi et al., 2012), and active localization with a robot hand. In approximate comparison-based learning, a user is searching through set of items (e.g., movies), and is not particularly interested in a single item, but rather any suggestion from a given category (e.g., the horror genre). The search terminates once all items consistent with user responses are contained in a single category. Similarly, many actions in robotic manipulation, such as pushing a button or grasping an object, inherently tolerate some uncertainty. The robot need not know the exact location of an object, but rather must localize an object to a decision region to ensure it can successfully accomplish the task. An optimal policy achieves each of these with the smallest test cost.

We make the following contributions:

1. We provide a necessary and sufficient condition for identifying if a decision region contains all hypotheses that are consistent with the tests performed.
2. We develop a novel algorithm – *Hyperedge Cutting* (HEC) – and prove that it is competitive with the intractable optimal algorithm.
3. We provide an efficient way to implement our algorithm based on computing sums of the complete homogeneous symmetric polynomials.
4. We demonstrate the empirical effectiveness of our approach for both comparison-based learning and active localization in a robotic manipulation task.

2 Problem Statement

We formalize our Bayesian active learning problem by assuming a prior probability distribution P on a set of hypotheses $\mathcal{H}$ (e.g., state of patient, location of target). By conducting tests from a set of tests $\mathcal{T}$, we gain information about the true, initially unknown hypothesis.

More formally, for a given hypothesis $h \in \mathcal{H}$, running a test $t \in \mathcal{T}$ produces an outcome (deterministically) from a finite set of outcomes/observations $\mathcal{O}$. Thus, each hypothesis $h \in \mathcal{H}$ can be considered a function $h : \mathcal{T} \to \mathcal{O}$ mapping tests to outcomes. Suppose we have executed a set of tests $T = \{t_1, \ldots, t_m\} \subseteq \mathcal{T}$ (e.g., medical tests we ran, items shown to the user, moves made by the robot), and have observed their outcomes $h(t_1), \ldots, h(t_m)$. Our evidence so far is captured by a set of test-outcome pairs, $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$, where $\mathcal{S} = \{(t_1, h(t_1)), \ldots, (t_m, h(t_m))\}$.

Upon observing $\mathcal{S}$, we can rule out hypotheses inconsistent with our observations. We denote the resulting set of hypotheses by

$$\mathcal{V}(\mathcal{S}) = \{h \in \mathcal{H} : \forall (t, o) \in \mathcal{S}, h(t) = o\} \qquad (1)$$

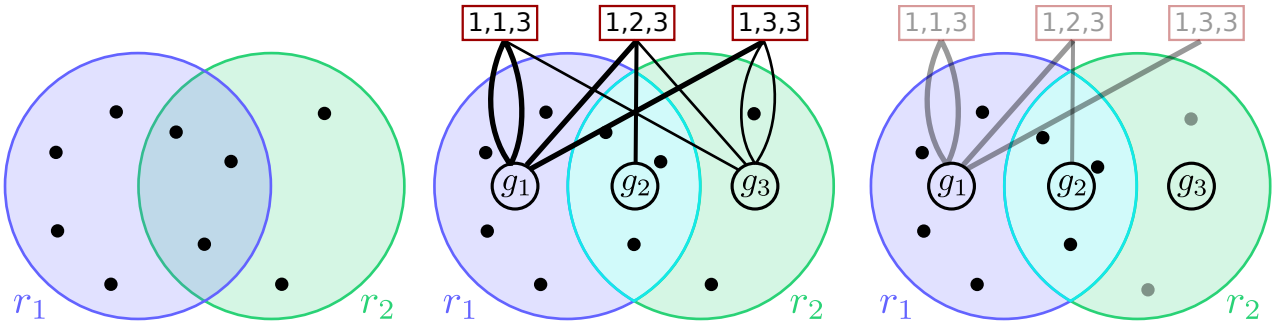
In principle, we can now choose tests that reduce our uncertainty about the set of hypotheses directly. In many practical problems, we are primarily concerned about *reducing uncertainty for the purpose of making a decision*: it is not necessary to remove all uncertainty, but it is necessary to reduce uncertainty in a structured way to ensure a decision action will be successful. Choosing tests that reduce uncertainty dramatically, but still leave it unclear what action to choose, will not be effective. We now formalize this idea.

Active learning for decision making. Suppose we have a set of decisions $\mathcal{R}$, and the eventual goal of selecting a decision $r \in \mathcal{R}$ after gathering information. For example, in medical diagnosis, we choose a treatment; in robotic manipulation, we press a button (Fig. 5); in content search, we recommend a particular movie.

Each decision region r corresponds to the set of hypotheses for which it would succeed, i.e., $r \subseteq \mathcal{H}$. Our problem is then captured by a *hypergraph*, a generalization of a graph in which an edge can connect to any number of nodes. Briefly, a hypergraph $\mathbf{G}$ is a pair $\mathbf{G} = (X, E)$, where X is a set of elements called *nodes*, and E is a collection of sets of X called *hyperedges*. We can specify our problem with a hypergraph, which we refer to as the *region hypergraph* $\mathbf{G}^r = (\mathcal{H}, \mathcal{R})$.[1]

Note that in general, multiple decisions are equally suitable for a hypothesis: In the robot example, multiple manipulation actions may succeed for an object location (Fig. 5); in movie recommendation, the user may be indifferent among sets of movies. Hence, we allow the decision regions to overlap (Fig. 1(a)). Formally, we also assume that the set of hypotheses is covered by the collection of decision regions, i.e., $\mathcal{H} = \cup_{\mathcal{R}} r$.

[1] We illustrate decision regions as circles (e.g., Fig. 1(a)) - however, our method treats regions as arbitrary sets.



(a) *Regions and hypotheses* (b) *Subregions and hypergraph* (c) *Edges cut if all $h \in g_3$ inconsistent*

Figure 1: (a) An instance of the Decision Region Determination (DRD) problem with two decision regions. Black dots represent hypotheses and circles represent decision regions. (b) The resulting subregions and splitting hyperedges constructed by Hyperedge Cutting (HEC) algorithm. Thickness of edge represents weight, which is proportional to weight in subregion. (c) Resulting hypergraph when all hypotheses in subregion g_3 inconsistent, causing all edges to be "cut".

The ultimate goal is to find a policy π for running tests that allows us to determine a decision region r the true hypothesis is guaranteed to lie in. In other words, upon termination we require that $\mathcal{V}(\mathcal{S}) \subseteq r$ for some $r \in \mathcal{R}$.

Thus, we seek a policy for selecting a minimal number of tests to determine a suitable decision. A *policy* π is a function from a set of evidence so far $\mathcal{S}$, to the next test to choose (or to stop running tests). A policy is feasible if and only if it drives all remaining uncertainty into any single decision region, $\mathcal{V}(\mathcal{S}) \subseteq r$. We define the expected cost (i.e., number of tests[2]) of policy π as:

$$\mathcal{C}(\pi) = \sum_{h \in \mathcal{H}} P(h) |\mathcal{T}(\pi, h)|,$$

where $\mathcal{T}(\pi, h)$ is the set of tests policy π chooses in case the correct hypothesis is h. Given this, we seek a feasible policy of minimal cost, i.e.,

$$\pi^* = \arg\min_{\pi} C(\pi) \text{ s.t. } \forall h, \exists r : \mathcal{V}(\mathcal{T}(\pi, h)) \subseteq r \quad (2)$$

We call Problem (2) the *Decision Region Determination (DRD) Problem.*

Special cases of Problem (2) have been studied before. In particular, the special case where each hypothesis is contained in a dedicated region is called the Optimal Decision Tree (ODT) problem (Kosaraju and Borgstrom, 1999). More generally, the special case where the regions *partition* the hypothesis space (i.e., do not overlap), is called the Equivalence Class Determination (ECD) Problem (Golovin et al., 2010). For both of these special cases, it is known that finding a policy π for which $\mathcal{C}(\pi) \leq \mathcal{C}(\pi^*)o(\log n)$ is NP-hard (Chakaravarthy et al., 2007). Here, π^* indicates the optimum policy. To the best of our knowledge, there are no efficient algorithms with theoretical approximation guarantees for the general DRD problem. In the following, we present such an algorithm.

[2]Note that while we focus on tests with unit cost, our results generalize to tests with non-uniform costs.

3 The HEC Algorithm

We now introduce and analyze our algorithm – the *Hyperedge Cutting* (HEC) approach.

3.1 Overview

Our key strategy is to transform the DRD Problem (2) into an alternative representation – a different hypergraph for splitting decision regions. Observing certain test outcomes corresponds to downweighting or cutting hyperedges in this hypergraph. The construction is chosen so that cutting all hyperedges is a necessary and sufficient condition for driving all uncertainty into a single decision region. We then prove that a simple greedy algorithm, which chooses tests that reduce hyperedge weight maximally (in expectation), implements a policy that is competitive with the optimal (intractable) policy for Problem (2). In Sec. 4, we show how this greedy algorithm can be efficiently implemented.

3.2 Splitting hypergraph construction

We construct a different hypergraph, the *splitting hypergraph* $\mathbf{G}^s$, and define our objective on that. Here, our hyperedges are not sets, but *multisets*, a generalization of sets where members are allowed to appear more than once. As a result, a node can potentially appear in a hyperedge multiple times. The cardinality of a hyperedge refers to how many nodes it is connected to.

We observe that for solving the DRD problem, we can group together all hypotheses that share the same region assignments. We refer to this grouping as a *subregion* g, and the set of all subregions as $\mathcal{G}$. More formally, for any pair $h_k \in g_i$ and $h_l \in g_i$, we have $h_k \in r_j$ if and only if $h_l \in r_j$. In a slight abuse of notation, we say that a subregion is contained in a region, $g \in r$, if $\forall h \in g, h \in r$ (Fig. 1(b)). Similarly, we say that $h \in e$ if $\exists g \in e$ s.t. $h \in g$. It is easy to see that

all remaining hypotheses $\mathcal{V}(\mathcal{S})$ are contained in r if and only if all remaining subregions are contained in r.

We construct the splitting hypergraph $\mathbf{G}^s$ over these subregions. Each subregion $g \in \mathcal{G}$ corresponds to a node. The hyperedges $e \in \mathcal{E}$ consist of all multisets of precisely k subregions, $e = \{g_1, \ldots, g_k\}$, such that a single decision region does not contain them all (we will describe how k is selected momentarily). Note that hyperedges can contain the same subregion multiple times. Formally,

$$\mathcal{E} = \{e : |e| = k \wedge \nexists\, r \text{ s.t. } \forall h \in e, h \in r\}. \quad (3)$$

Our splitting hypergraph is defined as $\mathbf{G}^s = (\mathcal{G}, \mathcal{E})$. Fig. 1(b) illustrates the splitting hypergraph obtained from the DRD instance of Fig. 1(a).

Hyperedge Cardinality k. Key to attaining our results is the proper selection of hyperedge cardinality k. If k is too small, our results won't hold, and our algorithm won't solve the DRD problem. If k is too large, we waste computational effort, and our theoretical bounds loosen. Here, we define the cardinality we use practically. Our theorems hold for a smaller, more difficult to compute k as well. See the extended version for details.

$$k = \min\left(\max_{h\in\mathcal{H}} |\{r : h \in r\}|, \max_{r\in\mathcal{R}} |\{g : g \in r\}|\right) + 1 \quad (4)$$

Note that each term is a property of the original region hypergraph $\mathbf{G}^r$ defined in Sec. 2: $\max_h |\{r : h \in r\}|$ is the maximum degree of any node, and $\max_r |\{g : g \in r\}|$ bounds the maximum cardinality of hyperedges in $\mathbf{G}^r$.[3]

3.3 Relating DRD and HEC

How does the hypergraph capture our progress towards solving Problem (2)? Observing a set of test-outcomes $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$ eliminates inconsistent hypotheses, and consequently downweights or eliminates ("cuts") incident hyperedges (Fig. 1(c)). Analogous to the definition of $\mathcal{V}(\mathcal{S})$ in (5), we define the set of hyperedges consistent with $\mathcal{S}$ by

$$\mathcal{E}(\mathcal{S}) = \{e \in \mathcal{E} : \forall (i, o) \in \mathcal{S}\ \forall h \in e, h(i) = o\} \quad (5)$$

The following result guarantees that cutting all hyperedges is a necessary and sufficient condition for success, i.e., driving all uncertainty into a single decision region.

Theorem 1. *Suppose we construct a splitting hypergraph by drawing hyperedges of cardinality k according to (3). Let $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$ be a set of evidence. All consistent hypotheses lie in some decision region if and only if all hyperedges are cut, i.e.,*

$$\mathcal{E}(\mathcal{S}) = \emptyset \quad \Leftrightarrow \quad \exists r : \mathcal{V}(\mathcal{S}) \subseteq r$$

[3] It is precisely the maximum cardinality of any hyperedge if we grouped hypotheses into subregions in $\mathbf{G}^r$.

Thus, the DRD Problem (2) is equivalent to finding a policy of minimum cost that cuts all hyperedges. This insight suggests a natural algorithm: select tests that cut as many edges as possible (in expectation). In the following, we formalize this approach.

3.4 The Hyperedge Cutting (HEC) Algorithm

Given the above construction, we define a suitable objective function whose maximization will ensure that we pick tests to remove hyperedges quickly, thus providing us with an algorithm that identifies a correct decision region. First, we define the weight of a subregion as the sum of hypothesis weights, $p(g) = \sum_{h\in g} p(h)$. We define the weight of a hyperedge $e = \{g_1, \ldots, g_k\}$ as $w(e) = \prod_{i=1}^{k} P(g_i)$. More generally, we define the weight of a collection of hyperedges as $w(\{e_1, \ldots, e_n\}) = \sum_{l=1}^{n} w(e_l)$. Now, given a pair of test/observation (t, o), we can identify the set of inconsistent hypotheses, which in turn implies the set of hyperedges that should be downweighted or removed. Formally, given a set of test/observation pairs $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$, we define its utility $f_{\text{HEC}}(\mathcal{S})$ as

$$f_{\text{HEC}}(\mathcal{S}) = w(\mathcal{E}) - w(\mathcal{E}(\mathcal{S})). \quad (6)$$

Thus $f_{\text{HEC}}(\mathcal{S})$ is the total mass of all the edges cut via observing set $\mathcal{S}$.

A natural approach to the DRD Problem is thus to seek policies that maximize (6) as quickly as possible. Arguably the simplest approach is a greedy approach that iteratively chooses the test that increases (6) as much as possible, in expectation over test outcomes.

Formally, we define the *expected marginal gain* of a test t given evidence $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$ as follows:

$$\Delta(t \,|\, \mathcal{S}) = \sum_h P(h \,|\, \mathcal{S}) \Big(f_{\text{HEC}}(\mathcal{S} \cup \{(t, h(t))\}) - f_{\text{HEC}}(\mathcal{S}) \Big)$$

Thus, $\Delta(t \,|\, \mathcal{S})$ quantifies, for test t, the expected reduction in hyperedge mass upon observing the outcome of the test. Hereby, the expectation is taken w.r.t. the distribution over hypotheses conditioned on our evidence so far. It is apparent that all hyperedges are cut if and only if $\Delta(t \,|\, \mathcal{S}) = 0$ for all tests $t \in \mathcal{T}$. Given this, our HEC Algorithm simply starts with $\mathcal{S} = \emptyset$. It then proceeds in an iterative manner, greedily selecting the test t^* that maximizes the expected marginal benefit, $t^* = \arg\max_t \Delta(t \mid \mathcal{S})$, observes the outcome $h(t^*)$ and adds the pair $(t^*, h(t^*))$ to $\mathcal{S}$. It stops as soon as all edges are cut (i.e., the marginal gain of all tests is 0).

3.5 Theoretical Analysis

The key insight behind our analysis is that the marginal gain $\Delta(t \mid \mathcal{S})$ satisfies two properties: *adaptive monotonicity* and *adaptive submodularity*, introduced by

Golovin and Krause (2011) and associated with certain sequential decision problems. Formally, adaptive monotonicity simply states that the benefit of each test is nonnegative, $\Delta(t \mid \mathcal{S}) \geq 0$ for all tests $t \in \mathcal{T}$ and evidence $\mathcal{S} \subseteq \mathcal{T} \times \mathcal{O}$. This is straightforward, since carrying out a test can never introduce hyperedges, but only remove them. The second, slightly more subtle property – *adaptive submodularity* – states that the marginal gain of any fixed test $t \in \mathcal{T}$ can never increase as we gain additional evidence. Formally, whenever $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{T} \times \mathcal{O}$, it must hold that $\Delta(t \mid \mathcal{S}) \geq \Delta(t \mid \mathcal{S}')$. Those properties are formally established for our f_{HEC} objective and the associated marginal gain Δ in the following Theorem:

Theorem 2. *The objective function f_{HEC} defined in (6) is adaptive submodular and strongly adaptive monotone.*

Why are these properties useful? Golovin and Krause (2011) prove that for sequential decision problems satisfying adaptive monotonicity and adaptive submodularity, greedy policies are competitive with the optimal policy. In particular, as a consequence of Theorem 2 and Theorem 5.8 of Golovin and Krause (2011), we obtain the following result for our HEC Algorithm:

Theorem 3. *Assume that the prior probability distribution P on the set of hypotheses is rational. Then, the performance of π_{HEC} is bounded as follows:*

$$\mathcal{C}(\pi_{HEC}) \leq (k \ln(1/p_{\min}) + 1)\mathcal{C}(\pi^*),$$

where $p_{\min} = \min_{h \in \mathcal{H}} P(h)$.

For the special case of disjoint regions (i.e., the ECD Problem, corresponding to $k = 2$), our objective f_{HEC} is equivalent to the objective function proposed by Golovin et al. (2010), and hence our Theorem 3 strictly generalizes their result. Furthermore, in the special case where each test can have at most two outcomes, and we set $k = 1$, the HEC Algorithm is equivalent to the Generalized Binary Search algorithm for the ODT problem, and recovers its approximation guarantee.

4 Efficient Implementation

Our HEC algorithm computes $\Delta(t|\mathcal{S})$ for every test in $\mathcal{T}$, and greedily selects one at each time step. Naively computing this quantity involves constructing the splitting hypergraph $\mathbf{G}^s$ for every possible observation, and summing the edge weights. This is computationally expensive, as constructing the graph requires enumerating every multiset of order k and checking if any region contains them all, resulting in a runtime of $O(|\mathcal{G}|^k)$. We can, however, quickly prune checks and iteratively consider multisets of growing cardinality during our computation by utilizing the following fact:

Algorithm 1 Hyperedge Weight

procedure HYPEREDGE WEIGHT($\mathcal{H}, k$)
  Compute subregions $\mathcal{G}$ from $\mathcal{H}$
  $W \leftarrow CHP_k(\mathcal{G})$
  Initialize queue Q_1 with every subregion $g \in \mathcal{G}$
  for all $\widehat{k} \leq k$ **do**
    for all $\boldsymbol{\zeta}_{\widehat{k}} \in Q_{\widehat{k}}$ **do**
      if $\exists r$ s.t. $\forall h \in \boldsymbol{\zeta}_{\widehat{k}},\ h \in r$ **then**
        $W \leftarrow W - \prod_{g \in \boldsymbol{\zeta}_{\widehat{k}}} p(g) CHP_{k-\widehat{k}}(\boldsymbol{\zeta}_{\widehat{k}})$
        Add all supersets of $\boldsymbol{\zeta}_{\widehat{k}}$ to $\mathrm{Q}_{\widehat{k}+1}$
  return W

Proposition 1. *A set of subregions G shares a region only if all subsets $G' \subset G$ also share that region.*

4.1 Utilizing Complete Homogeneous Symmetric Polynomials

Our general strategy will be to compute the sum of weights over *all* multisets of cardinality k, and subtract those that correspond to a shared region. To do so efficiently, we identify algebraic structure in computing a sum of multisets, where a multiset corresponds to a product. Namely, it is equivalent to computing a complete homogeneous symmetric polynomial.

For any $G \subseteq \mathcal{G}$ and cardinality $\widehat{k}$, we define $\mathcal{G}_{\widehat{k}}(G)$ as all multisets over groups G of cardinality $\widehat{k}$. Unlike hyperedges, these multisets can share a region. Formally

$$\mathcal{G}_{\widehat{k}}(G) = \left\{\{g_1, \ldots, g_{\widehat{k}}\} \subseteq \mathcal{G}\right\}$$

Recall that $w(\mathcal{G}_{\widehat{k}}(G)) = \sum_{\mathcal{G}_{\widehat{k}}(G)} \prod_g P(g)$. Computing $w(\mathcal{G}_{\widehat{k}}(G))$ can be performed efficiently as this quantity is exactly equivalent to the *complete homogeneous symmetric polynomial* (CHP) of degree $\widehat{k}$ over G. We will briefly review a well known variant of the Newton-Girard formulae which will make an efficient algorithm for computing $w(\mathcal{G}_{\widehat{k}}(G))$ clear.

Define any set of variables $\mathbf{x} = \{x_1, \cdots, x_n\}$.

$$PS_i(\mathbf{x}) = \sum_{x \in \mathbf{x}} x^i$$

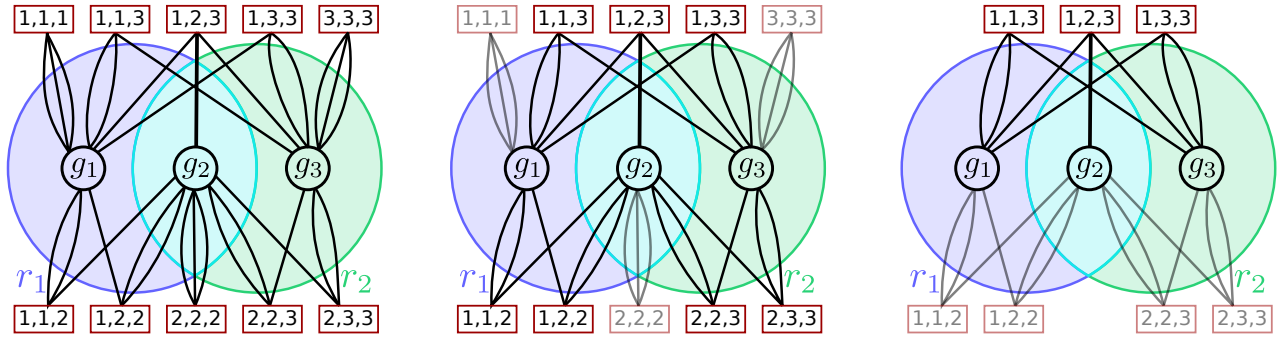
$$CHP_i(\mathbf{x}) = \sum_{l_1 + \ldots l_n = i; l_j \geq 0} \prod_{x_j \in \mathbf{x}} x_j^{l_j}$$

Here PS_i is the i-th *power sum*, and CHP_i is the i-th complete homogeneous symmetric polynomial.

We have the identity (Macdonald, 1998; Seroul, 2000):

$$CHP_i(\mathbf{x}) = \frac{1}{i} \sum_{j=1}^{i} CHP_{i-j}(\mathbf{x}) PS_j(\mathbf{x})$$

Thus, we iteratively compute $CHP_1(G) \ldots CHP_{\widehat{k}}(G)$ to compute $w(\mathcal{G}_{\widehat{k}}(G)) = CHP_{\widehat{k}}(G)$ with runtime $O(\widehat{k}|G|)$.



(a) *All Multisets* (b) $|\boldsymbol{\zeta}| = 1$ *sets removed* (c) $|\boldsymbol{\zeta}| = 2$ *sets removed*

Figure 2: A depiction of our algorithm as hyperedges. (a) The equivalent hyperedges of $CHP_3(\mathcal{G})$. (b) First iteration Alg. 1 which removes all $|\boldsymbol{\zeta}| = 1$ (light edges) by subtracting $g_1 CHP_2(\{g_1\}) + g_2 CHP_2(\{g_2\}) + g_3 CHP_2(\{g_3\})$. (c) Second iteration of Alg. 1 which removes all $|\boldsymbol{\zeta}| = 2$ (light edges) by subtracting $g_1 g_2 CHP_1(\{g_1, g_2\}) + g_2 g_3 CHP_1(\{g_2, g_3\})$

We now turn our attention to efficiently computing the weight of all multisets that correspond to subregions encapsulated by a region. Let $\boldsymbol{\zeta}$ be a set (not multiset) of subregions that shares a region. Formally:

$$\boldsymbol{\zeta} = \{g_1 \dots g_{\widehat{k}}\} \qquad \widehat{k} \leq k, \nexists r \text{ s.t. } \boldsymbol{\zeta} \subseteq r$$

We compute the term corresponding to $\boldsymbol{\zeta}$ we subtract from $CHP_k(\mathcal{G})$ (weight of all multisets), as $\boldsymbol{\zeta}$ shares a region. To avoid double counting, we force the term to include $\prod_{g \in \boldsymbol{\zeta}} p(g)$ as a factor, i.e. if we think of a hyperedge as a product, we force one link to each element of $\boldsymbol{\zeta}$.

$$\begin{aligned} w(\boldsymbol{\zeta}) &= \prod_{g \in \boldsymbol{\zeta}} p(g) \sum_{l_1 + \dots l_{\widehat{k}} = k - \widehat{k}; l_i > 0} p(g_1)^{l_1} \dots p(g_{\widehat{k}})^{l_{\widehat{k}}} \\ &= \prod_{g \in \boldsymbol{\zeta}} p(g) CHP_{k-\widehat{k}}(\boldsymbol{\zeta}) \end{aligned}$$

Using this, we compute $w(\mathcal{E}) = CHP_k(\mathcal{G}) - \sum_{\boldsymbol{\zeta} \subseteq \mathcal{G}} w(\boldsymbol{\zeta})$ by finding every set $\boldsymbol{\zeta} \subseteq \mathcal{G}$ that shares a region. Furthermore, we can utilize Proposition 1 to prune sets, and only consider $\boldsymbol{\zeta}_{\widehat{k}+1}$ which are supersets of any $\boldsymbol{\zeta}_{\widehat{k}}$. The algorithm is detailed in Alg. 1, and depicted in Fig. 2.

Additionally, we note that region assignments do not change as observations are received. In practice, we find all sets of subregions that share a region once. At each time step, we need only sum over the terms corresponding to remaining hypotheses.

Note that in the worst case, this algorithm still has complexity $O(|\mathcal{G}|^k)$. This occurs when many, at least k, subregions share a single region. The complexity is then controlled by how many distinct subregions a single region can be shattered into, and the largest number of regions a single hypothesis can belong to. However, for many practical problems, we might expect many regions to be separated, e.g., when $|\mathcal{R}| \gg k$. In this case, Alg. 1 will be significantly more efficient.

Finally, we note that we can utilize an *accelerated adaptive greedy* algorithm applicable to all adaptive submodular functions, which directly uses the diminishing returns property to skip reevaluation of actions (Golovin and Krause, 2011).

5 Experiments

In this section, we empirically evaluate HEC on the two applications - approximate comparison-based learning and touch based localization with a robotic end effector.

We compare HEC with five baselines. The first two are variants of algorithms for the specialized versions of the DRD problem described earlier - generalized binary search (Nowak, 2009) and equivalence class edge cutting (Golovin et al., 2010). For generalized binary search (GBS), we assign each hypothesis to its own decision region, and run HEC on this hypothesis-region assignment until only one hypothesis remains. To apply equivalence class edge cutting (EC2), decision regions must be disjoint. Thus, we randomly assign each hypothesis to one of the decision regions that it belongs to, and run EC2 until only one of these new regions remains. For each of these, we also run a slightly modified version, termed GBS-HEC and EC2-HEC respectively, which selects tests based on these algorithms, but terminates once all hypotheses are contained in one decision region in the original DRD problem (i.e. when the HEC termination condition is met).

The last baseline is a classic heuristic from decision theory: myopic value of information (VoI) (Howard, 1966). We define a utility function $U(h, r)$ which is 1 if $h \in r$ and 0 otherwise. The utility of $\mathcal{V}(\mathcal{S})$ corresponds to the maximum expected utility of any decision region, i.e., the expected utility if we made a decision now. VoI greedily chooses the test that maximizes (in expectation over observations) the gain in this utility. Note that if we could solve the intractable problem of nonmyopically optimizing VoI (i.e., look ahead arbitrarily to consider outcomes of sequences of tests),

we could solve the DRD problem optimally. In some sense, HEC can be viewed as a surrogate function for nonmyopic value of information.

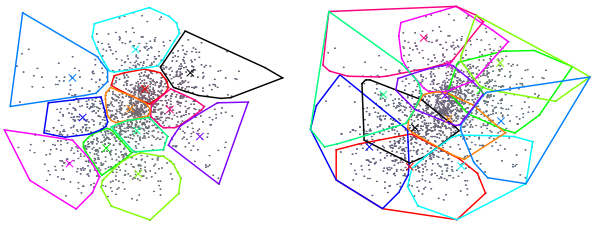
5.1 Approximate comparison-based learning

We evaluate HEC on the `MovieLens 100k`[4] dataset, which consists of 1 to 5 ratings of 1682 movies from 943 users. We partition movies into decision regions using these ratings, with the goal of recommending any movie in a decision region. In order to get a similarity measurement between movies, we map them into a 10-dimensional feature space by computing a low-rank approximation of the user/rating matrix through SVD. We then use k-means to partition the set of movies into $|\mathcal{R}|$ (non-overlapping) clusters, corresponding to decision regions. Each movie is then assigned to the α closest cluster centroids. See Fig. 4 for an illustration. A test corresponds to comparing two movies, an observation to selecting one of the two, and consist hypotheses are those which are closer to the selected movie (euclidean distance in 10-dimensional feature space).

Each experiment corresponds to sampling one movie as the "true" movie. The size of a decision region determines how close our solution is to this (exact) target hypothesis. As the number of regions increases, the size of each decision region shrinks. As a result, the problem requires the selected movie be closer to the true target, at the expense of increased query complexity. Fig. 3(a) shows the query complexity of different algorithms as a function of the number of regions, with the cardinality of the HEC hypergraph fixed to $k = 3$ (i.e., each hypothesis belongs to two decision regions). An extreme case is when there are only two regions and all hypotheses belong to both regions, giving a query complexity of 0. Other than that, we see that HEC performs consistently better than other methods (e.g., to identify the true region out of 8 regions, it takes on average 6.7 queries for HEC, as opposed to 8 queries for EC2-HEC, 8.5 queries for GBS-HEC, and 10.3 queries for VoI).

To see how the cardinality and region overlap influence performance, we compare the query complexity of different algorithms by varying the number of regions each hypothesis is assigned to. If we assign more regions to a hypothesis, then the search result is allowed to be further away from the true target, and thus the number of queries required for approximated search should be smaller. Fig. 3(b) demonstrates such an effect. We fix the number of clusters to 12, and vary the number of assigned regions (and thus the hyperedge cardinality) from 1 to 4 (k from 2 to 5, respectively). We see that higher cardinality enables HEC to save more queries. For $k = 5$, it takes HEC 5.3 queries to identify a movie,

[4]http://www.grouplens.org/datasets/movielens/



(a) *Partitions* ($k = 2$) (b) *Decision regions* ($k = 3$)

Figure 4: A 2-d illustration of (overlapping) decision regions for `MovieLens 100k` experiments. Dots represent movies, cross markers represent cluster centroids, and colored polygons represent decision region boundaries. (a) Movies are partitioned into 12 disjoint clusters. (b) Each movie is assigned to the two closest centroids.

k	2	3	4	5
t(HEC)	0.026s	0.071s	2.5s	$<$ 2min

Table 1: Running time of HEC on `MovieLens 100k` with different cardinality k ($|\mathcal{R}| = 12$)

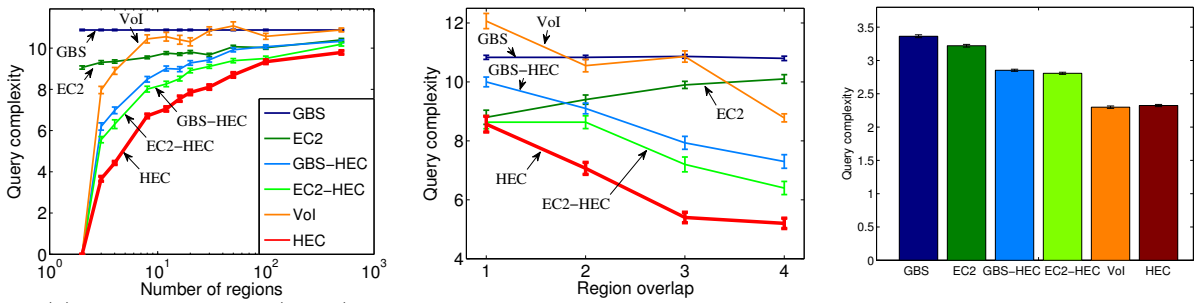
whereas VoI, GBS-HEC, and EC2-HEC took 8.8, 7.4, and 6.4 queries, respectively. Additionally, Table 5.1 shows the running time of HEC for these instances. We see that the accelerated implementation described in Sec. 4 enables HEC to run efficiently with reasonable hyperedge cardinality on this data set.

5.2 Touch Based localization

We evaluate HEC on a simple robotic manipulation example. Our task is to push a button with the finger of a robotic end effector. Given a distribution over object location, we generate a set of decisions, corresponding to the end effector going to a particular pose and moving forward in a straight line. Each of these decisions will succeed on a subset of hypotheses, corresponding to a decision region. Decision regions may overlap, as a button can be pushed with many decision actions. See Fig. 5.

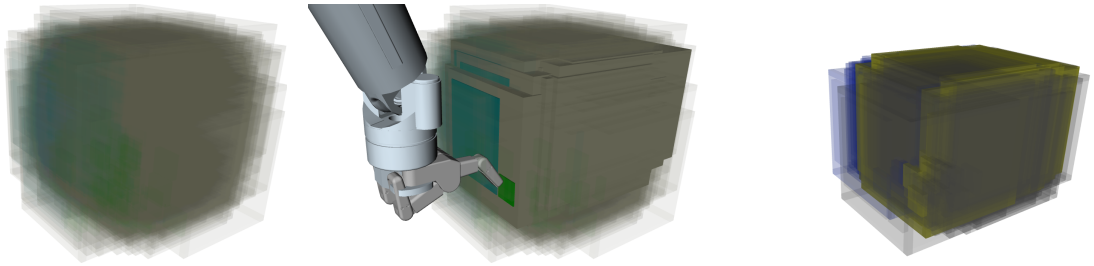
All hypotheses are not contained in a single decision region, so we perform tests to reduce uncertainty. These tests correspond to *guarded moves* (Will and Grossman, 1975), where the end effector moves along a path until contact is sensed. After sensing contact, hypotheses are updated by eliminating object locations which could not have produced contact, e.g., if they are far away. Our goal is to find the shortest sequence of tests such that after performing them, there is a single button-push decision that would succeed for all remaining hypotheses.

Given some object location X_s, we generate an initial set of 2000 hypotheses $\mathcal{H}$ by sampling from $N(\mu, \Sigma)$ with $\mu = X_s$, and Σ a diagonal matrix with $\Sigma_{xx} = \Sigma_{yy} = \Sigma_{zz} = 0.04$. The robot generates 50



(a) *MovieLens 100k* ($k = 3$) (b) *MovieLens 100k* ($|\mathcal{R}| = 12$) (c) *Robotic manipulation simulation*

Figure 3: Results on `MovieLens 100k` and `Robot` experiments. (a) Performance as we vary the number of regions $|\mathcal{R}|$. (b) Performance as we vary the cardinality k. (c) Average performance of different algorithms across button push instances.



(a) Hypotheses (b) A decision region (c) Two regions

Figure 5: Touch based localization for pushing the button of a microwave. Given hypotheses over object location (a), decision actions are generated. The corresponding decision regions are computed by forward simulating to find hypotheses for which it would succeed (b). Decision regions will overlap. In (c), we see two regions (blue and grey) and their overlap (yellow).

decision regions by picking different locations and simulating the end effector forward, and noting which object poses it would succeed on. Hypotheses range from being in zero decision regions to 6, giving us a cardinality $k = 7$. For tests, the robot generates 150 guarded moves by sampling a random start location and orientation.

We conduct experiments on 10 random environments, and randomly sample 100 hypotheses to be the "true" object location (for producing observations during execution), for a total of 1000 experiments. Fig. 3(c) shows the query complexity of different algorithms averaged over these instances. We see that HEC performs well, outperforming GBS, GBS-HEC, EC2, and EC2-HEC handily. Note that myopic VoI performs essentially the same as HEC on these experiments. This is likely due to the short horizon, where 2-3 actions were usually sufficient for reducing uncertainty to a single decision region. We would expect that for longer horizons, myopic VoI would not perform as well.

6 Conclusions

In this paper, we have addressed the problem of active learning in order to facilitate decision making. We defined the Decision Region Determination (DRD) problem, requiring that at the end of information gathering, all remaining hypotheses are confined within a single decision region (i.e., do not require further distinction from a decision making point of view). To address this problem, we proposed an equivalent representation in terms of a hypergraph. We prove that eliminating all edges in this hypergraph is a necessary and sufficient condition for success, suggesting a natural objective function. We show that this objective satisfies adaptive monotonicity and adaptive submodularity. This insight enabled us to prove that a greedy policy for removing hyperedges (HEC) has an approximation guarantee compared to the optimal policy. Finally, we note that at each iteration, we compute a particular polynomial, and can utilize a faster algorithm through efficient computations of complete homogeneous symmetric polynomials.

While our algorithm enables us to tackle problems of reasonable size, our computation is still exponential in hyperedge cardinality k. Additionally, our current scheme assumes *noise-free* observations, where a hypothesis deterministically maps a test to an observation. We hope to alleviate these limitations in future work.

Acknowledgements

This work was supported in part by the Intel Embedded Computing ISTC, NSF Grant No. 0946825, NSF-IIS-1227495, DARPA MSEE FA8650-11-1-7156, ERC StG 307036, and a Microsoft Research Faculty Fellowship.

References

N. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

V. T. Chakaravarthy, V. Pandit, S. Roy, P. Awasthi, and M. K. Mohania. Decision trees for entity identification: Approximation algorithms and hardness results. In *PODS*, 2007.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10:273–304, 1995.

S. Dasgupta. Analysis of a greedy active learning strategy. In *NIPS*, 2004.

D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *JAIR*, 42(1):427–486, 2011.

D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *NIPS*, 2010.

N. Goyal, Y. Lifshits, and H. Schütze. Disorder inequality: A combinatorial approach to nearest neighbor search. In *WSDM*, 2008.

A. Guillory and J. Bilmes. Average-case active learning with costs. In *ALT*, 2009.

R. A. Howard. Information value theory. In *IEEE Trans. Syst. Sci. Cybernetics*, 1966.

K. Hsiao, T. Lozano-Pérez, and L. P. Kaelbling. Robust belief-based execution of manipulation programs. In *WAFR*, 2008.

S. Javdani, M. Klingensmith, J. A. D. Bagnell, N. Pollard, and S. Srinivasa. Efficient touch based localization through submodularity. In *IEEE ICRA*, 2013.

A. Karbasi, S. Ioannidis, and L. Massoulie. Comparison-based learning with rank nets. In *ICML*, 2012.

T. Kollar and N. Roy. Efficient optimization of information-theoretic exploration in slam. In *AAAI*, 2008.

I. Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artif. Intell. Med.*, 23:89–109, 2001.

R. S. Kosaraju and T. M. P. A. R. S. Borgstrom. On an optimal split tree problem. In *WADS*, 1999.

D. V. Lindley. On a measure of the information provided by an experiment. *Ann. Math. Stat.*, 27: 986–1005, 1956.

I. G. Macdonald. *Symmetric Functions and Hall Polynomials*. Oxford mathematical monographs. Clarendon Press, 1998. ISBN 9780198504504.

R. Nowak. Noisy generalized binary search. In *NIPS*, 2009.

R. Seroul. *Programming for Mathematicians*. Universitext - Springer-Verlag. Springer, 2000. ISBN 9783540664222.

P. M. Will and D. D. Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Trans. Computers*, 24(9):879–888, 1975.