Acquisition Conditioned Oracle for Nongreedy Active Feature Acquisition

Michael Valancius [1] **Max Lennon** [2] **Junier B. Oliva** [2]

Abstract

We develop novel methodology for active feature acquisition (AFA), the study of sequentially acquiring a dynamic subset of features that minimizes acquisition costs whilst still yielding accurate inference. The AFA framework can be useful in a myriad of domains, including health care applications where the cost of acquiring additional features for a patient (in terms of time, money, risk, etc.) can be weighed against the expected improvement to diagnostic performance. Previous approaches for AFA have employed either: deep learning RL techniques, which have difficulty training policies due to a complicated state and action space; deep learning surrogate generative models, which require modeling complicated multidimensional conditional distributions; or greedy policies, which cannot account for jointly informative feature acquisitions. We show that we can bypass many of these challenges with a novel, nonparametric oracle based approach, which we coin the acquisition conditioned oracle (ACO). Extensive experiments show the superiority of the ACO to state-of-the-art AFA methods when acquiring features for both predictions and general decision-making.

1. Introduction

An overwhelming bulk of efforts in machine learning assume access to a fully observed feature vector, $x \in \mathbb{R}^d$. However, this paradigm largely ignores that the collection of features comes at a cost, limiting applicability in real-world scenarios that involve making judgements with some features or information missing. Unlike in conventional imputation and partially observed methodology, we note that many real-world situations allow for the dynamic collection of information on an instance at inference time. That is, often one may query (at a cost) the environment for additional information (features) that are currently missing for an instance. This shall be especially relevant in an automated future, where autonomous agents can routinely interact with an environment to obtain more information. In this work we develop novel methodology for *active feature acquisition* (AFA) (Saar-Tsechansky et al., 2009), the study of how to sequentially acquire a dynamic (on a per instance basis) subset of features that minimizes acquisition costs whilst yielding accurate inferences.

Consider the following illustrative application where AFA is useful: an automated survey system to perform psychological assessments. A traditional ML approach would require collecting responses on an exhaustive list of survey questions, and only after all responses (features) are collected would one make a prediction of the correct assessment. However, administering a long survey is slow, may lead to user fatigue (Early et al., 2016a), or may even decrease the accuracy of responses (Early et al., 2016b). Instead, an AFA approach would *sequentially* decide what next question (if any) to prompt the user with to help it make its assessment, ultimately leading to a prediction with a succinct, personalized subset of features per instance. Note that in contrast to traditional feature selection, which would always select the same subset of questions, an AFA approach will determine a custom subset of questions (of potentially varying cardinality) to ask on a per case basis. This is because, in AFA, the next feature (answer to a question) to acquire can depend on the values of previous acquisitions. Similar applications include educational assessments, automated trouble shooting systems, and cyberphysical systems.

In this work, we first present an alternative perspective from the two predominant approaches to active feature acquisition: reinforcement learning (RL) algorithms that are theoretically optimal but practically challenging to train, and greedy approaches that are computationally simpler but may lead to sub-optimal sets of acquired features. In contrast to both, we contribute the following: (1) ACO, a simple, nonparametric policy approach that is non-greedy while circumventing the need for training a general RL policy;

[1] Department of Biostatistics, University of North Carolina, Chapel Hill, North Carolina, USA [2] Department of Computer Science, University of North Carolina, Chapel Hill, North Carolina, USA. Correspondence to: Michael Valancius <mval@email.unc.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

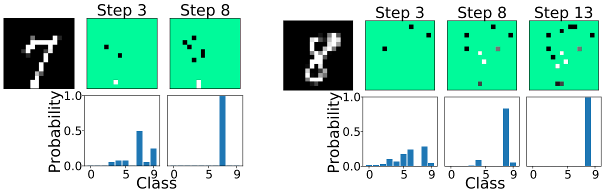


Figure 1. In this MNIST example, pixels are sequentially observed (acquired) until prediction confidence exceeds the cost of any more acquisitions. The acquisition process (top) and the prediction probabilities (bottom) using the ACO agent vary per instance.

(2) a generalization of the ACO policy to the less-explored AFA task of making a *decision* to maximize expected outcomes. We show that ACO achieves competitive empirical performance across a variety of established prediction benchmarks.

2. Related Directions

As noted by Li & Oliva (2021), AFA is related to feature selection and active learning, as follows.

Feature Selection Feature selection is a well studied task (e.g., see surveys by (Khaire & Dhanalakshmi, 2022; Venkatesh & Anuradha, 2019; Miao & Niu, 2016)) to ascertain a *constant* subset of features that are predictive. Feature selection can be seen as a special case of AFA (perhaps a stubborn one) that selects the same next features to observe regardless of the previous feature values it has encountered. In contrast with feature selection methods, which make predictions based on a fixed subset of features, AFA dynamically selects features on an instance-by-instance basis that is sequentially chosen (Fig. 1) for better prediction. (Note that AFA may be applied after an initial feature selection preprocessing step to reduce the feature space.) A dynamic strategy helps when covariates are indicative of other features that are dependant with the output.

Active Learning Active learning is another interactive task in ML to gather more labeled instances (e.g., see (Fu et al., 2013; Konyushkova et al., 2017; Yoo & Kweon, 2019; MacKay, 1992; Houlsby et al., 2011)). Active learning considers queries to an expert for the correct output label to a complete set of features in order to construct a training instance to build a better classifier. Instead, AFA considers queries to the environment for the feature value corresponding to an unobserved feature dimension, i, in order to provide a better prediction on the current instance. That is, while the active learning paradigm queries an *expert* during *training* to build a classifier with complete features, the AFA paradigm queries the *environment* at *evaluation* to obtain unobserved features of a current instance to help assess it (with few acquired features); thus, AFA is a distinct problem.

3. Methods

We develop a data-driven, deployable approximation to an oracle that solves a novel AFA objective. Furthermore, we extend the scope of AFA beyond prediction, modifying the objective and method to the active acquisition of features when making a decision. The strong performance of our methodology (see **Experiments**) compared to more computationally-intensive methods provides evidence for the advantage of our newly proposed objective and lays the groundwork for the development of future solutions.

3.1. AFA MDP

We now expound on the formal definition of the AFA problem. Throughout, we consider underlying instances $x \in \mathbb{R}^d$ and corresponding labels y. We denote the i^{th} feature as $x_i \in \mathbb{R}$, and a subset of feature values $o \subseteq \{1, \dots, d\}$ as $x_o \in \mathbb{R}^{|o|}$. As several previous works have noted (Zubek & Dietterich, 2002; Rückstieß et al., 2011; Shim et al., 2018), AFA can be succinctly encapsulated as the following Markov decision process (MDP): states $s = (x_o, o)$ are comprised of the currently observed features o, and the respective values x_o; actions $a \in (\{1, \dots, d\} \setminus o) \cup \{\phi\}$ indicate whether to acquire a new feature value, $a \in \{1, \dots, d\} \setminus o$, or to terminate with a prediction, $a = \phi$; when making a prediction ($a = \phi$) rewards r are based on a supervised loss $-\ell(\hat{y}(x_o, o), y)$ of a prediction based on observed features $\hat{y}(x_o, o)$ and the ground truth label y, otherwise the reward is a negative cost of acquiring another feature onto o, $-c(a, o)$ (commonly at a constant cost $c(a, o) = \alpha$); lastly, for non-terminal actions, the state transitions $(x_o, o) \to (x_{o \cup \{a\}}, o \cup \{a\})$. Then, the objective of the AFA problem essentially reduces to learning an acquisition policy $\pi(x_o, o)$ that maximizes (possibly a variant of) the value function.

3.2. Challenges

One of the core difficulties of the AFA problem is that while its MDP formulation is general and encapsulating, it yields an RL problem with a large action space (acquire feature i or terminate to predict), a complicated state space of evolving dimension (values of acquired features), and sparse rewards. As a concrete example, even if each of the d features has values lying in only k categories, the dimension of the state space is $\sum_{i=0}^{d} \binom{d}{i} k^i$ (super exponential in d). These challenges make for a difficult RL agent optimization task (Dulac-Arnold et al., 2015; Minsky, 1961), especially with a finite amount of training data (a finite number of potential states seen during training roll-outs). When the labels and feature space are discrete (or can be discretized), previous work has shown promise by noting special properties about the optimal solution and utilizing dynamic programming (Liyanage & Zois, 2021).

One approach to assuage these issues is to attempt to incorporate more explicit distributional knowledge into the acquisition policy and/or tweak the value function. For instance, Li & Oliva (2021) provided the policy with surrogate information about feature uncertainties and used auxiliary rewards via a generative model to guide its acquisitions. While this led to empirical improvements over direct RL approaches such as (Shim et al., 2018), it requires learning a complicated, accurate deep generative model for all conditional dependencies (which, as noted above, grows at a super-exponential rate) in addition to still needing to train a deep RL agent.

An alternative approach restricts the complexity of the learning task by limiting the search to a greedy policy class (Ma et al., 2018; He et al., 2012; 2016; Covert et al., 2023). For instance, (Ma et al., 2018; Gong et al., 2019) use a generative approach to estimate the expected utility (e.g., mutual information) of any one acquisition, employing a greedy strategy to maximize utility; alternatively, Covert et al. (2023) advances this learning approach by showing how amortized optimization can be used to approximate the conditional mutual information. This greedy approach allows one to implicitly construct a policy by only learning the locally optimum policy at each each acquisition step, easing the computational burden compared to the general RL approach. However, this comes at the expense of possibly overlooking jointly informative sets of features.

We motivate our proposed oracle by first contrasting it with previous "retrospective oracle" approaches (He et al., 2012; 2016; Madasu et al., 2022). Such approaches will look for the next feature, i, that, when added to the already acquired features, o, results in the best improvement of a loss ℓ (e.g., MSE or cross-entropy) using an estimator $\hat{y}$:

$$i = \operatorname*{arg\,min}_{j \in \{1,\ldots,d\} \setminus o} \ell\left(\hat{y}(x_{o \cup \{j\}}), y\right). \tag{1}$$

This approach has been previously described as an "oracle" (e.g., the "forward-selection oracle," (He et al., 2012)). We offer an alternative perspective by defining an oracle which solves a novel AFA objective meant to balance the limitations posed by the (full) RL and greedy approaches. We argue that our oracle improves on the oracle defined by eq. (1) by being:

1. **Non-greedy**: Previous retrospective approaches have acquired the feature which most decreases the prediction loss at one time step. Greedily acquiring the next best feature ignores jointly informative groups of multiple features that may be acquired. Thus, we consider acquisitions *collectively* for the final prediction.
2. **Deployable**: In principle, an oracle should be deployable in one's environment, as it should be able to yield the correct action given *the same information* (the same states) as the agent. However, the resulting action from eq. (1) depends on inputs x (the entire feature vector), y (the respective true label), and o (the already acquired feature indices), thereby "cheating" by operating over more information than the AFA policy, $\pi(x_o, o)$, has available. This has a practical implication: as a consequence of utilizing information that is not available during a roll-out, the cheating oracle is not deployable at inference time. Therefore, it can only serve as a reference teacher policy in AFA (He et al., 2012; 2016; Madasu et al., 2022). Instead, our approach shall yield a directly deployable oracle, making it possible to judge its performance at test time and disentangling any degradation in performance due to learning a policy to imitate the oracle.

3.3. Acquisition Conditioned Oracle

We now define our deployable oracle, which we coin the *acquisition conditioned oracle* (ACO), through introducing a novel objective. As the oracle is defined in terms of full distributional knowledge over the environment ($p(x, y)$), it is unattainable in practice. Later, we describe approximations that allow us to define an AFA policy as an estimate of the ACO.

Our new objective generalizes the greedy, non-deployable optimization in eq. (1) using a weighted (by $\alpha > 0$) cost c for subsets:

$$u(x_o, o) = \operatorname*{arg\,min}_{v \subseteq \{1,\ldots,d\} \setminus o} \mathbb{E}_{\mathbf{y}, \mathbf{x}_v | x_o} \left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha c(v; o). \tag{2}$$

At a high-level, eq. (2) imagines *likely scenarios* of the unacquired feature values and labels (based on conditional dependencies with acquired features), and determines which subset of additional features leads to the greatest expected reduction in prediction loss (adjusting for acquisition costs). The optimization is non-greedy since it measures the expected loss under the acquisition of *subsets* of features. Furthermore, it is deployable since it makes decisions only based on information, (x_o, o), available to the agent.

Note that sequentially minimizing the ACO objective (2) does not yet define a policy. Whenever the ACO minimization (2) returns $u(x_o, o) = \varnothing$, it is clear that there are no further acquisitions that are worth the cost. However, when $u(x_o, o) \neq \varnothing$, the optimization only indicates that there is an expected net benefit (based on the acquired information) to acquiring all the features in $u(x_o, o)$ jointly to make a prediction. As the AFA MDP acquires one feature at a time, the ACO oracle must return a single feature to a acquire. We finalize the ACO policy by proposing to select the feature $j \in u(x_o, o)$ that most minimizes the expected loss $\mathbb{E}_{\mathbf{y}, \mathbf{x}_j | x_o} \left[\ell\left(\hat{y}(x_o, \mathbf{x}_j), \mathbf{y}\right)\right]$ to break ties. In the Appendix

Algorithm 1 Acquisition Conditioned Oracle

```
Input: Observed features $o$ (possibly $o = \varnothing$), instance values $x_o$, distribution $p(x, y)$, estimator $\hat{y}$
Initialize do_predict := false
while $|o| < d$ and not do_predict do
  $u(x_o, o) := \arg\min_{v \subset \{1,\dots,d\} \setminus o} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\big[\ell\big(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\big)\big] + \alpha c(v; o)$
  if $u(x_o, o) = \varnothing$ then
    do_predict = true
  else
    $o := o \cup \{\arg\min_{j \in u(x_o,o)} \mathbb{E}_{\mathbf{y},\mathbf{x}_j|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_j), \mathbf{y}\right)\right]\}$
  end if
end while
Return Prediction $\hat{y}(x_o)$
```

(Section B), we provide a more formal justification for why this selection has desirable properties.

This oracle, in addition to satisfying our two desiderata, provides an intuitive solution to the RL policy optimization by leveraging distributional knowledge over the features and labels to directly navigate the complicated environment and decide useful acquisitions. In contrast, Li & Oliva (2021) use distributional knowledge to guide an RL optimized agent; while that approach is tuned to the AFA MDP, it encounters a more difficult optimization problem.

While the ACO policy does not solve the full RL optimization problem, when casted as the equivalent maximization problem we have that:

Theorem. *(Informal.) The optimal value of the AFA MDP policy is lower bounded by the value of the ACO policy.*

Thus, feature subset acquisition guided by the ACO relates directly to the MDP of interest. A proof of this result can be found in Section C of the Appendix.

3.4. Approximate ACO

There are two main limitations that render the ACO infeasible in practice: 1) the ground truth data distribution is unknown; and 2) the search space over subsets $v \subseteq \{1, \dots, d\} \setminus o$ can be large. Below we discuss approximation techniques to yield an ACO that is deployable in practice. Throughout, we assume that we have a training dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ of n input/output tuples as is common in prior AFA approaches.

First, we note that in practice the data distribution, $p(x, y)$, is unknown and therefore cannot be used in the ACO minimization (2) and resulting policy (Alg. 1). Furthermore, since the data distribution must be conditioned as $p(y, x_v | x_o)$, for $v \subseteq \{1, \dots, d\} \setminus o$ (e.g., for the expectation in eq. (2)), we must be able to condition on *arbitrary* subsets of features o. One approach is to leverage advances in *deep arbitrary conditional* models (Ivanov et al., 2018; Belghazi et al., 2019; Li et al., 2020; Molina et al., 2019; Strauss & Oliva, 2021), which are able to approximate conditional distributions $p(y, x_u \mid x_o)$ for arbitrary subsets u, o. However, this approach requires learning an expensive generative model, which can be challenging due to computation, hyperparameter-optimization, and sample-complexity. Simpler estimators of these (conditional) distributions include k-nearest-neighbors and kernel density estimators (Holmes et al., 2012). Experiments showed that the ACO yielded performant policies by sampling labels and unacquired features through neighbors, i.e. $\mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] \approx \frac{1}{k}\sum_{i \in N_k(x_o)} \ell\left(\hat{y}(x_o, x_v^{(i)}), y^{(i)}\right)$, where $N_k(x_o)$ is the set of k nearest neighbor indices in $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, to x_o (i.e., comparing instances only using features $o \subseteq \{1, \dots, d\}$ to values x_o via a distance function $d(x_o, x_o^{(j)}) \mapsto \mathbb{R}_+$).

Second, because the proposed ACO policy considers all possible additional subsets of features to append to a current set of features o, the optimization in eq. (2) will be over large space when the number of acquirable sets of features is high. We note that minimizing over $v \subseteq \{1, \dots, d\} \setminus o$ can be posed as a discrete optimization problem (as it is equivalent to searching over binary membership indicator vectors); hence, one may deploy a myriad of existing discrete optimization approaches (Parker & Rardin, 2014) including relaxations (Pardalos, 1996), and genetic algorithms (Rajeev & Krishnamoorthy, 1992). In practice, we observed that ACO yielded a performant policy when using a simple upperbound of eq. (2) based on a subsample of potential subsets to minimize over, $\mathcal{O} \subseteq \{v | v \subseteq \{1, \dots, d\} \setminus o\}$, indicating that slightly suboptimal subsets were still effective and enabling embarrassingly parallelizable optimization.

3.5. Parametric Policies

The approximate ACO (AACO) policy, $\hat{\pi}_{\mathrm{ACO}}(x_o, o)$, defined using the approximations above, is a valid nonparametric policy in that, for a new instance drawn at test time, it is deployable and can actively acquire features and make a prediction without using unacquired features or the instance's label. One may, however, want a parametric policy $\pi_\theta(x_o, o)$ that is able to decide what actions to take without having to sample unobserved features and optimizing eq. (2) (e.g. where π_θ is a function stemming from neural network weights θ). Fortunately, mimicking a teacher policy is the focus of the well studied problem of imitation learning (Hussein et al., 2017), where we are able to leverage algorithms such as DAgger (Ross et al., 2011) to supervise and train a parametric policy $\pi_\theta(x_o, o)$ based on the approximate ACO policy $\hat{\pi}_{\mathrm{ACO}}(x_o, o)$. In practice, we observed that a simple behavioral cloning approach (Bain & Sammut, 1995) that directly trains $\pi_\theta(x_o, o)$ based on roll-outs of $\hat{\pi}_{\mathrm{ACO}}(x_o, o)$

was an effective way of supervising the parametric policy.

3.6. AFA for Decision Making

We now consider an important extension to settings where one wishes to actively acquire features to determine a *decision* that will maximize an *outcome*, rather than for determining a *prediction* to match a *label*, as before. For example, actively acquiring information about a patient (e.g., running blood tests, getting biopsies, etc.) to determine a treatment (e.g., choose a drug) that will maximize the outcome (e.g., based on mortality). A fundamental challenge in constructing algorithms for decision tasks is that typically the outcome under only one decision is ever observable for a given instance. Therefore, decision making requires constrasting counterfactuals that are unobserved in the training data. The construction of policies to make optimal decisions from data with *fully observed* contexts has been extensively studied in statistics, and operations research under the titles dynamic treatment regimes (DTRs) or individualized treatment rules (ITRs) (Murphy, 2003; Kosorok & Laber, 2019).

Below, we develop a novel policy for general decision-making with AFA through analogies between components in a causal inference setup and ACO for prediction. Let $y(a)$ denote the potential outcome (Rubin, 2005) under decision (intervention) $a \in \mathcal{A}$, where we assume larger values of $y(a)$ are better without loss of generality. It can be shown that, under certain conditions (see Appendix), one can train a partially observed decision-making policy $\hat{\pi}_{\mathcal{A}}(x_o)$ which maps observed feature values x_o to interventions to maximize $\mathbb{E}[y(\hat{\pi}_{\mathcal{A}}(x_o))]$. Note that this decision-making policy $\hat{\pi}_{\mathcal{A}}(x_o)$ is analogous to $\hat{y}(x_o)$, the classifier given partially observed inputs, since $\hat{\pi}_{\mathcal{A}}(x_o)$ similarly maps partially observed inputs to outputs. Moreover, one can also train an estimator, $\hat{Q}(x, a)$, of $\mathbb{E}[Y(a) \mid x]$, the expected outcome for an instance x with action a. $-\hat{Q}(x, a)$ is akin to a loss function ℓ as it judges the effectiveness of an output a for an instance. Leveraging $\hat{Q}(x, a)$, and $\hat{\pi}_{\mathcal{A}}(x_o)$, it is now possible to utilize a AACO to construct a decision-making acquisition policy $\pi_{\text{acq}}(x_o)$, which determines what (if any) new features are worth acquiring in order to make a better decision (one that shall yield a higher expected outcome). Analogously to eq. (2), we may minimize:

$$\arg\min_{v \subseteq u} -\mathbb{E}_{\mathbf{x}_u|x_o}\left[\hat{Q}\left((x_o, \mathbf{x}_u), \hat{\pi}_{\mathcal{A}}(x_o, \mathbf{x}_v)\right)\right] + \alpha c(v; o), \quad (3)$$

where $u = \{1, \ldots, d\} \setminus o$, determining if acquiring new features v will lead to better decisions (made by $\hat{\pi}_{\mathcal{A}}$ and assessed using $\hat{Q}$). As before, the ACO's policy $\pi_{\text{acq}}(x_o)$ stemming from eq. (3) will determine what new features to sequentially acquire until reaching a subset o' for which no new acquisitions are worth the cost, at which time we choose an intervention according to $\hat{\pi}_{\mathcal{A}}(x_{o'})$. To the best of our knowledge, this represents the first oracle based approach for AFA decision-making with observational (offline) data.

4. Experiments

We perform extensive experiments to assess the AACO's performance relative to alternative AFA methods, characterize the influence of approximations and policy-design decisions on performance, and demonstrate how AFA can be utilized for decision making.

Performance Measurement AFA methods face a trade-off between task performance and acquisition costs. Therefore, we measure performance by reporting the methods' inference results across different values of α (2) (as distinct ticks, e.g. Fig. 3), assuming equal feature costs. Note that the AACO, unlike some alternatives, acquires different numbers of features for different instances at a given cost depending on the complexity of the instance's task.

Comparisons Many alternative AFA methods, especially ones leveraging generative models, are complex to train, and therefore performance can be heavily dependent upon implementation details. Thus, to facilitate more fair comparisons, we compare our experimental results against results from the original sources (Li & Oliva, 2021). Because they represent a variety of recognized approaches in AFA and have experimental results recorded on several benchmark datasets, we throughout compare to: `JAFA` (Shim et al., 2018), which jointly trains a deep learning RL agent (using Q-Learning) and a classifier for AFA; `GSMRL` (Li & Oliva, 2021), which learns a generative surrogate arbitrary conditioning model to derive auxiliary information and rewards that are used to train a deep learning agent; `EDDI` (Ma et al., 2018), a greedy policy that estimates the information gain for each candidate feature using a VAE-based model with a set embedding and selects one feature with the highest expected utility at each acquisition step; and `GSM+Greedy` (Li & Oliva, 2021), which similarly acquired features greedily using a surrogate arbitrary conditioning model that estimates the utility of potential feature acquisitions. Please refer to (Shim et al., 2018; Li & Oliva, 2021) for baseline hyperparameters details. In the Appendix (Section A.3), we also compare to a variation in which the AACO is used as a teacher policy in behavior cloning (`AACO + BC`, 3.5). As simpler alternatives, we also compare to a baseline classifier using all features and a static classifier (`STATIC`) where the same features were used for each instance and chosen based on their permutation feature importance.

Implementation Details AFA (prediction) methods require a predictor $\hat{y}(x_o)$ that can make predictions on arbitrary subsets of features. We found that gradient boosted trees tended to perform well and were therefore used for our AACO implementations as well as for the baseline and static classifiers. Moderate feature dimensions (≤ 10) al-

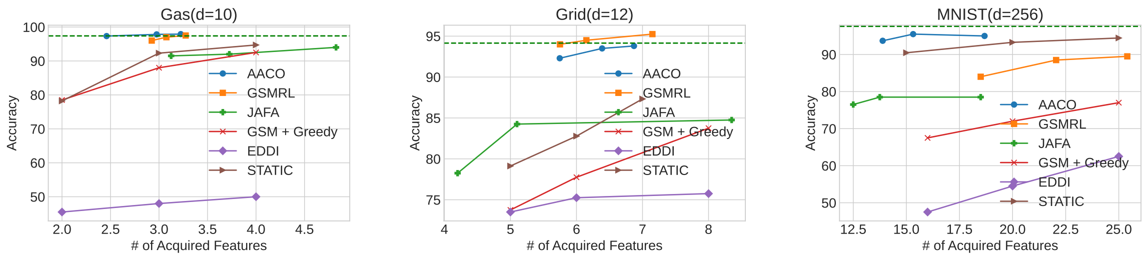
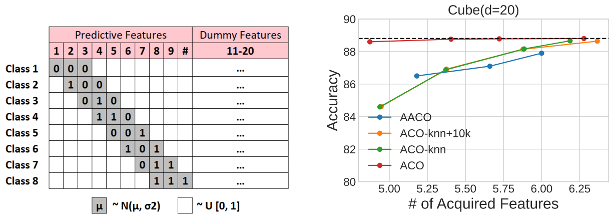


Figure 2. Test accuracy on Real-World datasets. Full-feature classifier accuracy denoted by dashed line.

lowed for feasible training and caching of separate predictors for each possible subset of features. For higher dimensional acquisition spaces, we utilized a masking strategy where the feature vector, with unobserved entries imputed with a fixed value, was concatenated with a binary mask indicating the indices of observed entries (Li et al., 2020). During training, these masks were drawn at random to simulate making predictions with missing features. For the AACO approximations (3.4), we approximated the distribution of $p(y, x_u|x_o)$ in AACO using a $k = 5$ nearest neighbors density estimate. Furthermore, we enumerated all potential subsets in moderate dimensional problems but took random subsamples (10,000) in higher dimensions. See 4.3 for a discussion of sensitivity to this choice. Performance measures were evaluated on a test set independent of the training sets used for the predictor and density estimator training. Code for the AACO policy can be found at https://github.com/lupalab/aaco.

4.1. Cube Dataset



	Predictive Features										Dummy Features
	1	2	3	4	5	6	7	8	9	#	11-20
Class 1	0	0	0								...
Class 2		1	0	0							...
Class 3			0	1	0						...
Class 4				1	1	0					...
Class 5					0	0	1				...
Class 6						1	0	1			...
Class 7							0	1	1		...
Class 8								1	1	1	...

μ ~ N(μ, σ2) □ ~ U [0, 1]

Figure 3. Left: Distribution of features in CUBE-σ. Right: Accuracy, with multiple ticks correspond to different cost scales α.

We begin with a study of the CUBE-$\sigma = 0.3$ dataset (as described by Shim et al. (2018)), a synthetic classification dataset designed for feature acquisition tasks. We consider a $d = 20$-dimensional version with 8 classes. Each data point consists of 17 uniform and 3 normally distributed features. For a data point of class k, features k through $k + 2$ are normally distributed, with means as shown in Fig. 3; remaining features are uniformly distributed.

The synthetic environment allows us to better isolate the effect each approximation (of the ACO) has on the AACO's performance. We compare the following policies: ACO (Alg. 1), which is known in this environment (up to importance-sampling error); ACO-knn, identical to ACO except it approximates $p(y, x)$ (used implicitly in eq. (2)) with a nearest neighbors estimate; ACO-knn+10k, identical to ACO-knn except that is chooses a random subset of $|\mathcal{O}| = 10,000$ subsets of features to search over; and AACO, which is identical to ACO-knn+10k except that it no longer has access to the Bayes-rule classifier to make predictions (during eq. (2) and at prediction time). From the results in Fig. 3, we see that ACO performs admirably, achieving near-optimal predictions with under 5 features acquired on average. Encouragingly, ACO-knn+10k, despite not having searching over all possible (2^{20}) subsets achieves results near identical to ACO-knn, suggesting that the search approximation may not lead to significant decreases in accuracy.

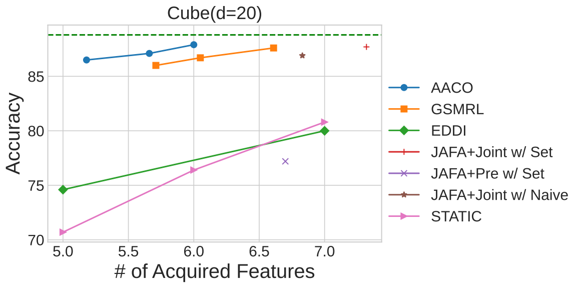


Figure 4. Cube Results; Full-feature classifier accuracy denoted by dashed line.

We also compare to previous AFA approaches on the CUBE-0.3 dataset (as reported by Shim et al. (2018)), which include: JAFA++ variations (Shim et al., 2018) and GSMRL (Li & Oliva, 2021). In Fig. 3, we see that AACO performs, despite not using deep RL methodology.

4.2. Real World Datasets

Next we perform experiments on real-world datasets stemming from the UCI ML repository and MNIST, for which experimental results from a multitude of alternative approaches exist. Results of JAFA, GSMRL, EDDI, GSM+Greedy are reported from (Li & Oliva, 2021). Due to the higher dimensionality of MNIST, *most baselines are unable to scale*, and hence we consider a downsampled 16×16 version were $d = 256$. (See below for an ablation on the full-dimensional MNIST.)

We observe across datasets that our AACO approach often outperforms the other methods (Fig. 2). This is especially impressive considering that most of the baselines utilize complicated deep-learning approaches, whereas the AACO utilizes simple nonparametric techniques. We also note that the performance of the parametric policy AACO+BC is often competitive with that of AACO, despite being supervised using a relatively simple behavioral cloning approach (Fig. 11). Consistent with the findings of (Covert et al., 2023), dynamic approaches to AFA don't always outperform simpler static baselines, which underscores the complexity of the task as well as the promise our new objective has for effectively navigating the complex policy space.

Psychological Assessments Psychological assessments based on survey responses provide a compelling use-case. Here AFA policies (ideally) dynamically determine a small personalized subset of questions to assess an individual without the need of a lengthy survey, reducing user fatigue and potentially improving accuracy (Early et al., 2016a). We consider the "Big Five Personality Test" [(OSPP, 2023)], which consists of 50 questions, each asking the user to rate their agreement with a given statement on an ordinal scale with 5 options. Questions pertain to one of the "big five" personality traits referenced in academic psychology. We attempt to classify each survey instance by the quartile (leading to 4 classes) of the associated *emotional stability (ES) score* (based on the sum of 10 ES questions) with respect to the overall population of survey takers. Given its consistent performance in the previous comparisons, we compare to GSMRL in this task. We also compare to a Rescale baseline that uses ground-truth information to randomly sub-select from the 10 ES questions. We can see (Fig. 5) that there is a high amount of variance in responses to ES questions; e.g., Rescale has considerable ambiguity between quartiles even when acquiring a majority of pertinent questions (6). In contrast, AACO is better able to leverage feature dependencies (without any ground-truth annotations) and also outperforms GSMRL.

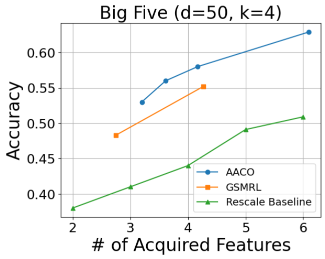


Figure 5. Big 5 results.

4.3. Ablations

Through ablation studies, we provide empirical results that investigate the sensitivity of approximation choices (3.4), compare our novel AFA objective to greedy alternatives, and provide promising evidence for the AACO's ability to scale to high (acquisition) dimension tasks. Extended discussion and full results can be found in the Appendix.

Approximations As discussed above (3.4), implementing the ACO requires two approximations. First, full distributional knowledge ($p(y, x)$) is unknown. Consequently, the conditional expected loss of acquiring an additional subset of features, which forms the basis for the objective in eq. (2), must be estimated. Our experiments demonstrate that a simple k nearest neighbors density estimate already performs well, and we found that the results were relatively insensitive to the choice of k in a range from 5 to 50 (Table 1). Preliminary attempts to instead use a deep arbitrary conditional model were largely unsuccessful, perhaps highlighting the challenges involved with their training. The second approximation, present in higher dimensional settings, avoids searching all possible subsets of acquirable features. Remarkably, ablations demonstrate that uniformly sampling subsets can be effective once just a moderate (~ 100) number of subsamples were considered (Fig. 10).

Greedy/Cheating Comparison Our ACO objective eq. (2) differs from greedy alternatives in that it minimizes an expected loss of a *sequence* of future acquisitions. To better understand these differences, we compare AACO with variations that modify eq. (2) to only search over subsets with one extra feature (greedy) or terminate the acquisition process at the same number of features for all instances. In both cases, we find the AACO to be performant (Fig. 11). Additionally, we extensively compare to previous oracle approaches that are not deployable because they observe information (y and x_u) that is not accessible to an AFA agent ("cheating"). Thus, utilizing these cheating oracles requires supervising a parametric policy. We find that the AACO and AACO with behavior cloning outperform the policies supervised by cheating oracles, regardless of whether the oracle greedily acquired features or not (Fig. 11). This provides evidence that the benefit of our ACO (compared to the greedy oracle) extends beyond its non-greedy objective, suggesting that a deployable oracle (one that utilizes the same information as the student) can be more easily emulated.

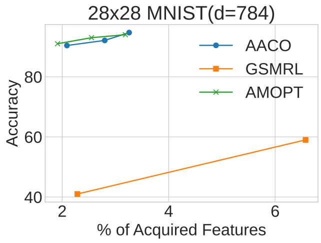


Figure 6. $786d$ MNIST. AACO and AMOPT are SOTA for this high dimensional task.

Scaling As noted by Li & Oliva (2021), most existing AFA baselines fail to scale to higher dimensional settings, such as the full 28×28 MNIST dataset. Indeed, even GSMRL, which is able to learn a reasonable policy in the 784-d MNIST, actually sees a degradation of performance when compared to the policy in 256-d MNIST (Fig. 2). A notable exception is Covert et al. (2023) (AMOPT), whose greedy policy achieved near 90% accuracy after only 10 pixel acquisitions, a result that is (to our knowledge) the best performance yet. In spite of its relative simplicity (no deep neural network architectures or challenging optimization procedures), we find comparable SOTA from the AACO policy (Figure 6). This finding, as well as the relative success on the 256-d MNIST, provide evidence that the ACCO policy can successfully navigate the high dimensional feature spaces that pose considerable challenges to alternative methods.

4.4. Decision Making

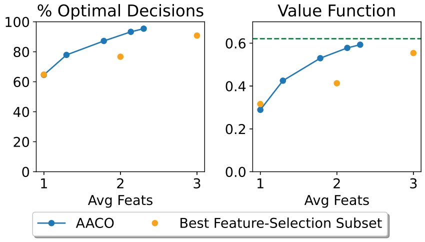


Figure 7. Results on synthetic data. Left: % of time policy made the ground-truth optimal decision. Right: value function, with optimal value given by dashed line.

We investigate the ability of the AACO's policy $\pi_{\text{acq}}(x_o)$ to learn what features are relevant for making decisions that maximize expected outcomes according to a decision policy $\hat{\pi}_{\mathcal{A}}(x_o)$. We first evaluate its performance under a synthetic environment, where the ground-truth optimal decisions are known (i.e. $y(a)$ is known for every $a \in \mathcal{A}$). We generate four features $x_0, .., x_3$, a treatment $a \in \{0, 1\}$, and an outcome y that depends on $x_0, ..., x_3$ and a. Full details of the environment are provided in the Appendix. As depicted in Figure 7, we find that increasing acquisition costs prompts the AACO policy to choose relevant features for decision making. By construction, an average of 2.25 features are sufficient for optimal decision making. At similar levels of AACO acquired features, the partially-observed decision-making policy ($\hat{\pi}_{\mathcal{A}}(x_o)$) attains near-perfect decision making with a value function ($\mathbb{E}[y(\hat{\pi}_{\mathcal{A}}(x_o))]$) approaching the value function of the full-context optimal decision policy (green line). As AFA in this setting has been previously understudied, here we compare to a feature selection approach that uses a *constant* subset of features (depicted in orange). This highlights AACO's better performance with its ability to dynamically acquire relevant features per-instance.

Real-World Commerce Data Next, we demonstrate the practical utility of our method in the contextual bandit setting, which encompasses many decision making problems such as internet adverting (Chapelle & Li, 2011), content recommendation (Agarwal et al., 2013), and medical treatment strategies (Kosorok & Laber, 2019). When contexts are associated with a *cost* (possibly in the form of time, computation, or money), a fruitful strategy is to acquire features based on their relevance for decision making. We illustrate a novel application of this end-goal using the Open Bandit Dataset (Saito et al., 2020), which contains data from a large-scale fashion e-commerce platform and is a common benchmark for off-policy evaluation. Our objective is to actively acquire a user's context (features) to recommend a product (clothing items) that maximizes the expected click-rate by the user. The user's context, which is collected at a cost per feature, consists of user demographic data as well as user-specific affinity scores induced by historic clicks. The data set consists of over 1 million instances with 65 features. For simplicity, the action space is restricted to the two most frequent clothing item recommendations.

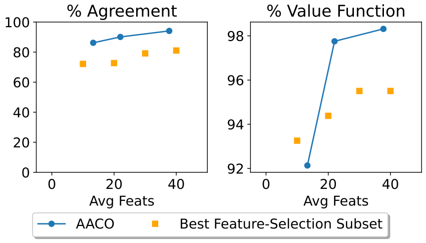


Figure 8. Left: % agreement (in decision-making) with the full feature policy. Right: estimated value function (as % of the full-feature value function).

In this real-world setting, the ground-truth optimal decisions are unknown. Thus, we measure the effectiveness of our methods relative to a learned recommendation policy based on full contexts. As a baseline, we compare our strategy to a natural cost-aware alternative, feature selection, in which a fixed subset of features are used by the decision-policy for all instances. We see (Fig. 8) that the partially-observed decision policy under AACO makes the same decisions as the full-context policy over 90% of the time with only 20 features, has an estimated value close to the full-context policy, and generally outperforms the best feature selection alternative. This highlights the ability of the policy to make satisfactory decisions without observing full contexts and to tailor the acquired contexts to each instance.

5. Conclusion

The ACO represents a rethinking of ideas in AFA methodology by introducing a novel objective leading to a new, effective method. ACO directly identifies a general optimization problem eq. (2) that yields a deployable policy. We believe that the ACO optimization encapsulates the crux of the AFA problem, allowing one to achieve SOTA performance with simple approximations, even in higher dimensionalities. Lastly, we showed that our ACO framework seamlessly extends to a setting where we are acquiring features not for prediction, but instead for decision making with general observational training data.

While the ACO framework provides an intuitive formulation of the AFA challenge, further work is needed to understand when it might be suboptimal compared to the MDP formulation in **AFA MDP**. Furthermore, the approximations evaluated in this paper (using neighbors to approximate the expected conditional loss, evaluating the loss at random subsets of unacquired features) are elementary; however, we are encouraged by their effectiveness, and exploring what additional benefit might be derived from more complex approximations is an exciting endeavor.

Acknowledgements

We extend our gratitude to the reviewers for their valuable feedback and insights. Special thanks to our colleagues and mentors for their guidance throughout the process. This research was partly funded by NSF grants IIS2133595, DMS2324394, and by NIH grants 1R01AA02687901A1, 1OT2OD032581-02-321.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, D., Chen, B.-C., Elango, P., and Ramakrishnan, R. Content recommendation on web portals. *Commun. ACM*, 56(6):92–101, jun 2013. ISSN 0001-0782. doi: 10.1145/2461256.2461277. URL https://doi.org/10.1145/2461256.2461277.

Bain, M. and Sammut, C. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Belghazi, M., Oquab, M., and Lopez-Paz, D. Learning about an exponential amount of conditional distributions. *Advances in Neural Information Processing Systems*, 32, 2019.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf.

Covert, I. C., Qiu, W., Lu, M., Kim, N. Y., White, N. J., and Lee, S.-I. Learning to maximize mutual information for dynamic feature selection. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6424–6447. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/covert23a.html.

Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., and Coppin, B. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

Early, K., Fienberg, S., and Mankoff, J. Cost-effective feature selection and ordering for personalized energy estimates. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016a.

Early, K., Fienberg, S. E., and Mankoff, J. Test time feature ordering with focus: Interactive predictions with minimal user burden. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 992–1003, 2016b.

Fu, Y., Zhu, X., and Li, B. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283, 2013.

Gong, W., Tschiatschek, S., Nowozin, S., Turner, R. E., Hernández-Lobato, J. M., and Zhang, C. Icebreaker: Element-wise efficient information acquisition with a bayesian deep latent gaussian model. In *Advances in Neural Information Processing Systems*, pp. 14820–14831, 2019.

He, H., Eisner, J., and Daume, H. Imitation learning by coaching. In *Advances in Neural Information Processing Systems*, pp. 3149–3157, 2012.

He, H., Mineiro, P., and Karampatziakis, N. Active information acquisition. *arXiv preprint arXiv:1602.02181*, 2016.

Holmes, M. P., Gray, A. G., and Isbell, C. L. Fast nonparametric conditional density estimation. *arXiv preprint arXiv:1206.5278*, 2012.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Ivanov, O., Figurnov, M., and Vetrov, D. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

Khaire, U. M. and Dhanalakshmi, R. Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, 34(4): 1060–1073, 2022.

Konyushkova, K., Sznitman, R., and Fua, P. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pp. 4225–4235, 2017.

Kosorok, M. R. and Laber, E. B. Precision medicine. *Annual Review of Statistics and Its Application*, 6(1):263–286, 2019. doi: 10.1146/annurev-statistics-030718-105251. URL https://doi.org/10.1146/annurev-statistics-030718-105251.

Li, Y. and Oliva, J. Active feature acquisition with generative surrogate models. In *International Conference on Machine Learning*, pp. 6450–6459. PMLR, 2021.

Li, Y., Akbar, S., and Oliva, J. Acflow: Flow models for arbitrary conditional likelihoods. In *International Conference on Machine Learning*, pp. 5831–5841. PMLR, 2020.

Liyanage, Y. W. and Zois, D. Optimum feature ordering for dynamic instance–wise joint feature selection and classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3370–3374, 2021. doi: 10.1109/ICASSP39728.2021.9414669.

Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., and Zhang, C. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018.

MacKay, D. J. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

Madasu, A., Oliva, J., and Bertasius, G. Learning to retrieve videos by asking questions. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 356–365, 2022.

Miao, J. and Niu, L. A survey on feature selection. *Procedia Computer Science*, 91:919–926, 2016.

Minsky, M. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

Molina, A., Vergari, A., Stelzner, K., Peharz, R., Subramani, P., Di Mauro, N., Poupart, P., and Kersting, K. Spflow: An easy and extensible library for deep probabilistic learning using sum-product networks. *arXiv preprint arXiv:1901.03704*, 2019.

Murphy, S. A. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003. doi: https://doi.org/10.1111/1467-9868.00389. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00389.

OSPP. Take a personality test - Open Source Psychometrics Project, 2023. URL https://openpsychometrics.org/.

Pardalos, P. M. Continuous approaches to discrete optimization problems. *Nonlinear optimization and applications*, pp. 313–325, 1996.

Parker, R. G. and Rardin, R. L. *Discrete optimization*. Elsevier, 2014.

Rajeev, S. and Krishnamoorthy, C. Discrete optimization of structures using genetic algorithms. *Journal of structural engineering*, 118(5):1233–1250, 1992.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Rubin, D. B. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005. ISSN 01621459. URL http://www.jstor.org/stable/27590541.

Rückstieß, T., Osendorfer, C., and van der Smagt, P. Sequential feature selection for classification. In *Australasian Joint Conference on Artificial Intelligence*, pp. 132–141. Springer, 2011.

Saar-Tsechansky, M., Melville, P., and Provost, F. Active feature-value acquisition. *Management Science*, 55(4): 664–684, 2009. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/40539177.

Saito, Y., Shunsuke, A., Megumi, M., and Yusuke, N. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*, 2020.

Shim, H., Hwang, S. J., and Yang, E. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in neural information processing systems*, pp. 1368–1378, 2018.

Strauss, R. and Oliva, J. B. Arbitrary conditional distributions with energy. *Advances in Neural Information Processing Systems*, 34:752–763, 2021.

Venkatesh, B. and Anuradha, J. A review of feature selection and its methods. *Cybernetics and information technologies*, 19(1):3–26, 2019.

Yoo, D. and Kweon, I. S. Learning loss for active learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2019.

Zubek, V. B. and Dietterich, T. G. Pruning improves heuristic search for cost-sensitive learning. In *ICML*, 2002.

A. Ablations

A.1. Evidence for Adaptive Feature Acquisition

In active feature acquisition, the agent (AFA policy) sequentially acquires a dynamic (on a per instance basis) subset of features that minimize acquisition costs whilst yielding accurate inferences. Therefore, a couple of the salient aspects of the AFA paradigm are determining *which* and *how many* features are relevant to make an accurate prediction for a particular instance. In tasks exhibiting significant heterogeneity, one might expect different features were more or less relevant for different groups or that different groups needed more features to feel achieve the requisite confidence in prediction.

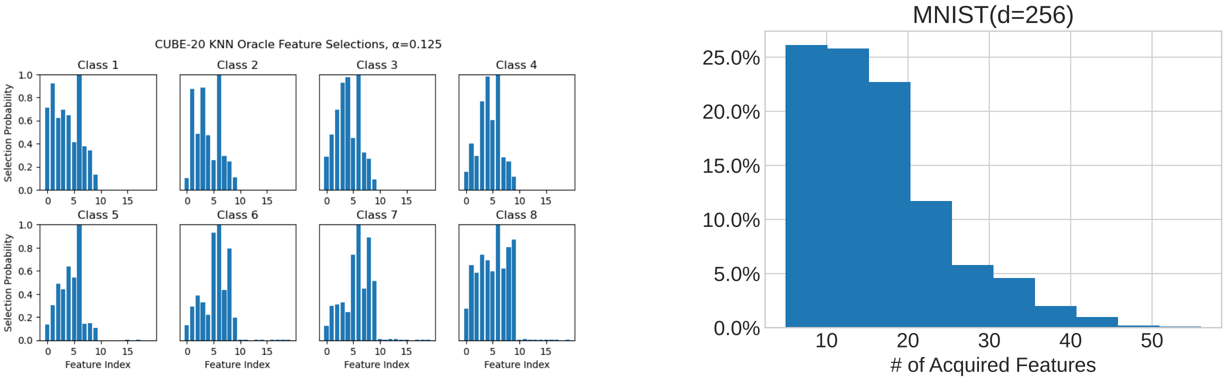


Figure 9. Left: Acquisition actions taken by AACO, expressed as a histogram per class for each feature. I.e., we report the portion of the time that a feature was acquired on average for making a prediction for instances of each class. Right: A histogram of the number of selected features per instance for the AACO policy on the MNIST-256 dataset (16.635 selected features on average).

We provide evidence that our AACO policy is able to successfully distinguish which features are most relevant for prediction in the synthetic `CUBE` dataset. In Fig. 9, we report the portion of the time that a feature was acquired on average for making a prediction for instances of each class. Note that, in addition to not acquiring the noise features 10-19 in most cases, the agent tends to focus on the most defining features for each specific class, as indicated by Fig. 3 in the main paper. (Here feature 6 was deterministically chosen as the initial feature.)

To highlight the AACO's dynamic nature in allowing for variable numbers of acquisitions per instance, we examine the distribution of the number of features acquired for the MNIST-256 experiment (Fig. 9). While a grand majority of instances terminate the acquisition process with fewer than 20 features, a small minority of the instances exceed 40 feature acquisitions. Inspection of the average number of features acquired by the (true) label of the instance, we find that images of the digit "1" had the fewest features acquired on average (10.74) while the digit "8" had the most features acquired on average (20.37).

A.2. Role of Approximations

As previously discussed (3.4), the AACO makes two approximations compared to the ACO. The first approximation is that full distributional knowledge ($p(y, x)$) is unknown. Consequently, the conditional expected loss of acquiring an additional subset of features, which forms the basis for the objective in eq. (2), is unknown and must be estimated. Our experiments demonstrate that a simple a k nearest neighbors density estimate already performs well. To assess the sensitivity of the AACO's performance to the number of neighbors k, we ran an experiment on the `CUBE` dataset where k was varied (while the cost remained fixed). The results are shown in Table 1; accuracy and feature selection are fairly stable across the choice of k. Similar trends hold across other data. While a deep arbitrary conditional model would also be used to estimate the unknown density (and a Monte Carlo approximation made to the integral by sampling from the model), preliminary attempts to learn a deep arbitrary conditional model were largely unsuccessful, perhaps highlighting the challenge involved with their training.

The second approximation involves the search over subsets posed by the minimization in eq. (2). When the dimension of the acquirable sets of features is moderate, as in the Gas and Grid experiments, it is feasible to search over all possible subsets and find the minimizing subset $u(x_o, o)$. However, when the cardinality of (sets of) acquirable features in large, such brute-force optimization is impractical. While minimizing over $v \subseteq \{1, \ldots, d\} \setminus o$ can be posed as a discrete optimization problem, our experiments were conducted using a simple random sample of $|\mathcal{O}|$ subsets ($|\mathcal{O}| = 10,000$ for the `CUBE` and

Table 1. Ablation on the sensitivity of the AACO to the number of neighbors k used on the `CUBE` dataset.

Neighbors	Accuracy	Features
k=5	0.846	5.824
k=10	0.856	5.916
k=25	0.85	5.576
k=50	0.842	5.49

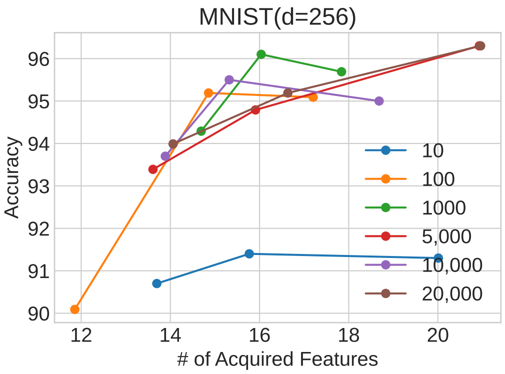


Figure 10. A comparison of the predictive performance of the AACO policy under different cardinalities of $\mathcal{O}$ on the MNIST-256 dataset, with three cost scales α used for each cardinality. 10,000 is the original mask size used in other experiments. We find that even with only a small number of sampled subsets (100 and above), the prediction accuracy is fairly consistent.

MNIST examples). In this ablation, we investigate the sensitivity of the AACO's predictive performance as we vary this hyperparameter. Because of its high dimensionality, the MNIST-256 dataset is used for the experiment. As seen in Figure 10, the prediction accuracy is consistent even under a small number ($\sim$ 100) of sampled subsets, a promising finding that provides some evidence of the method's robustness to this parameter. Because the computational complexity scales linearly with $|\mathcal{O}|$, reducing the number of subsets searched can lead to significant gains in the speed of inference.

A.3. Greedy/Cheating Comparisons

The ACO policy bypasses many of these computational challenges faced by deep RL policies while still optimizing a non-greedy objective. While challenging to study theoretically, we find that the AACO's empirical performance rivals and often exceeds that of these competitor approaches across a wide array of empirical benchmarks. In this ablation, we try to better understand which components of the AACO policy might be responsible for its empirical success.

Our ACO objective (eq. (2)) differs from greedy alternatives in that it minimizes an expected loss of not just the next acquisition but a *sequence* of future acquisitions, where the benefit of the sequence of acquisitions is weighed against the cost of acquiring them. Ideally, this design choice allows the ACO to (1) favor features that are jointly informative and (2) tailor the number of acquisitions to the complexity of the instance's task. To assess potential benefits from (1), we compare the AACO policy to a variant that uses the following greedy modification of eq. (2)

$$u(x_o, o) = \underset{j \in \{1,\ldots,d\} \setminus o}{\arg\min} \; \mathbb{E}_{\mathbf{y},\mathbf{x}_j | x_o} \left[\ell \left(\hat{y}(x_o, \mathbf{x}_j), \mathbf{y} \right) \right] + \alpha. \tag{4}$$

Compared to eq. (2), this greedy objective only searches over subsets that contain one more feature than is in o. We title this (approximated) policy `AACO Greedy`. We also compare to a variant of the ACO policy (`AACO Fixed Acquisitions`) that always terminates after fixed number of acquisitions, regardless of whether $u(x_o, o) = \varnothing$ or not.

Another aspect that differentiates our objective from some other retrospective oracle approaches (He et al., 2012; 2016; Madasu et al., 2022), which greedily determine what next feature would ideally be acquired for a particular instance x with corresponding label y given an observation set o of already acquired features, is that the ACO solving the objective is *deployable*. That is, the oracle can be deployed to solve eq. (2) while respecting the AFA paradigm. On the contrary, previous retrospective oracle approaches ”cheat” by performing a search over information that is not available to the agent. Beyond necessitating some form of imitation learning to construct a valid AFA policy, we speculate these oracles are harder to emulate than a deployable oracle, i.e. one that utilizes the same information as the student. To investigate this, we compare the AACO to a policy learned from behavior cloning that emulates an oracle that chooses the next feature using either (`Cheating + BC`)

$$i \in \underset{v \subseteq \{1,\ldots,d\} \setminus o}{\arg\min} \; \ell\left(\hat{y}(x_{o \cup v}), y\right) + \alpha|v|$$

or (`Greedy Cheating + BC`)

$$i = \underset{j \in \{1,\ldots,d\} \setminus o}{\arg\min} \; \ell\left(\hat{y}(x_{o \cup \{j\}}), y\right) + \alpha.$$

Note that the lack of an expectation means that the oracle, for a given instance, is using that instance’s full features and label.

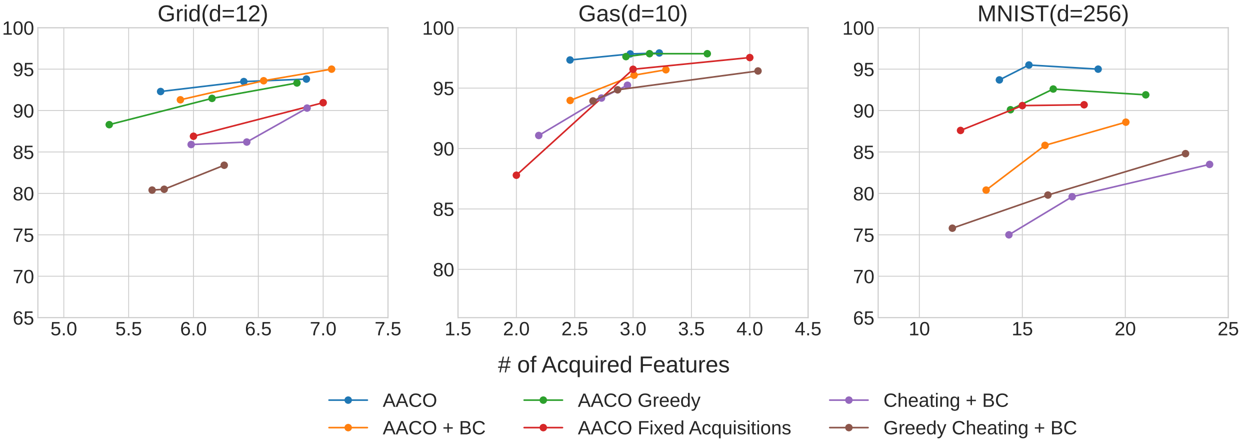


Figure 11. A comparison of the AACO policy with alternative modifications on the Grid, Gas, and MNIST-256 datasets..

Results from these experiments are displayed in Fig. 11. These experiments provide evidence that the empirical success of AACO can be attributed to at least several factors. First, we find that a non-greedy objective (comparing `AACO` to `AACO Greedy`) provides benefits in some contexts, such as the MNIST example. We also find that having using behavioral cloning generally leads to approximation error that reduces performance compared to a deployable oracle. Finally, the poor performance of the cheating oracles (`Cheating + BC` and `Greedy Cheating + BC`) relative to `AACO + BC` suggests that there is benefit to emulating an oracle that only operates over the same information as a (valid) AFA policy.

A.4. Scaling

As noted by Li & Oliva (2021), most existing AFA baseslines fail to scale to higher dimensional settings, such as the full 28 × 28 MNIST dataset. With 784-d MNIST `JAFA` struggles to learn a policy that selects a small number of features (and instead learns to select all or no features) (Li & Oliva, 2021). Furthermore, greedy methods like `EDDI` and `GSM+Greedy` are unable to scale their searches. Indeed, even `GSMRL`, which is able to learn a reasonable policy in the 784-d MNIST, actually sees a degradation of performance when compared to the policy in 256-d MNIST (Fig. 2). A notable exception is Covert et al. (2023) (`AMOPT`), whose greedy policy achieved near 90% accuracy after only 10 pixel acquisitions, a result that is (to our knowledge) the best yet performance. In spite of its relative simplicity (no deep neural network architectures or challenging optimization procedures), we find comparable SOTA from the AACO policy (Figure 6). This finding, as well as the relative success on the 256-d MNIST, provide evidence that the ACCO policy can successfully navigate the high dimensional feature spaces that pose considerable challenges to alternative methods.

B. Internal Consistency of Policy's Acquisition

In 3.4, we discussed how sequentially minimizing the ACO objective (2) does not yet define a policy. Specifically, whenever $u(x_o, o) \neq \varnothing$, the optimization only indicates that there is an expected net benefit to acquiring $u(x_o, o)$ but does not provide concrete guidance on which feature $j \in \{1, \dots, d\} \setminus o$ would be best to acquire, which is necessary to define an AFA policy.

One desirable property a policy in this setting ideally has is that it is "internally consistent" with respect to (2) in the sense that it never selects a feature $j \in \{1, \dots, d\} \setminus o$ that leads to higher expected loss. This is, it will never select j if

$$\begin{aligned}
&\mathbb{E}_{x_j|x_o}\left[\min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o,x_j}\left[\ell\left(\hat{y}(x_o, x_j, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|\{j\} \cup v|\right] \\
&> \min_{v \subseteq \{1,\dots,d\} \setminus o} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|v|.
\end{aligned}$$

The term on the left corresponds to the expected loss if j were selected and the optimization (2) were repeated.

Proposition B.1. *If predictions are made according to the Bayes rule classifier, $y \perp x_j | x_o, x_v$, and $x_j \perp x_v | x_o$ for all $v \subseteq \{1, \dots, d\} \setminus o \cup \{j\}$, then selecting j is **not** internally consistent when $\hat{y}$ corresponds to the Bayes rule classifier.*

Proof. We have that

$$\begin{aligned}
&\mathbb{E}_{x_j|x_o}\left[\min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o,x_j}\left[\ell\left(\hat{y}(x_o, x_j, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|\{j\} \cup v|\right] \\
&= \mathbb{E}_{x_j|x_o}\left[\min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|\{j\} \cup v|\right] \\
&= \min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{x_j|x_o}\left[\mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|\{j\} \cup v|\right] \\
&= \min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{x_j|x_o}\left[\mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o,x_j}\left[\ell\left(\hat{y}(x_o, x_j, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|\{j\} \cup v|\right] \\
&= \min_{v \subseteq \{1,\dots,d\} \setminus o \cup \{j\}} \mathbb{E}_{\mathbf{y},\mathbf{x}_v,x_j|x_o}\left[\ell\left(\hat{y}(x_o, x_j, \mathbf{x}_v), \mathbf{y}\right) + \alpha|\{j\} \cup v|\right] \\
&> \min_{v \subseteq \{1,\dots,d\} \setminus o} \mathbb{E}_{\mathbf{y},\mathbf{x}_v|x_o}\left[\ell\left(\hat{y}(x_o, \mathbf{x}_v), \mathbf{y}\right)\right] + \alpha|v|
\end{aligned}$$

□

To follow the steps of the proof, note that because $y \perp x_j | x_o, x_v$, we have that $\hat{y}(x_o, x_j, x_v) = \hat{y}(x_o, x_v)$. Furthermore, the independence statements $y \perp x_j | x_o, x_v$ and $x_j \perp x_v | x_o$ imply that the density $p(y, x_v | x_o, x_j)$ reduces to $p(y, x_v | x_o)$. This demonstrates the first equality. The second equality trivially holds since the terms inside the outer expectation do not involve x_j. The third equality reverses the first equality using the same logic, while the fourth equality is due to $p(y, x_v | x_o, x_j) p(x_j | x_o) = p(y, x_v, x_j | x_o)$. Finally, the inequality is due to the fact that the expected loss is the same, but the left hand term has an additional cost term ($\alpha c(o \cup j \cup v) > \alpha c(o \cup v)$

Proposition B.2. *If $y \perp x_j | x_o, x_v$, and $x_j \perp x_v | x_o$ for all $v \subseteq \{1, \dots, d\} \setminus o \cup j$, then $j \notin u(x_o, o)$.*

We note that choosing $j \in u(x_o, o)$ does not guarantee that it is internally consistent. However, it does prevent the undesirable scenario of selecting a variable that which is irrelevant (in the sense that $y \perp x_j | x_o, x_v$, and $x_j \perp x_v | x_o$ for all $v \subseteq \{1, \dots, d\} \setminus o \cup \{j\}$), which would lead to an internally inconsistent selection acquisition.

C. Additional Theory

In the following, we show that the optimal value for the AFA MDP described in Section 3.1 is lower bounded (approximated) by the maximizer of the (negative of the) ACO's objective. Thus, feature subset acquisition guided by the ACO relates directly to the MDP of interest. We outline a sketch of the proof.

Theorem C.1. *(Informal.) The optimal value of the AFA MDP is lower bounded by the value of the ACO policy.*

Proof. In sticking with the usual conventions in reinforcement learning, we will re-express the relevant quantities in terms of maximization as opposed to minimization. Therefore, let us denote $\text{ACO}^k(x_o)$ as:

$$\text{ACO}^k(x_o) \coloneqq \max_{v \subset \{1,\ldots,d\} \setminus o \text{ s.t.} |v| \leq k} - \mathbb{E}_{y, x_v | x_o}[\ell(\hat{y}(x_{o \cup v}), y)] - \alpha |v|,$$

i.e. the negative of the objective that the ACO minimizes eq. (2) subject to the constraint that the number of additional features in less than or equal to k. Furthermore, we define

$$V^0(x_o) \coloneqq -\mathbb{E}_{y|x_o}[\ell(\hat{y}(x_o), y)]$$

and

$$V^k(x_o) \coloneqq \max\{V^0(x_o), \max_{j \in \{1,\ldots,d\} \setminus o} - \alpha + \mathbb{E}_{x_j | x_o}[V^{(k-1)}(x_{o \cup j})]\}.$$

We note that $V^0(x_o)$ is the expected negative loss (from prediction) given currently observed x_o and it is thus the value of both the optimal AFA and ACO policies of the 'termination action,' which prompts a prediction. Thus, we observed that $V^k(x_o)$ is the value following the optimal policy of the (k-acquisition) AFA MDP problem; either terminate, or choose the best additional feature according the the value with a reduced budget. Noting the equivalence of the optimal AFA MDP and ACO policies when there are 0 remaining features remaining (the base case, $V^0(x_o) = \text{ACO}^0(x_o)$), we proceed by induction on k, noting that[1]:

$$\begin{aligned}
&V^k(x_0) \\
&= \max\{V^0(x_o), \max_{j \in \{1,\ldots,d\} \setminus o} - \alpha + \mathbb{E}_{x_j | x_o}[V^{(k-1)}(x_{o \cup j})]\} \\
&\geq \max\{V^0(x_o), \max_{j \in \{1,\ldots,d\} \setminus o} - \alpha + \mathbb{E}_{x_j | x_o}[\text{ACO}^{(k-1)}(x_{o \cup j})]\} \\
&= \max\left\{ V^0(x_o), \max_{j \in \{1,\ldots,d\} \setminus o} - \alpha + \mathbb{E}_{x_j | x_o}\left[\max_{\substack{v \subset \{1,\ldots,d\} \setminus o \cup j \\ \text{s.t. } |v| \leq k-1}} - \mathbb{E}_{y, x_v | x_o, x_j}[\ell(\hat{y}(x_{o \cup j \cup v}), y)] - \alpha |v| \right] \right\} \\
&\overset{*}{\geq} \max\{V^0(x_o), \max_{j \in \{1,\ldots,d\} \setminus o} \max_{\substack{v \subset \{1,\ldots,d\} \setminus o \cup j \\ \text{s.t. } |v| \leq k-1}} - \mathbb{E}_{y, x_v, x_j | x_o}[\ell(\hat{y}(x_{o \cup j \cup v}), y)] - \alpha(1 + |v|)\} \\
&= \max\{V^0(x_o), \max_{\substack{v \subset \{1,\ldots,d\} \setminus o \\ \text{s.t. } 1 \leq |v| \leq k-1}} - \mathbb{E}_{y, x_v | x_o}[\ell(\hat{y}(x_{o \cup v}), y)] - \alpha |v|\} \\
&= \text{ACO}^k(x_o),
\end{aligned}$$

□

where $*$ follows from

$$\begin{aligned}
\forall v' \in \Omega, x_j, \ \max_{v \in \Omega} \mathcal{L}(x_j, v) \geq \mathcal{L}(x_j, v') \implies \\
\forall v' \in \Omega, \ \mathbb{E}_{x_j | x_o}\left[\max_{v \in \Omega} \mathcal{L}(x_j, v)\right] \geq \mathbb{E}_{x_j | x_o}\left[\mathcal{L}(x_j, v')\right] \implies \\
\mathbb{E}_{x_j | x_o}\left[\max_{v \in \Omega} \mathcal{L}(x_j, v)\right] \geq \max_{v \in \Omega} \mathbb{E}_{x_j | x_o}\left[\mathcal{L}(x_j, v)\right]
\end{aligned}$$

[1] We denote the singleton $\{j\}$ as j for clarity.

for $\Omega := \{v \subset \{1, ..., d\} \backslash o \cup j \text{ s.t. } |v| \leq k-1\}$, and $\mathcal{L}(x_j, v) := -\mathbb{E}_{y, x_v | x_o, x_j}[\ell(\hat{y}(x_{o \cup j \cup v}), y)] - \alpha|v|$. Note that one may recover the non-cardinality constrained ACO problem by considering large enough k (e.g., $\text{ACO}^d(\varnothing)$ for an empty initialization to acquisition).

D. Causal Inference and Decision Making

For a more comprehensive introduction to causal inference for optimal decision making, we refer interested readers to Kosorok & Laber (2019). In the following, we provide a concise summary of some relevant details. Let $y(a)$ denote the *potential outcome* under decision (intervention) $a \in \mathcal{A}$, which is the counterfactual outcome that would have been observed under action a (for a particular unit). Because the outcome under at most one action can be observed for a given unit, only one of the potential outcomes is ever observable. More formally, the observed outcome y is related to the potential outcomes through $y = \sum_{a' \in \mathcal{A}} y(a')\mathbb{I}(a = a')$, where a is the observed action for the unit.

A decision policy $\pi_{\mathcal{A}}(x)$ is defined as a mapping from the features x to an action $a \in \mathcal{A}$. We note that the policy, by definition, depends on which set of features x are used to determine the action. For a class of decision policies Π, an optimal decision policy $\pi^*_{\mathcal{A}}(x)$ is any policy maximizing $\mathbb{E}[y(\pi_{\mathcal{A}})]$, where $y(\pi_{\mathcal{A}})$ is the potential outcome under the action recommended by $\pi_{\mathcal{A}}$ (i.e., $y(\pi_{\mathcal{A}}) = \sum_{a' \in \mathcal{A}} y(a')\mathbb{I}(\pi_{\mathcal{A}}(x) = a')$). Note that this is a counterfactual (causal) parameter, since for some instances in the training data set, $y(\pi_{\mathcal{A}})$ might be unobserved since the observed action might differ from that recommended by $\pi_{\mathcal{A}}(x)$. In what follows, we will outline sufficient causal assumptions that allow counterfactual quantities such as $\mathbb{E}[y(\pi_{\mathcal{A}})]$ to be expressed and estimated in terms of the observed data.

We make the assumption that the complete features x are sufficient to adjust for confounding. That is, we assume that either **(1)** interventions a are marginally independent of $y(a)$ or that **(2)** $y(a) \perp a|x$. **(1)** is satisfied when a is marginally randomized such as in randomized controlled trials or A/B tests while **(2)** occurs when a depends on a behavior policy $\pi_b(x)$. Furthermore, we assume $p(a|x)$, the conditional distribution of interventions in the collected data, is greater than 0 for all $a \in \mathcal{A}$ and x for which $p(x) > 0$. This assumption essentially ensures that, in theory, all the potential outcomes can be observed.

Under scenario **(1)**, we have that $\mathbb{E}[y(a)|x_o, a] = \mathbb{E}[y|x_o, a]$. That is, the potential outcome under action a is, in (conditional) expectation, equal to the outcomes of those who were assigned action a in the observed data. Then, for a given context x_o, the optimal policy (when Π is unrestricted), can be identified as

$$\pi^*_{\mathcal{A}}(x_o) = \underset{a \in \mathcal{A}}{\text{argmax}}\ Q(x_o, a),$$

where $Q(x_o, a) := E[y|a, x_o]$. The identification of the optimal policy under scenario **(2)** is complicated because, in general, it is not true that $\mathbb{E}[y(a)|x_o, a] = \mathbb{E}[y|x_o, a]$. Instead, we have that $\mathbb{E}[Y(a)|x_o, a] = \mathbb{E}[\mathbb{E}[y|x, a]|x_o]$, which depends on the (unknown) distribution of $p(x|x_o)$. However, by contrasting $\mathbb{E}[y(\pi^*_{\mathcal{A}})]$ with $\mathbb{E}[y(\pi_{\mathcal{A}})]$ (i.e. examining the regret from using $\pi_{\mathcal{A}}(x_o)$ instead of $\pi^*_{\mathcal{A}}(x)$), we find that

$$\pi^*_{\mathcal{A}}(x_o) = \underset{\pi_{\mathcal{A}} \in \Pi}{\text{argmin}}\ \mathbb{E}[Q(x, \pi^*_{\mathcal{A}}(x)) - Q(x, \pi_{\mathcal{A}}(x_o))],$$

Therefore, a partial-information policy $\pi_{\mathcal{A}}(x_o)$ can be estimated by minimizing the empirical version of this loss. Then, with this learned policy $\hat{\pi}_{\mathcal{A}}(x_o)$ that uses partially observed information, we can proceed with the approximate AACO as outlined in the Methods section.

E. Additional Experiment Details

For the Cube and MNIST experiments, searching over all possible subsets $v \subseteq \{1, \ldots, d\} \setminus o$ to find the best (additional) subset of features is infeasible. As discussed in the Methods section, we approximate this minimization by choosing random subsets of features $\mathcal{O} \subseteq \{v | v \subseteq \{1, \ldots, d\} \setminus o\}$. In our experiments, we chose $|\mathcal{O}| = 10,000$.

The AACO policy approximates the distribution of $p(y, x_u|x_o)$ nonparametrically by using the set of k nearest neighbors. When getting nearest neighbors to perform the search over, we found that results where relative stable to choice and as few as $k = 5$ neighbors performed well; this is the number of neighbors used in the experiments (with $k = 20$ performing slightly better in the synthetic Cube dataset, see ablation below). In general, increasing the number of neighbors can lead to neighbors having values x_o less similar to the test instance's $x_o^{(i)}$, but having larger numbers of neighbors could lead to a better approximation of the conditional expectation of the loss, representing a bias-variance trade off. To standardize the relative importance of each feature, all features were mean-centered and scaled to have a variance of 1.

We found that gradient boosted trees tended to perform well and were therefore used for the AACO policy predictor $\hat{y}$. The training procedure for $\hat{y}$ varied upon the dimensionality of the features. Small to moderate dimension settings allowed for the separate training of a $\hat{y}$ for each possible subset of features, leading to a dictionary of predictors. In higher dimensions, we utilized a masking strategy where the feature vector, whose unobserved entries were imputed with a fixed value, was concatenated with a binary mask that indicated the indices of observed entries (Li et al., 2020). During training, these binary masks were drawn at random to simulate making predictions with missing features. While this predictor allows for predictions with arbitrary subsets of features and could be used to make the final predictions, we found better performance by using the aforementioned dictionary-of-predictors approach on the subsets of features available at prediction, which often saw relatively few unique subsets and could use caching to avoid retraining models. AACO models were ran on individual Titan Xp GPUs.

The AACO acquisition policy is a valid nonparametric policy in that, for a new instance drawn at test time, it is deployable and can actively acquire features and make a prediction without ever using unacquired features or the instance's label. In our experiments, we also considered using behavioral cloning (Bain & Sammut, 1995) to train a parametric policy $\pi_\theta(x_o, o)$ that imitates the AACO policy. After rolling the AACO policy out on the validation dataset, we trained gradient boosted classification trees to mimic the actions in this data. Then, these trained classification-based policies were rolled out on a test data set. As with training the arbitrary classification models, we utilized a masking strategy where the feature vector, whose unobserved entries were imputed with a fixed value, was concatenated with a binary mask that indicated the indices of observed entries (Li et al., 2020).

The Open Bandit Dataset (Saito et al., 2020) is a real-world logged bandit dataset provided by ZOZO, Inc., a Japanese fashion e-commerce company. The data contains information collected from experiments where users are recommended one of 34 fashion items, with the response variable being whether or not the user clicked on the recommended item. The Open Bandit Dataset contains several campaigns under different recommendation policies. We analyzed the AACO policy under the Men's campaign with the Thompson Sampling policy. To simplify the presentation, we filtered the data to only include events where the two most frequently recommended products were recommended. This leads to a setting where the objective is to learn a binary decision policy that chooses between these two clothing recommendations. Altogether, the filtered data set has over 1 million instances and 65 features (i.e. the context x).

Code for the AACO policy can be found at `https://github.com/lupalab/aaco`.

F. Synthetic Decision-Making Environment

In the synthetic decision-making environment, we create 4 features (x_0, x_1, x_2, x_3), where $x_0 \sim U(0, 1)$, x_1 follows a Rademacher(0.5) distribution, and (x_3, x_4) are jointly normal with a correlation of 0.3. Furthermore, to imitate the realistic scenario in which interventions a in a previously collected data set are not randomized, we draw a a a Bernoulli random variable according to a probit model with a linear dependence on the four features. To create a setup where different numbers of features are relevant to decision making, we draw the outcome y from a normal distribution with a mean equal to the following:

$$\begin{aligned} a * \big[& \mathrm{I}(0 < x_0 \leq 0.25) + I(0.25 < x_0 \leq 0.5)x_1 \\ & + \mathrm{I}(0.5 < x_0 < 0.75)x_1x_2 + \mathrm{I}(0.75 < x_0 < 1)x_1(x_3^2 - 1)\big] \end{aligned}$$

Clearly, the optimal decision when all of x_1, x_2, x_3, and x_4 are observed is to assign $a = 1$ when the above quantity in brackets is positive and assign $a = 0$ otherwise. The optimal decision when only a proper subset of the four features are observed requires marginalizing the bracketed quantity over the unknown features. Furthermore, the number of features relevant for decision making varies from instance to instance. For example, if $0 < x_0 \leq 0.25$, then only x_0 is needed to make an optimal decision. The feature x_0 is particularly important, since it is informative about which other features should be acquired.gins, page numbering, etc.) should be kept the same as the main body.

G. Time Complexity and Runtimes

Time complexity (and real-world runtimes) :

For the AACO with a random subset search of size $|\mathcal{O}|$, number of features d, number of training points n, and an average number of acquisitions m, the time complexity is $O(nd|\mathcal{O}|m)$ for a prediction, where a naive nearest neighbors search is $O(nd)$. For a pre-trained RL-based method (such as (Li & Oliva, 2021)), the time complexity at inference is $O(m)$ assuming the pre-learned policy makes decisions at a constant time. Therefore, the time complexity at inference is greater for the AACO. However, note that this is the typical tradeoff between nonparametric and parametric methods. Alternative Deep Learning based methods require complex computation during training, which is not necessary for the AACO. Additionally, we provide practitioners with options to choose between which of these tradeoffs is most applicable to their use-case through parametric policies (Section 3.5), which use imitation learning to mimic the AACO policy, speeding up inference time at the possible slight expense of some accuracy (and also achieving $O(m)$ time complexity). While these parametric policies generally are quicker at inference (due to no need to search over subsets and find neighbors), the AACO is an embarrassingly parallel policy. Furthermore, even without parallelization, we find that the AACO can efficiently perform inference. Please see Table 2 for experimental runtimes.

Table 2. Runtimes from experimental results. Time is minutes per 1k instances at inference.

Dataset	Time	Avg. # of Features
Gas	29	3.1
Grid	37	5.5
MNIST	80	15.4