Deletion-Anticipative Data Selection with a Limited Budget

Rachael Hwee Ling Sim [1] **Jue Fan** [1] **Xiao Tian** [1] **Patrick Jaillet** [2] **Bryan Kian Hsiang Low** [1]

Abstract

Learners with a limited budget can use supervised data subset selection and active learning techniques to select a smaller training set and reduce the cost of acquiring data and training *machine learning* (ML) models. However, the resulting high model performance, measured by a data utility function, may not be preserved when some data owners, enabled by the GDPR's right to erasure, request their data to be deleted from the ML model. This raises an important question for learners who are temporarily unable or unwilling to acquire data again: *During the initial data acquisition of a training set of size k, can we proactively maximize the data utility after future unknown deletions?* We propose that the learner anticipates/estimates the probability that (i) each data owner in the feasible set will independently delete its data or (ii) a number of deletions occur out of k, and justify our proposal with concrete real-world use cases. Then, instead of directly maximizing the data utility function, the learner can maximize the expected or risk-averse post-deletion utility based on the anticipated probabilities. We further propose how to construct these *deletion-anticipative data selection* (DADS) maximization objectives to preserve monotone submodularity and near-optimality of greedy solutions, how to optimize the objectives and empirically evaluate DADS' performance on real-world datasets.

1. Introduction

Training *machine learning* (ML) models with high predictive performance often requires large datasets. For example, online services such as Netflix and Amazon require different users' usage and purchase history to predict good recommendations, while healthcare startups require various patients' health records (e.g., heart statistics, CT scans) to predict diseases well. However, using a larger dataset incurs longer training and prediction time and more expensive labelling costs. Thus, conventionally, *supervised data subset selection* and *active learning* have often been used to select a smaller training set to reduce the time and cost incurred without significantly sacrificing accuracy (Wei et al., 2015); the former assumes accessibility of data labels but the latter does not. Concretely, both classes of *data selection* (DS) methods select a subset K (of size at most k) from a large feasible set M as the training dataset to maximize some chosen *data utility function* u, i.e., $K := \arg\max_{B \subseteq M : |B| \leq k} u(B)$, that positively correlates with model accuracy but can be more efficiently computed without model training.

The recent interest in the rights of individuals to own and protect their data as their properties, evidenced by the *General Data Protection Regulation* (GDPR), has created a stronger impetus for DS and a new important challenge at the same time: Firstly, as data owners require monetary compensations for their data (Ghorbani & Zou, 2019; Jia et al., 2019) and would only grant ML model *learners* temporary access to their data for DS, learners with a **limited monetary budget** (e.g., a healthcare startup) have a stronger impetus to acquire high utility data from *fewer* data owners. Secondly, as some selected data owners may request timely deletions of their data (e.g., sensitive health records) under GDPR's "right to erasure" in the future, the high data utility and model accuracy achieved via DS is not preserved after their data are deleted from the model (as empirically shown in Sec. 5). The learner may neither have the budget[1] nor logistics to acquire replacement data from unselected data owners for some time period. For example, a startup may find it tedious to attract and request consent/data access from data owners after every data deletion and prefer doing data acquisition annually instead. Thus, a novel gap and challenge arise: *During data acquisition with a limited*

[1]Department of Computer Science, National University of Singapore, Republic of Singapore [2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA. Correspondence to: Bryan Kian Hsiang Low <lowkh@comp.nus.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]A learner that reserves part of its budget to purchase replacement data later undesirably sacrifices better *current* data utility. We further describe scenarios when DADS is more useful in App. A.

*budget (Fig. 1), how can a learner **proactively** maximize the post-deletion data utility as in Fig. 2?*

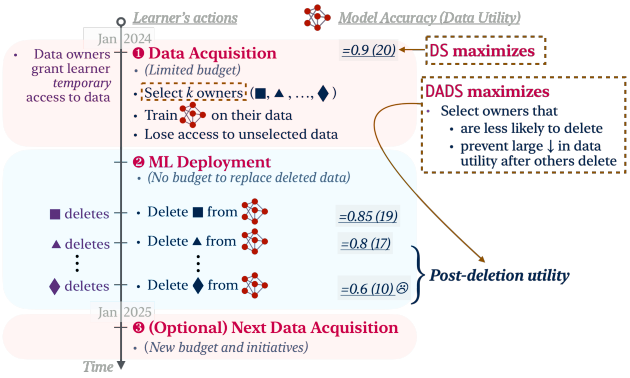


Figure 1: An overview of data selection with a limited budget. Using conventional DS can lead to a significant fall in the model accuracy/data utility after multiple future deletions from the ML model which may not be corrected until the next data acquisition cycle (when new budget and logistics become available). We propose deletion-anticipative data selection (DADS) to address the challenge.

Our **key perspective** is that although the learner does not know future deletions beforehand, the learner should proactively *anticipate* the probability of subset of owners staying present (i.e., not delete their data) in the period of interest (before the next data acquisition) and account for these probabilities in the *deletion-anticipative data selection* (DADS) objectives. Realizing this perspective involves addressing three sub-challenges. Firstly, instead of requiring the learner to tediously anticipate the probabilities for an exponential number of subsets, we must **(D1) reduce the number of probability parameters to anticipate** (or estimate). Secondly, given the anticipated probabilities, we must design **(D2) the DADS objectives to capture the post-deletion data utility**. Lastly, the learner desires guarantees that **(D3) the DADS objectives can be maximized greedily and near optimally**.

To address (D1), we propose that the learner assumes that each owner in the feasible set M will decide **(i) independently** or **(ii) dependently but similarly** to stay present. For (i), the learner estimates the probability each data owner stays based on surveys/histories of data owners ($\mathcal{O}(|M|)$ parameters); for (ii), the learner decides the probability of only a owners staying present out of k for each size a based on its preferences ($\mathcal{O}(k)$ parameters). We justify our proposal with concrete use cases in Sec. 3.

To address (D2), we design two DADS objectives (Sec. 3) for *risk-neutral* and *risk-averse* learners who, respectively, care about the average and worst-cases post-deletion data utility: Optimizing the **expected objective** EDADS maximizes the *expected* data utility (w.r.t. the probability owners stay present). In contrast, optimizing the **risk-averse objective** RA_α−DADS maximizes the α-level *conditional value-at-risk* data utility — the expected data utility under the worst $100\alpha\%$ cases. A lower $\alpha \in (0, 1]$ signifies greater concern for rare/worst utilities (associated with more deletions).

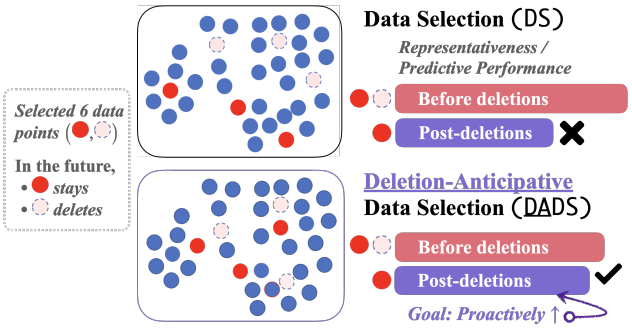


Figure 2: DS and DADS maximize the data utility and post-deletion data utility respectively. DADS selects a different set of data points that are closer to the centre of the data distribution. As a result, DADS is likely to have better post-deletion predictive performance and its staying red selected points are more representative of the initial full dataset.

To achieve (D3), we focus on DS objectives whose data utility function u is monotone submodular. In Sec. 4, we seek to preserve monotonicity and submodularity in the expected objective EDADS so as to exploit existing greedy submodular maximization algorithms (Nemhauser et al., 1978). For (i), we prove that the independence assumption is sufficient. For (ii), we prescribe how to set the probabilities of a data owners staying out of each size $t < k$. Lastly, we discuss techniques to reduce the extra computational cost of evaluating the DADS over the DS objective in Sec. 4.4 and empirically show that DADS lead to higher post-deletion predictive performance on real-world datasets in Sec. 5.

2. Background and Related Work

2.1. Background

Consider a set function u that maps any subset $A \subseteq M$ to its non-negative value, and denote the *marginal gain* of an element (in our context, datum) $j \in M$ to the set $A \subseteq M$ as $\Delta_u(j|A) \coloneqq u(A \cup j) - u(A)$.[2] Set function u is *monotone* if for every $j \in M$ and set $A \subseteq M$, $\Delta_u(j|A) \geq 0$ (i.e., adding j does not decrease utility). Set function u is *submodular* over M if for any sets A, B where $A \subseteq B \subseteq M$ and element $j \in M \setminus B$, $\Delta_u(j|A) \geq \Delta_u(j|B)$ (i.e., adding j yields a smaller gain on a larger subset). The maximization of submodular functions under cardinality constraints has been studied extensively. In particular, the greedy algorithm

[2] To ease notations, we use j in place of $\{j\}$ in set operations.

(Nemhauser et al., 1978) that iteratively selects the element with the largest marginal gain returns a solution that is at least $(1 - 1/e)$ of the optimal solution when the function is non-negative and monotone. App. C.1 details the greedy algorithm and operations preserving submodularity.

In DS, the learner maximizes some data utility function u that maps each data subset A to the utility of a model only trained on A. Existing supervised data subset selection and active learning works use heuristics or theoretical analysis to choose an efficiently computable u that positively correlates with model performance. App.C.2 details the definition of some submodular utility functions, including the *nearest neighbor* (NN) and *naïve Bayes* (NB) submodular functions (Wei et al., 2015), which consider the log-likelihood on the feasible set for their respective classifiers, and other functions in App. C.2. Maximizing the NN or NB submodular functions will, respectively, select a subset *representative* of the training set or a subset that has *diverse* feature coverage. It is meaningful to examine if these existing submodular DS objectives can be extended to submodular DADS objectives to exploit existing submodular maximization algorithms (Minoux, 1978; Mirzasoleiman et al., 2015; Nemhauser et al., 1978).

2.2. Related Work

Next, we will explain how our work is related to and differs from others that have considered data deletions, the "right to erasure", and stochasticity in the submodular function.

Machine unlearning (Bourtoule et al., 2021; Cao & Yang, 2015) is concerned with the *hows* of erasing data and its challenges such as efficiency, handling stochasticity, and incrementality of training. We view machine unlearning as complementary to our work. Our work selects the training data while anticipating and accounting for deletions during DS. Subsequently, when deletions are requested, the learner is free to use any unlearning method; in our experiments, we assume retraining from scratch.

Deletion-robust submodular maximization considers the deletion of some set D with maximum size d. Krause et al. (2008) have proposed how to achieve the (optimal) robust objective value after adversarial deletions of d elements (i.e., $\max_{B\subseteq M:|B|\leq k} \min_{D:|D|\leq d} u(B \setminus D)$) by selecting the set B to be logarithmically larger than k. Orlin et al. (2018) and Bogunovic et al. (2017) have proposed algorithms that achieve 0.387 of the optimal robust objective value when $d = o(\sqrt{k})$ and $o(k)$, respectively. Mirzasoleiman et al. (2017) have considered a streaming set where data can be dynamically added and deleted but required the deleted set D to be known. Separately, Kazemi et al. (2018) have proposed how to select a coreset M' (of the feasible set) that ensures a near-optimal solution to $\max_{B\subseteq M\setminus D:|B|\leq k} u(B)$ can still be found in M' after the deletion of some set D (unknown *a priori*). Dütting et al. (2022) and Mitrovic et al. (2017) have considered the streaming setting instead. Our work differs from the above as (a) we consider D to be unknown (see Sec. 3), (b) select a dataset of exactly size k to fit within a budget, and (c) maximize the expected and conditional value-at-risk instead of the robust adversarial case to preserve higher data utility in the average cases; see Sec. 5 for negative examples on how optimizing for worse cases (e.g., lower staying probability) results in lower data utility when there are few or no deletions.

Stochastic submodular maximization. Asadpour et al. (2008) have considered a setting where each element in the feasible set is an independent random variable (e.g., may be removed with some probability). They have proven that the *expected* set function is monotone submodular and can thus be maximized near-optimally by the greedy algorithm (Nemhauser et al., 1978) or the continuous greedy algorithm followed by rounding to an integral solution (Vondrák, 2008). While their non-adaptive results correspond to ours for the expected EDADS objective with independent decisions (Sec. 4.1), our work additionally (a) uses an alternative proof to show monotone submodularity for some *dependent* variables (Sec. 4.2), (b) prescribes how model learners can intuitively set the probability distribution and compute the DADS objectives efficiently (Secs. 3 & 4.4), (c) discusses the risk-averse RA_α-DADS objective, and (d) demonstrates DS and DADS applications empirically.

3. Problem Formulation

Let u be a data utility function. Consider any subset B of the feasible set M. Let p_B denote a *probability mass function* (pmf) mapping each subset $A \subseteq B$ to the probability that *only* the data owners in A stay present in the period of interest. So, the support of p_B is the power set of B. The learner's goal is to maximize the EDADS objective defined as the expected data utility if owners stay present or delete according to pmf's (plural of pmf) $\{p_B\}_{B\subseteq M}$:

$$u_{\mathbb{E}}(B) \coloneqq \mathbb{E}_{A\sim p_B(\cdot)}\left[u(A)\right] \tag{1}$$

under some constraints. Our work considers a cardinality constraint, i.e., at most k elements in the selected set $K \subseteq M$ and exactly k elements for monotone submodular set functions. Deletions within K are not known during DS and may happen at any time in the period of interest *after* DS.

Next, we address the challenge (D1) and answer the question: *How can a learner (e.g., healthcare startup) anticipate deletions and set the pmf's $\{p_B\}_{B\subseteq M}$ readily in real-world applications?* Naively, the learner has to anticipate the probabilities for an exponential number of subsets A, B s.t. $A \subseteq B \subseteq M$. To reduce the number of probability parameters to estimate, we propose simplified yet realistic settings. The learner can follow the flowchart

in Fig. 7 (App. B) and consider that data owners decide to stay present (i) *independently* (Sec. 3.1) or (ii) dependently but the probability only depends on the *number of owners* (Sec. 3.2). App. B.4 gives a summary of notations.

3.1. Independent Decisions

The learner can assume that each data owner $j \in M$'s decision to stay present (i.e., not delete its data) is an independent Bernoulli variable with staying probability s_j.[3] To estimate s_j of data owner j, the learner can survey owner j indirectly on its privacy preferences or directly on its staying probability, observe its data deletion history in data sharing platforms, or enforce $s_j = 1$ through binding contracts (e.g., each newly selected data owner cannot delete its data in a fixed period, which is similar to how a new tenant must commit to leasing the property for a short fixed period). It is also natural that owners of data with different class labels may exhibit different staying probability s_j. For example, patients with health conditions may be more likely to request deletions of their sensitive medical information than patients without and have a smaller s_j.

The probability $p_B(A)$ is thus the product of the probability that only owners in set A stay while owners in set $B \setminus A$ delete, i.e., $p_B(A) = \prod_{j \in A} s_j \prod_{\ell \in B \setminus A} (1 - s_\ell)$.

3.2. Dependent Decisions

Next, we allow each data owner's decision to stay present to *depend* on others' decisions but reduce the number of probability parameters by assuming that each data owner behaves *similarly* and is *equally* likely to delete their data. Thus, the probability $p_B(A)$ of only subset A in B staying present only depends on the size of A and B. Here, we propose how the learner should use its experience, data acquisition plans or limited surveys to decide the probabilities (or the relative weights) $(r_a)_{a=0}^{k}$ of only a owners staying present (out of k selected owners) to non-negative values that sum to 1 as follows.

(I) Decide tolerance to different number of deletions.

- An uninformed learner may place equal weights on any number of deletions and set $r_a = 1/(k+1)$ for all $a = 0, 1, \ldots, k$.
- A startup that only experienced $\leq z$ deletions in the last year or plans to reacquire data again after z (e.g., $= .1k$) deletions can set $r_a = 0$ for $a < k - z$. For example, the startup can model the number of data owners staying follows the (discrete) uniform distribution $\texttt{U}(k - z, k)$.
- A pessimistic learner is pessimistic, who wishes to tolerate more than z deletions, can set $r_a = 0$ for $a \geq k - z$. For example, the learner can model the number of data owners staying follows the uniform distribution $\texttt{U}(0, k - z - 1)$.

(II) Estimate common staying probability less confidently. When the learner cannot confidently decide on a *value* for the staying probability s of every owner/class in Sec. 3.1, we propose that the learner can model s as a *Beta* distribution to reflect its uncertainty. Thus, the number of data owners staying follows a *beta-binomial* distribution with k trials. The probability r_a that only a selected owners stay is $\texttt{BetaBin}(a|k, \alpha, \beta)$ and the expected staying probability $\mathbb{E}[s] = \alpha/(\alpha + \beta)$. Suppose that the learner estimates the staying probability to be 0.5.

- An unconfident learner can set the probability r_a as $\texttt{BetaBin}(a|k, 1, 1)$ which follows the uniform distribution.
- A moderately confident learner who has surveyed a small group of data owners can set $r_a = \texttt{BetaBin}(a|k, 10, 10)$.
- More generally, $\alpha - 1, \beta - 1$ can encode the change in belief from surveying a small group of owners.

For any subset B_k of the maximum size k, the probability $p_{B_k}(A)$ of only subset A of size a (in B_k) staying present is then obtained by dividing r_a by the number of subsets of size a, i.e., $p_{B_k}(A) = r_a / \binom{k}{a}$. We prescribe how to set the pmf p_{B_t} for other subset B_t of size $t < k$ in Sec. 4.2.

3.3. Risk-Averse DADS

The *expected* data utility (Eq. 1) does not inform a learner about the worst-case data utility after many data owners delete their data. Thus, a risk-averse learner, who cares about the worst cases, may prefer the *value-at-risk* (VaR) and *conditional value-at-risk* (CVaR) (Sarykalin et al., 2008) utility with parameter $\alpha \in (0, 1]$ that encodes concern for only the worst $100\alpha\%$ possible utility. Let A be a set random variable (r.v.) sampled according to $A \sim p_B(\cdot)$ and $u(A)$ be a *discrete* r.v. representing the corresponding data utility. The VaR is the α-percentile of $u(A)$, i.e., $u_{\text{VaR}_\alpha}(B) \coloneqq \inf\{\tau \in \mathbb{R} : \mathbb{P}[u(A) \geq \tau] \leq \alpha\}$. The risk-averse RA_α-DADS objective is the expected value of $u(A)$ when $u(A)$ is restricted to at most its α-percentile:

$$u_{\text{CVaR}_\alpha}(B) \coloneqq (1 - \lambda)\, u_{\text{VaR}_\alpha}(B) + \lambda\, \mathbb{E}_{A \sim p_B(\cdot)}[u(A) | u(A) < u_{\text{VaR}_\alpha}(B)] \quad (2)$$

where $\lambda \coloneqq (1/\alpha)\ \mathbb{P}[u(A) < u_{\text{VaR}_\alpha}(B)]$ "splits" the pmf at $u_{\text{VaR}_\alpha}(B)$ to obtain the α-percentile. Rockafellar & Uryasev (2002) have re-expressed the risk-averse RA_α-DADS objective (2) as

$$\max_{\tau \geq 0} H(B, \tau) \coloneqq \tau - \frac{1}{\alpha} \mathbb{E}_{A \sim p_B(\cdot)}[\max(0, \tau - u(A))]\,. \quad (3)$$

Remark. When $\alpha = 1$ and the learner is risk-neutral, the EDADS objective (1) is recovered. We choose the

[3] Independence is a realistic assumption when the data owners are diverse or do not know one another.

$\texttt{RA}_\alpha\texttt{-DADS}$ objective over other objectives that combine the mean and standard deviation of $u(A)$ as it is a coherent risk measure (Sarykalin et al., 2008), affected by the distribution's asymmetry, and allows us to exploit greedy submodular maximization algorithms described in Sec. 4.3.

4. Optimizing DADS Objectives

Sec. 2 describes that the data utility set function u is often monotone submodular and amenable to efficient optimization via greedy algorithms. However, the EDADS objective $u_\mathbb{E}$ is not necessarily monotone submodular as it takes an expectation w.r.t. p_B that changes with the input set B, as illustrated by the counterexample in App. D.1. Here, we propose Prop. 1 specifying conditions on the pmf's to ensure the monotone submodularity of $u_\mathbb{E}$ and address the challenge (D3). In App. B, we describe our algorithms to optimize the DADS objectives (Algos. 1-2) and their time complexity.

Proposition 1. *Let u be (monotone) submodular over the feasible set M. Consider a fixed subset $K \subseteq M$ with a* valid *pmf p_K, i.e., the probabilities of all subsets of K are non-negative and sum to 1. For every subset $B \subseteq K$, let the pmf p_B be a marginalization over the pmf p_K, i.e., $p_B(A) = \sum_{T \subseteq K \setminus B} p_K(A \cup T)$ (‡). Then,*

$$u_\mathbb{E}(B) \coloneqq \mathbb{E}_{A\sim p_B(\cdot)}\left[u(A)\right] = \mathbb{E}_{C\sim p_K(\cdot)}\left[u(B\cap C)\right] \quad (4)$$

is (monotone) submodular over the set K.

Its proof is in App. D.2 and holds for *any* valid pmf p_K, including those in Secs. 4.1 and 4.2. The main intuition is that we can rewrite the expression to take an expectation over the *same* distribution (regardless of the input set) by applying a change-of-(set)-variable and Lemma 1.

However, during earlier rounds of DADS, we would not know the selected set K to set p_B accordingly. To circumvent this issue, we (a) apply Prop. 2 (a modification of Prop. 1) which impose conditions on the pmf p_M instead in Sec. 4.1 and unique use cases in App. E or (b) enforce that the pmf only depends on the size k (instead of the specific set K) and use (monotone) submodularity for sets up to size k in Sec. 4.2.

Proposition 2. *Let u be (monotone) submodular over the feasible set M. If there exists a valid pmf p_M such that every pmf p_B can be expressed as a marginalization over the pmf p_M, i.e., $p_B(A) = \sum_{T \subseteq M \setminus B} p_M(A \cup T)$ (‡), then*

$$u_\mathbb{E}(B) \coloneqq \mathbb{E}_{A\sim p_B(\cdot)}\left[u(A)\right] = \mathbb{E}_{C\sim p_M(\cdot)}\left[u(B\cap C)\right] \quad (5)$$

is (monotone) submodular.

Its proof is similar to Prop. 1 and in App. D.3. Thereafter, the learner can maximize $u_\mathbb{E}$ by applying the greedy algorithm (Nemhauser et al., 1978) (see App. C.1): At round $t+1$, when the set K_t has been selected, the learner greedily selects the next element $j \in M \setminus K_t$ with the largest marginal gain $\Delta_{u_\mathbb{E}}(j|K_t) \coloneqq u_\mathbb{E}(K_t \cup j) - u_\mathbb{E}(K_t) = \mathbb{E}_{C\sim p_K(\cdot)}\left[\Delta_u(j \cap C | K_t \cap C)\right]$.

4.1. Independent Decisions

Let $\mathbb{I}_j$ be the indicator variable that indicates if owner j stays. Since the expected EDADS objective can be rewritten as an expectation over the joint pmf of $\{\mathbb{I}_j\}_{j\in M}$ (5), i.e., $p_M(A) = \prod_{j\in A} s_j \prod_{\ell \in M\setminus A}(1-s_\ell)$ and

$$\begin{aligned} u_\mathbb{E}(B) &= \mathbb{E}_{\{\mathbb{I}_j\}_{j\in B}}\left[u\left(\{j|\mathbb{I}_j=1\}\right)\right] \\ &= \mathbb{E}_{\{\mathbb{I}_j\}_{j\in M}}\left[u\left(B\cap\{j|\mathbb{I}_j=1\}\right)\right], \end{aligned} \quad (6)$$

it is (monotone) submodular by Prop. 2. Intuitively, the realization of the indicator variables of unselected owners $\{\mathbb{I}_\ell\}_{\ell\in M\setminus B}$ can be ignored. However, their inclusion makes the RHS an expectation over the same distribution p_M.

Interpretation of marginal gain. In App. D.4, we simplify the marginal gain $\Delta_{u_\mathbb{E}}(j|K_t)$ of owner j to the product of its staying probability and expected marginal contribution to subsets of K_t:

$$\Delta_{u_\mathbb{E}}(j|K_t) = s_j \times \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[\Delta_u(j|A)\right]. \quad (7)$$

So, an owner j is preferred when it has a higher staying probability, or leads to a strictly higher marginal gain to some staying subset A and at least the same gain to other subsets. In our experiments in Sec. 5.1, the EDADS objective still selects owners with lower staying probability due to their larger expected marginal contribution (e.g., due to few owners with similar data anticipated to stay present).

4.2. Dependent Decisions

In Sec. 3.2, we suggest that the learner can assume each data owner is equally likely to delete their data. This assumption allows us to simplify the probability of only set A of size a (in B_k of size k) staying present from $p_{B_k}(A) = r_{|A|}/\binom{k}{|A|}$ to $p_k(a) = r_a/\binom{k}{a}$ for notational convenience. Now, we will outline how to decide the pmf's p_t for sizes $t < k$.

The learner should apply Prop. 1 to decide the pmf's $\{p_t\}_{t=1}^{k-1}$ of earlier rounds (with $< k$ selected), and obtain a (monotone) submodular set function for sets up to size k. For any a, the probability $p_t(a)$ of a subset of size a (out of t) staying present can be more conveniently computed in a recursive manner using Cor. 1 below, as proven in App. D.5:

Corollary 1. *(‡) is equivalent to "owner j's decision to stay/delete does not affect the probability of any other subset $A \subseteq B$ staying present". Formally,*

$$\forall A \subseteq B\ \forall j \in K\setminus B\ \left[p_B(A) = p_{B\cup j}(A) + p_{B\cup j}(A\cup j)\right].$$

The learner starts by computing the probability $p_{k-1}(a)$ of a subset of size $a < k$ (out of $(k-1)$ selected) staying present from the decided probability of a subset of size a or $a+1$ (out of k) staying present: $p_{k-1}(a) = p_k(a) + p_k(a+1) = r_a/\binom{k}{a} + r_{a+1}/\binom{k}{a+1}$. Next, the probability $p_{k-2}(a)$ can be computed as $p_{k-1}(a) + p_{k-1}(a+1)$, and so forth.

*Example from Sec. 3.2 (**I**).* Consider the startup that tolerates $\leq z$ deletions and sets $r_a = 0$ for $a < k - z$ and $r_a > 0$ otherwise. Then, the probability $p_{k-1}(a) = 0$ iff $a < k-1-z$. Recursively, the probability $p_t(a)$ is only positive for the z largest subsets out of t.

*Example from Sec. 3.2 (**II**).* Consider the learner who less confidently estimates the common staying probability and sets $r_a = \texttt{BetaBin}(a|k, \alpha, \beta)$. In App. D.6, we prove that setting $p_t(a) = \texttt{BetaBin}(a|t, \alpha, \beta)/\binom{t}{a}$ for each $t < k$ satisfies Cor. 1. Thus, $p_t(a)$ can be directly computed instead without knowing k in advance: the learner can more flexibly consider that the number of data owners staying in different groups (e.g., classes) follows different independent distributions without knowing the number selected per group in advance. For example, the learner can model that the positive class stays with probability s_+ where $s_+ \sim \texttt{Beta}(\alpha, \beta)$ and the negative class always stays, as empirically shown in Fig. 17.

In App. G, we highlight interesting connections between the marginal gain $\Delta_{u_{\mathbb{E}}}(j|K_t)$ in round $t+1$ and cooperative game theory (CGT) based data valuation (DV) (Chalkiadakis et al., 2011; Sim et al., 2022). For example, when the learner considers each owner equally likely to delete its data and any number of deletions equally likely (Sec.3.2 (**I**)), in each round, our EDADS objective will select the owner with the largest Shapley value (Shapley, 1953) which is an equitable data value (Ghorbani & Zou, 2019).

4.3. Risk-Averse DADS

Moreover, when the learner considers independent decisions (Sec. 4.1) or sets the dependent decisions' pmf's via Sec. 4.2, we can use Prop. 1 to express the $\texttt{RA}_\alpha\texttt{-DADS}$ objective (3) as an expectation over the same distribution. $\texttt{RA}_\alpha\texttt{-DADS}$ can then be approximated by taking n samples $C^i \sim p_K(\cdot)$:

$$\max_{\tau \geq 0} H(B,\tau) \coloneqq \tau - \frac{1}{\alpha}\mathbb{E}_{C\sim p_K(\cdot)}\left[\max(0, \tau - u(B\cap C))\right] \approx \tau - \frac{1}{\alpha n}\sum_{i=1}^{n}\max(0, \tau - u(B\cap C^i)) \,.$$

Zhou & Tokekar (2022) have proven that for any τ, $H(B,\tau)$ preserves the set function u's monotone increasing and submodularity (over B) properties, and for any set B, $H(B,\tau)$ is concave in τ. Consequently, Zhou & Tokekar (2022) have proposed using the greedy algorithm at regular intervals of τ and selecting (B,τ) with the best *sampled* objective value. The approximation quality is better when risk aversion is low (i.e., α is close to 1) and the curvature of the submodular function u is low.[4] The learner can improve the optimality guarantee by increasing n or decreasing the separation between the evaluated τ's. We propose that the learner can achieve the latter by finding τ that maximizes $H'(\tau) \coloneqq \max_B H(B,\tau)$ using numerical optimization techniques like Brent's method (Brent, 2013).

4.4. Efficient Evaluation of $u_{\mathbb{E}}$

Computing the set functions (4) and (6) in the DADS objectives exactly and naïvely requires exponential $\mathcal{O}(2^k)$ time and is therefore intractable for a larger maximum selected size k. To avoid this cost, we consider the following solutions.

Sample average approximation (SAA) is a general technique that would work for any data utility function u and pmf p_K satisfying Prop. 1. SAA replaces $u_{\mathbb{E}}(B) = \mathbb{E}_{C\sim p_K(\cdot)}\left[u(B\cap C)\right]$ with the mean computed from n i.i.d. samples of subsets $C^i \subseteq K$, i.e., $u_{\text{avg}}(B) \coloneqq n^{-1}\sum_{i=1}^{n} u(B\cap C^i)$. So, its computational cost is n times that of evaluating the DS objective. The probability the SAA solution is optimal for the original problem approaches 1 exponentially fast (Kleywegt et al., 2002): Theorem 3 of Yu & Ahmed (2016) suggests that an algorithm that solves the SAA problem (i.e., $\max_B u_{\text{avg}}(B)$) with an approximation ratio ρ (e.g., $1-1/e$) also achieves a $[\rho(1-\epsilon)-\epsilon]$-optimal solution on the original problem (i.e., $u_{\mathbb{E}}$) with probability at least $1-2\delta$. This approximation requires $2k\sigma^2\epsilon^{-2}\log(|M|/\delta)$ samples where ϵ bounds the scaled absolute difference $|u_{\text{avg}}(A) - u_{\mathbb{E}}(A)|/\max_B u_{\mathbb{E}}(B)$ across every set A of size $\geq k$ and σ^2 is a problem specific constant. Let the marginal gain of data owner $j \in M \setminus K_t$ to the selected set K_t be the independent r.v. $\delta_j^i \coloneqq u((K_t\cup j)\cap C^i) - u(K_t\cap C^i)$ bounded below by 0 (due to monotonicity) and above by the largest marginal gain $\overline{\Delta}_{t-1} = \max_{j\in M\setminus K_{t-1}}\Delta_u(j|K_{t-1})$ in round $t-1$ (due to submodularity). Then, we can apply Hoeffding's inequality (Hoeffding, 1963) to obtain

$$\mathbb{P}\left(\left|n^{-1}\textstyle\sum_{i=1}^{n}\delta_j^i - \Delta_{u_{\mathbb{E}}}(j|K_t)\right| \geq \varepsilon\overline{\Delta}_{t-1}\right) \leq 2\exp\left(-2n\varepsilon^2\right).$$

For example, with $n = 5000$ samples, the estimated marginal gain will deviate from the true expected gain by more than $\varepsilon = 2\%$ of $\overline{\Delta}_{t-1}$ with probability at most $2\exp\left(-2n\varepsilon^2\right) = 3.7\%$. So, SAA can distinguish between data owners with significantly different marginal gains.

How should the learner sample the staying subset C^i without knowing K? We propose that the learner can

[4] For submodular functions, increasing the selected size k may decrease the minimum marginal gain of each element j to any set A (of size $< k$), increase the curvature and worsen the approximation guarantee.

incrementally construct C^i as follows. At round $t+1$ when K_t has been selected, the learner should sample whether to add owner $j \in M \setminus K_t$ to C^i from the Bernoulli distribution with success probability s_j for the case of independent decisions (Sec. 4.1) and $p_{t+1}(|C^i|+1)/p_t(|C^i|) \in [0,1]$ (Cor. 1) for the case of dependent decisions (Sec. 4.2). Note that the learner does not resample whether owners in K_t stay. Thus, when deletion (i.e., no addition) is sampled, no update is needed. Otherwise, we can reuse DS's efficient techniques and compute the utility of the updated sampled staying subset C^i incrementally. For example, for the *diversity* data utility function (App. C.2), incrementally updating the Cholesky factor (Chen et al., 2018b) (with stored information from the Cholesky decomposition) reduces the time complexity per round from $\mathcal{O}(|M|^3)$ to $\mathcal{O}(t|M|)$.

Alternatively, when the utility function u (e.g., nearest neighbor submodular function) is a linear combination of **polytime-computable *multilinear* functions**, the learner can apply Cor. 2 (proven in App. D.7 and adapted from the study of multilinear extension of submodular functions) to efficiently compute the EDADS exactly:

Corollary 2 (Expectation of multilinear functions)**.** *Consider the vector $\mathbf{v} \coloneqq (v_j)_{j\in M}$ and the multilinear function $\mu : \mathbf{v} \in \mathbb{R}^{|M|} \mapsto \mathbb{R}_{\geq 0}$. A multilinear function is a polynomial in which each component v_j has a degree of 0 or 1 in every monomial, i.e., $\mu(\mathbf{v}) \coloneqq \sum_{\ell=1}^{L} \left(c_\ell \prod_{j\in\mathcal{J}_\ell} v_j\right)$ with weight c_ℓ and $\mathcal{J}_\ell \subseteq M$ for $\ell = 1,\ldots,L$.*

Independent decisions.*(Lemma 4.1 of of (Özcan et al., 2021)) Let $\mathbf{i} \coloneqq (i_j)_{j\in M} \in \{0,1\}^{|M|}$ be a random vector of independent Bernoulli variables parameterized by $\mathbf{s} \coloneqq (s_j)_{j\in M} \in [0,1]^{|M|}$. The j-th component i_j indicates if owner j stays and is 1 with probability s_j. Then, $\mathbb{E}_{\mathbf{i}\sim \mathtt{Bern}(\mathbf{s})}[\mu(\mathbf{i})] = \mu(\mathbf{s})$.*

Dependent decisions. *Let $\mathbf{i} \coloneqq (i_j)_{j\in M} \in \{0,1\}^{|M|}$ be a vector whose j-th component i_j indicates if owner j is selected and stays. For any set B_k of k selected owners, $\mathbf{i}$ can be decomposed into the random vector $\mathbf{i}_{B_k} \coloneqq (i_j)_{j\in B_k}$ of dependent variables and the zero vector corresponding to the unselected owners. Also, let $p_k(\sum_{j\in B_k} i_j)$ sums to 1 over all realizations of $\mathbf{i}_{B_k}$, and p_k be as defined in Sec. 4.2. Then, $\mathbb{E}_{\mathbf{i}_{B_k}\sim p_k(\cdot)}[\mu(\mathbf{i})] = \sum_{\ell=1}^{L} c_\ell\, \mathbb{I}[\mathcal{J}_\ell \subseteq B_k]\, p_{|\mathcal{J}_\ell|}(|\mathcal{J}_\ell|)$ where the probability $p_{|\mathcal{J}_\ell|}(|\mathcal{J}_\ell|)$ of $|\mathcal{J}_\ell|$ owners (out of $|\mathcal{J}_\ell|$) staying present satisfies Prop. 1 (e.g., computed recursively as described in Sec. 4.2).*

5. Experiments

In this section, we focus on the empirical performance of *supervised data subset selection* (which use data labels) and defer results on *active learning* data utility functions (e.g., variance reduction for Gaussian processes), which do not use data labels, to App. H.6. As our approach should improve the *post-deletion* utility of *any* non-anticipative DS objectives (which often correspond to some submodular data utility function to be maximized), we only consider the following **(submodular function-dataset)** combinations to compare the performance of our deletion-anticipative DADS objectives vs. the conventional DS objective:

(NN-S) *nearest neighbor* (NN) submodular function on a 2-class Synthetic dataset.

(NN-H) NN submodular function on the combined Heart disease dataset (Lapp, 2019). The selected set is used to train an NN classifier to predict if a patient has heart disease. We consider that a patient without the disease will never delete its data (i.e., staying probability $s_- = 1$) and vary the staying probability s_+ of every patient with the disease.[5] The learner measures the F1 score metric, which balances the precision and recall of identifying heart disease patients, on the validation set.

(NN-F) NN submodular function on the benchmark Fashion MNIST image dataset (Xiao et al., 2017) with $|M| = 60000$. The selected set is used to train an NN classifier and predict the clothing type. We vary the common staying probability s across all image owners in the feasible set. This experiment simulates learners using the NN objective to select other image data (e.g., CT scans) in practice. The learner measures the accuracy metric on the validation set.

(NB-A) *naïve Bayes* (NB) submodular function on the Adults income dataset (Becker & Kohavi, 1996) with $|M| = 30718$. The selected set is used to train a Categorical NB model to predict whether a person has an income of at least 50k. We vary the common staying probability s across all data owners in the feasible set. The learner measures the balanced accuracy metric on the validation set.

In these applications, each owner owns a single datum and may want to delete its data with sensitive attributes (e.g., blood pressure, education, income). We provide details on the submodular functions and experimental setup in App. C.2 and App. H.1, respectively.

5.1. Independent Decisions

2D visualization with (NN-S). In Fig. 3, we mark the $k = 10$ data points (red) selected by EDADS and DS objectives under varying common staying probability s. It can be observed that as s decreases, EDADS selects data closer to one another and the center of the class because the centers have higher net similarity to others and help preserve higher

[5] For simplicity, we fix the same staying probability across all data of the same class. However, in practice, DADS will work when the probabilities vary across owners and inherently prefer owners with higher probabilities of staying. In contrast, naïve heuristics (e.g., manually selecting more of a class) may not work well.

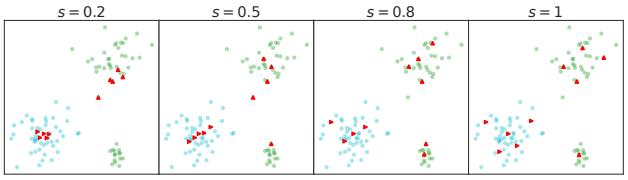
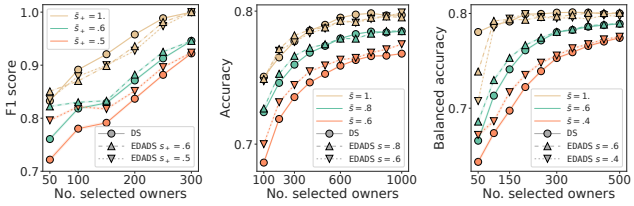


Figure 3: (NN-S) 10 data points (red) selected by EDADS with varying s where $s = 1$ corresponds to DS objective.



(a) (NN-H) (b) (NN-F) (c) (NB-A)

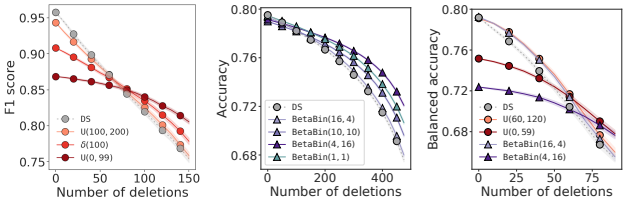
Figure 4: Graphs of mean validation set metric scores (with $2\times$ standard error (shaded) across 2500 simulations) with an increasing no. of data owners (horizontal axis) selected by DS and EDADS (with varying s) for various datasets (a-c) when owners stay with simulated staying probability $\bar{s}$.

NN objective value if other selected owners delete their data as anticipated. Moreover, when the staying probability s is too low (e.g., $= 0.2$), EDADS will forego the smaller green cluster to cover more of the larger green cluster. We have included other visualization experiments (e.g., when the probability differs across classes and owners and when owners make dependent decisions) in App. H.

Observations. In Fig. 4, we plot the validation set metrics with and without anticipated deletions. It can be observed that when owners stay according to our probability model (non-beige curves), our EDADS objective outperforms DS on various validation set metrics. This advantage (i.e., the gap between curves) is more evident when fewer owners are selected (left of graphs) or when the simulated staying probability $\bar{s}$ is lower (orange case). In App. H.3, we have shown that this better performance after deletions is due to selecting more redundant data in important regions likely to be deleted (e.g., positive class in (NN-H)). However, we often observe that such an advantage comes with a trade-off: Without deletions (beige curves), EDADS may achieve a poorer validation set metric score than DS. In App. H.3, we have also compared against the random selection baseline.

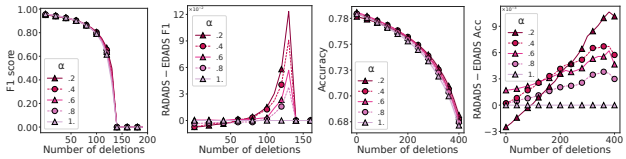
5.2. Dependent Decisions

Next, in Fig. 5, we plot the validation set metrics vs. the number of deletions when the learner considers, in Sec. 3.2, (I) tolerance to $\leq$ / $>$ z deletions or (II) uncertainty in the staying probability. The number of owners staying



(a) (NN-H), k=200 (b) (NN-F), k=600 (c) (NB-A), k=120

Figure 5: Graphs of mean validation set metric scores (with $2\times$ standard error (shaded) across 500 simulations) with an increasing no. of deletions (horizontal axis) for various datasets (a-c). k owners are selected by DS or EDADS. For EDADS, the no. of owners staying (out of k) follows various distributions given in the legend.



(a) (NN-H), k=240 (b) (NN-F), k=500

Figure 6: Graphs of mean validation set metrics obtained from the selections of k data owners by EDADS and RA_α−DADS (with varying α) (and additionally their differences) across 500 simulations with an increasing no. of deletions for various datasets (a-b).

follows (I) Uniform U or Dirac delta δ distribution and (II) BetaBin distribution, respectively. It can be observed that for different anticipated probabilities, the validation set metric falls at different rates as the number of deletions increases. DS has the best metric score with no deletions (leftmost of graphs) but has the steepest fall and worst metric with many deletions. In contrast, EDADS achieves a better metric score than DS under moderate and more deletions. Moreover, as the anticipated deletions increase (see darker curves corresponding to $\text{U}(0, \cdot)$ and lower $\alpha/(\alpha + \beta)$ in BetaBin), the metric under no deletions (leftmost) worsens but the performance falls at a slower rate. In App. H.4, we have verified that when owners stay present according to our probability model, EDADS preserves higher expected objective values and metric scores than DS.

5.3. Risk-Averse DADS

Now, we fix the anticipated staying probability distributions and vary α in the RA_α−DADS objectives. We use (NN-H) with $k = 240$ to consider the case of independent decisions with $s_+ = 0.5$ and $s_- = 1.0$, and (NN-F) with $k = 500$ to consider the case of dependent decisions with $p_k(a) = \text{BetaBin}(a|500, 4, 2)$. In Fig. 6, it can be verified that smaller α (darker curves) leads to better metric scores

in the worst cases with more deletions. However, with no deletions, the scores decrease. We observe that the decrease from lowering α is less drastic than the decrease from lowering the staying probability s. Thus, the learner can vary α in the $\texttt{RA}_\alpha\texttt{-DADS}$ objectives to improve the metric score in the worst cases without calibrating s and without markedly impacting the metric when there are few deletions. In App. H.5, we have further verified that $\texttt{RA}_\alpha\texttt{-DADS}$'s selection leads to a higher CVaR at level α of the objective value and metric than EDADS for the same staying probability distribution.

6. Conclusion

This paper describes how ML model learners should anticipate deletions in practice and maximize the expected or risk-averse DADS objective to preserve higher data utility after anticipated deletions. A limitation of the work is that the reliance on sampling for some utility functions and $\texttt{RA}_\alpha\texttt{-DADS}$ increases the computational cost and only returns an approximate solution. This limitation can be mitigated by taking the average/best of more parallel runs or future work that considers more advanced sampling methods (e.g., stratified sampling (Maleki et al., 2013; Wu et al., 2023)) that can still benefit from an efficient incremental update of the utility function. Future work can also explore the behavior of the deletion-anticipative version of other data selection algorithms (Wei et al., 2015; Killamsetty et al., 2021; Mirzasoleiman et al., 2020) (e.g., algorithms which select a new batch every few epochs) and improve DADS performance by accounting for class imbalance. App. I additionally discusses some questions a reader may have.

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-RP-2022-029).

Impact Statement

When learners use DADS in place of DS to reduce the time and cost incurred in labelling data and training an ML model, one potential ethical concern is that they may select a dataset with an imbalanced number of points from groups with different deletion probabilities (e.g., race, health status). This may create unfairness across groups and classes. To mitigate the imbalance and correct for any bias, the learner should consider works that address fairness across groups (Yan et al., 2020; Farrand et al., 2020).

We foresee more new concerns when data owners are surveyed on their staying probability and stand to receive monetary compensation for their selection. Data owners may commit to binding contracts, request deletions less frequently to maintain a better history or misreport their staying probability to increase their chance of selection. The former is undesirable as the data owners would have to temporarily forego their right to erasure. The latter is undesirable as the set selected by EDADS may be sub-optimal and have negative consequences on the unselected data owners and the ML model users. To mitigate the former, the learner or the data sharing platforms should educate the data owners. To mitigate the latter, the learner or the data sharing platforms can make use of historical behavior and a more thorough audit.

References

Asadpour, A., Nazerzadeh, H., and Saberi, A. Stochastic submodular maximization. In Papadimitriou, C. and Zhang, S. (eds.), *Internet and Network Economics. WINE 2008*, volume 5385 of *Lecture Notes in Computer Science*, pp. 477–489. Springer Berlin Heidelberg, 2008.

Becker, B. and Kohavi, R. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Bogunovic, I., Mitrović, S., Scarlett, J., and Cevher, V. Robust submodular maximization: A non-uniform partitioning approach. In *Proc. ICML*, pp. 508–516, 2017.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning. In *Proc. IEEE S&P*, pp. 141–159, 2021.

Brent, R. P. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

Cao, Y. and Yang, J. Towards making systems forget with machine unlearning. In *Proc. IEEE S&P*, pp. 463–480, 2015.

Carreras, F. and Puente, M. A. Multinomial probabilistic values. *Group Decision and Negotiation*, 24(6):981–991, 2015.

Chalkiadakis, G., Elkind, E., and Wooldridge, M. Computational aspects of cooperative game theory. In Brachman, R. J., Cohen, W. W., and Dietterich, T. G. (eds.), *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2011.

Chen, J., Low, K. H., and Tan, C. K.-Y. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*, 2013.

Chen, L., Feldman, M., and Karbasi, A. Weakly submodular maximization beyond cardinality constraints: Does

randomization help greedy? In *Proc. ICML*, pp. 804–813, 2018a.

Chen, L., Zhang, G., and Zhou, E. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Proc. NeurIPS*, pp. 5627–5638, 2018b.

Das, A. and Kempe, D. Algorithms for subset selection in linear regression. In *Proc. STOC*, pp. 45–54, 2008.

Domenech, M., Giménez, J. M., and Puente, M. A. Some properties for probabilistic and multinomial (probabilistic) values on cooperative games. *Optimization*, 65(7):1377–1395, 2016.

Dütting, P., Fusco, F., Lattanzi, S., Norouzi-Fard, A., and Zadimoghaddam, M. Deletion robust submodular maximization over matroids. In *Proc. ICML*, pp. 5671–5693, 2022.

Farrand, T., Mireshghallah, F., Singh, S., and Trask, A. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proc. CSS Workshop on Privacy-Preserving Machine Learning in Practice*, pp. 15–19, 2020.

Feige, U., Mirrokni, V. S., and Vondrák, J. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Friedman, J. H. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–67, 1991.

Gharan, S. O. and Vondrák, J. Submodular maximization by simulated annealing. In *Proc. SODA*, pp. 1098–1116, 2011.

Ghorbani, A. and Zou, J. Data Shapley: Equitable valuation of data for machine learning. In *Proc. ICML*, pp. 2242–2251, 2019.

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

Guo, C., Zhao, B., and Bai, Y. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

Hemachandra, A., Dai, Z., Singh, J., Ng, S.-K., and Low, B. K. H. Training-free neural active learning with initialization-robustness guarantees. In *Proc. ICML*, 2023.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *JASA*, 58(301):13–30, 1963.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gurel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. Towards efficient data valuation based on the Shapley value. In *Proc. AISTATS*, pp. 1167–1176, 2019.

Kazemi, E., Zadimoghaddam, M., and Karbasi, A. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *Proc. ICML*, pp. 2544–2553, 2018.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. Glister: Generalization based data subset selection for efficient and robust learning. In *Proc. AAAI*, pp. 8110–8118, 2021.

Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12 (2):479–502, 2002.

Krause, A. and Golovin, D. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *JMLR*, 9(93): 2761–2801, 2008.

Krause, A., Roper, A., and Golovin, D. Randomized sensing in adversarial environments. In *Proc. IJCAI*, pp. 2133–2139, 2011.

Kwon, Y. and Zou, J. Beta Shapley: a unified and noise-reduced data valuation framework for machine learning. Technical report, 2021.

Lapp, D. Heart Disease Dataset. URL https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset, 2019.

Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., and Rogers, A. Bounding the estimation error of sampling-based Shapley value approximation. arXiv:1306.4265, 2013.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. In Stoer, J. (ed.), *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, pp. 234–243. Springer Berlin Heidelberg, 1978.

Mirchandani, P. B. and Francis, R. L. *Discrete Location Theory*. Wiley, 1990.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proc. AAAI*, pp. 1812–1818, 2015.

Mirzasoleiman, B., Karbasi, A., and Krause, A. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *Proc. ICML*, pp. 2449–2458, 2017.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. Coresets for data-efficient training of machine learning models. In *Proc. ICML*, pp. 6950–6960, 2020.

Mitrovic, S., Bogunovic, I., Norouzi-Fard, A., Tarnawski, J. M., and Cevher, V. Streaming robust submodular maximization: A partitioned thresholding approach. In *Proc. NIPS*, pp. 4557–4566, 2017.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.

Orlin, J. B., Schulz, A. S., and Udwani, R. Robust monotone submodular function maximization. *Mathematical Programming*, 172:505–537, 2018.

Özcan, G., Moharrer, A., and Ioannidis, S. Submodular maximization via Taylor series approximation. In *Proc. SDM*, pp. 423–431, 2021.

Pace, R. K. and Barry, R. Sparse spatial auto-regressions. *Statistics and Probability Letters*, 33(3):291–297, 1997.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Rockafellar, R. T. and Uryasev, S. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

Sarykalin, S., Serraino, G., and Uryasev, S. Value-at-risk vs. conditional value-at-risk in risk management and optimization. In Chen, Z.-L. and Raghavan, S. (eds.), *State-of-the-art Decision-making Tools in the Information-intensive Age*, INFORMS TutORials in Operations Research, pp. 270–294. 2008.

Shapley, L. S. A value for n-person games. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games*, volume 2, pp. 307–317. Princeton Univ. Press, 1953.

Sim, R. H. L., Zhang, Y., Chan, M. C., and Low, B. K. H. Collaborative machine learning with incentive-aware model rewards. In *Proc. ICML*, pp. 8927–8936, 2020.

Sim, R. H. L., Xu, X., and Low, B. K. H. Data valuation in machine learning: "ingredients", strategies, and open challenges. In *Proc. IJCAI*, pp. 5607–5614, 2022.

Sviridenko, M. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Vondrák, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. STOC*, pp. 67–74, 2008.

Weber, R. J. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pp. 101–119, 1988.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *Proc. ICML*, pp. 1954–1963, 2015.

Wu, M., Jia, R., Lin, C., Huang, W., and Chang, X. Variance reduced Shapley value estimation for trustworthy data valuation. *Computers & Operations Research*, 159: 106305, 2023.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747, 2017.

Yan, S., Kao, H.-t., and Ferrara, E. Fair class balancing: Enhancing model fairness without observing sensitive attributes. In *Proc. CIKM*, pp. 1715–1724, 2020.

Yu, J. and Ahmed, S. Maximizing expected utility over a knapsack constraint. *Operations Research Letters*, 44(2): 180–185, 2016.

Yu, Y.-L. Submodular analysis, duality and optimization, 2015. URL http://www.cs.cmu.edu/~yaoliang/mynotes/submodular.pdf.

Zhan, X., Wang, Q., Huang, K.-h., Xiong, H., Dou, D., and Chan, A. B. A comparative survey of deep active learning. arXiv:2203.13450, 2022.

Zhou, L. and Tokekar, P. Risk-aware submodular optimization for multirobot coordination. *IEEE T-RO*, 38(5):3064–3084, 2022.

A. Suitable Scenarios

In Sec. 5 and App. H, we show that DADS leads to a more apparent improvement in post-deletion predictive performance when fewer data can be selected (a limited budget), the staying probability is low or the staying probabilities vary across data owners (there exists similar data with different staying probabilities).

In Tab. 1 and Fig. 1, we describe some scenarios where DADS will be more useful. On the other hand, DADS may be less relevant when the learner does not require consent to access and train the data and does not have a limited budget constraint. For example, if only 1000 points are needed to achieve a high model accuracy and the learner has the budget to acquire 5000 data, deletions may not affect model accuracy significantly.

	Setting	**Example**	**Impact of DADS**
1)	Learner requires data owners' legal consent to access and train on their data. Data owners would only grant learners *temporary* access to data for DS as continued access risks privacy and unauthorized use. After the data acquisition period, the learner can only access and train on the data of selected owners.	A healthcare startup wants to acquire sensitive health data/scans from the local population as it is unavailable online. Under the GDPR laws, the startup needs a patient's informed consent to use her data. During data acquisition, a patient grants the startup temporary access to her medical records stored on the national database. However, the patient would object to the learner preserving her data without compensation for her privacy loss and contribution.	DADS helps the learner to make good use of the temporary access to identify and select data owners whose data mitigate the fall in model performance should other owners delete in the future (e.g., DADS may select more common data instead of outlier data that only increase the model accuracy on rare occasions).
2)	Learner has a limited budget. We argue that the learner should not set aside part of its budget to acquire replacement data later as it undesirably sacrifices better current data utility.	A startup may have limited funding for data acquisition (in a year) and can only afford to acquire data from a subset of all feasible data owners.	DADS helps the learner to wisely spend the budget on data owners who are less likely to delete among owners with similar data.
3)	Learner does not have the logistics to acquire more replacement data after the initial data acquisition. Alternatively, learners may also find it tedious to attract and inform data owners after every data deletion and prefer doing data acquisition at regular intervals (e.g., annually) instead.	A startup may only want to do a *one-time data acquisition* initiative (e.g., advertisements and campaign) to collect data from independent data owners. They may not have the resources to react to data deletions subsequently e.g., publicize data acquisition and communicate with data owners to seek informed consent. Data owners may also become less interested and aware of the data acquisition without these new initiatives.	DADS helps to proactively maximize the post-deletion data utility without or before the next data acquisition.

Table 1: Description on scenarios where DADS is more useful.

B. Overview and Summary of Notations

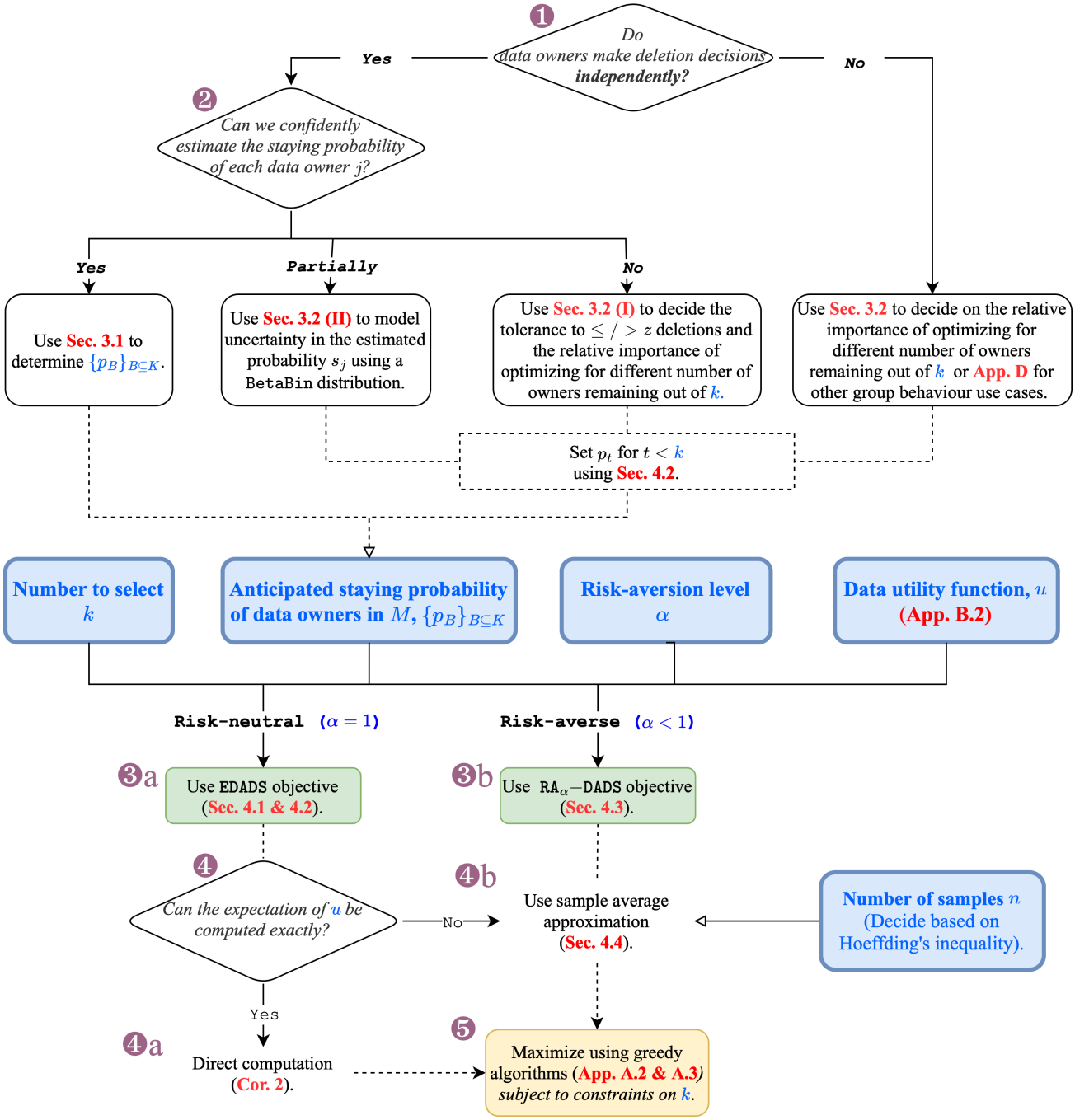


Figure 7: An overview of our DADS objectives and contributions from the learner's perspective. The learner sets the parameters (in blue) and follows our instructions/contributions (detailed in the referenced section) to decide on the anticipated staying probabilities, DADS objective, and the optimization technique.

B.1. Direct Computation

If u is multilinear, the EDADS objective $u_{\mathbb{E}}$ can be computed in closed form according to Cor. 2. Then, $u_{\mathbb{E}}$ can be maximized by using existing greedy algorithms.

B.2. Sample Average Approximation for EDADS

In Sec. 4, we show that the EDADS objective will preserve the monotonicity and submodularity property of u if (i) the data owners decide independently (Sec. 4.1) or (ii) the learner decides on the probability a number of owners stay out of k and sets the other pmf's using Sec. 4.2. Thus, the greedy algorithm can be applied.

We illustrate the greedy algorithm for monotone submodular functions in Algo. 1. The number of marginal gain evaluations is $\mathcal{O}(k|M|n)$, thus the time complexity is $\mathcal{O}(k|M|n)$ multiplied by the time needed to compute a marginal gain (such as $\mathcal{O}(k|M|)$ for the diversity data utility function mentioned in Sec. 4.4). In practice, it is possible to reduce the factor $|M|$ by using the lazy greedy algorithm (App. C.1). It is also possible to reduce the factor n when the data utility function (or marginal gain) for multiple samples can be simultaneously computed using vectorization and matrix operations (e.g., nearest neighbor submodular function (Wei et al., 2015), diversity function (Sim et al., 2020), variance-reduction function and EV-GP criterion (Krause et al., 2008; Hemachandra et al., 2023)). Moreover, the learner should compute the marginal gain Δ_u more efficiently (incrementally) by storing $\mathcal{O}(n)$ times the information than when computing the DS objective.

When data utility function u is non-monotone submodular, the learner could use other algorithms (Feige et al., 2011; Gharan & Vondrák, 2011) instead.

Algorithm 1 Sample average approximation for EDADS.

Input: Number to select k, feasible set M, data utility function u, (optional) efficient function to compute marginal gain Δ_u, the anticipated staying probability for (i) independent decisions $\{s_j\}_{j\in M}$ or (ii) dependent decisions p_k, number of samples n.
Output: Selected set $K \subseteq M$, $|K| \leq k$ to be acquired.

Initialize selected set $K \leftarrow \emptyset$
Initialize sampled subsets remaining out of selected set $\mathcal{C} = (C^1, \ldots, C^n) \leftarrow (\emptyset, \ldots, \emptyset)$
For dependent decisions, extend p_k to other sizes $t < k$:
if dependent decisions **then**
  for t **in** $k-1 \ldots 0$ **do**
    for a **in** $0 \ldots t$ **do**
      Compute probability of a staying out of t: $p_t(a) \leftarrow p_{t+1}(a) + p_{t+1}(a+1)$
    end for
  end for
end if
Select k owners greedily:
for round t **in** $1 \ldots k$ **do**
  Initialize best index $j_{max} \leftarrow$ null
  Initialize n-tuple of singleton sets $\iota_{max} \leftarrow$ null, which denotes sampled presence/deletions of j_{max} in $C^1 \ldots C^n$
  Initialize largest marginal gain $\delta_{max} \leftarrow -\infty$
  for j **in** $M \setminus K$ **do**
    Initialize marginal gain $\delta \leftarrow 0$
    Initialize empty tuple of singleton sets recording j's presence/deletions $\iota_j \leftarrow ()$
    for sample C^i indexed i **in** $1 \ldots n$ **do**
      if independent decisions **then**
        Use staying probability s_j from Sec. 3.1
      else
        Compute dependent staying probability $s_j \leftarrow \frac{p_t(|C^i|+1)}{p_{t-1}(|C^i|)}$
      end if
      Sample a random value $\omega \sim \mathrm{U}(0,1)$ (s.t. $\omega < s_j$ with a probability of s_j)
      if $\omega < s_j$ **then**
        $\delta \leftarrow \delta + \Delta_u(j|C^i)/n$
        Append the singleton set $\{j\}$ to ι_j, i.e., $\iota_j \leftarrow \iota_j + (\{j\},)$
      else
        Append an empty set to ι_j, i.e., $\iota_j \leftarrow \iota_j + (\emptyset,)$

end if
end for
if $\delta \geq \delta_{max}$ **then**
$j_{max}, \delta_{max}, \iota_{max} \leftarrow j, \delta, \iota_j$
end if
end for
$K \leftarrow K \cup \{j_{max}\}$
$\mathcal{C} \leftarrow \mathcal{C} \uplus \iota_{max}$ where $\uplus$ denote the element-wise union
end for
return selected set K

B.3. Sample Average Approximation for $\texttt{RA}_\alpha\texttt{-DADS}$

In Sec. 4.3, we explain that the $\texttt{RA}_\alpha\texttt{-DADS}$ objective can be maximized by optimizing $H'(\tau)$ over τ using numerical optimization techniques. To evaluate H' for some τ, we exploit the property that taking expectation over the same distribution (ensured by Secs. 4.1 and 4.2) preserves monotonicity and submodularity of the set function u. Thus, the greedy algorithm can be applied.

We illustrate the algorithm of Zhou & Tokekar (2022) for maximizing CVaR_α on monotone submodular functions in Algo. 2. The number of marginal gain evaluations is $\mathcal{O}(k|M|n)$ multiplied by the number of iterations for τ. In practice, it is possible to reduce the factor $|M|$ using the lazy greedy algorithm (App. C.1). It is also possible to reduce the factor n when the data utility function marginal gains for multiple samples can be simultaneously computed using vectorization and matrix operations.

Algorithm 2 Sample average approximation for $\texttt{RA}_\alpha\texttt{-DADS}$. The parameter τ is bounded above by $\max_{|B|\leq k} u(B)$ which can be estimated from the result of DS.

Input: Number to select k, feasible set M, data utility function u, (optional) efficient function to compute marginal gain Δ_u, the anticipated staying probability for (i) independent decisions $\{s_j\}_{j\in M}$ or (ii) dependent decisions p_k, risk-aversion level α, level of absolute tolerance τ_ϵ, number of samples n.
Output: Selected set $K \subseteq M$, $|K| \leq k$ to be acquired.

For dependent decisions, extend p_k to other sizes $t < k$:

if dependent decisions **then**
for t **in** $k-1 \dots 0$ **do**
for a **in** $0 \dots t$ **do**
Compute probability of a staying out of t: $p_t(a) \leftarrow p_{t+1}(a) + p_{t+1}(a+1)$
end for
end for
end if

Repeatedly select k owners greedily till convergence:

while not converged within $\pm\tau_\epsilon$ **do**
Select τ using a method that finds minimizer of scalar functions without derivatives, e.g., Brent's method (Brent, 2013)
Initialize selected set $K \leftarrow \emptyset$
Initialize sampled subsets remaining out of selected set $\mathcal{C} = (C^1, \dots, C^n) \leftarrow (\emptyset, \dots, \emptyset)$
Initialize vector of sampled utility values $\mathbf{v} = (v^i)_{i=1}^n \leftarrow (0)_{i=1}^n$
for round t **in** $1 \dots k$ **do**
Initialize best index $j_{max} \leftarrow \texttt{null}$
Initialize n-tuple of singleton sets $\iota_{max} \leftarrow \texttt{null}$, which denotes sampled presence/deletions of j_{max} in $C^1 \dots C^n$
Initialize vector of sampled utility values after adding j_{max}, $\mathbf{v}_{max} = (v^i)_{i=1}^n \leftarrow (0)_{i=1}^n$
Initialize largest marginal gain $\delta_{max} \leftarrow -\infty$
for j **in** $M \setminus K$ **do**

Initialize marginal gain $\delta \leftarrow 0$
Initialize empty tuple of singleton sets recording j's presence/deletions $\iota_j \leftarrow ()$
Initialize vector of sampled utility values after adding j, $\mathbf{v}_j = (v^i)_{i=1}^n \leftarrow (0)_{i=1}^n$
for sample C^i indexed i **in** $1 \dots n$ **do**
  if independent decisions **then**
    Use staying probability s_j from Sec. 3.1
  else
    Compute dependent staying probability $s_j \leftarrow \frac{p_t(|C^i|+1)}{p_{t-1}(|C^i|)}$
  end if
  Sample a random value $\omega \sim \mathrm{U}(0,1)$ (s.t. $\omega < s_j$ with a probability of s_j)
  if $\omega < s_j$ **then**
    Compute new value $v_j^i \leftarrow v^i + \Delta_u(j|C^i)$
    Update the i-th component of $\mathbf{v}_j$ to v_j^i
    Compute the threshold gain
$\Delta_H(j|C^i) \leftarrow (\tau - \frac{1}{\alpha}\max(0, \tau - v_j^i)) - (\tau - \frac{1}{\alpha}\max(0, \tau - v^i)) = \frac{1}{\alpha}(\max(0, \tau - v^i) - \max(0, \tau - v_j^i))$
    $\delta = \delta + \Delta_H(j|C^i)/n$
    Append the singleton set $\{j\}$ to ι_j, i.e., $\iota_j \leftarrow \iota_j + (\{j\},)$
  else
    Append an empty set to ι_j, i.e., $\iota_j \leftarrow \iota_j + (\emptyset,)$
  end if
 end for
 if $\delta \geq \delta_{max}$ **then**
  $j_{max}, \delta_{max}, \iota_{max}, \mathbf{v}_{max} \leftarrow j, \delta, \iota_j, \mathbf{v}_j$
 end if
 end for
 $K \leftarrow K \cup \{j_{max}\}$
 $\mathcal{C} \leftarrow \mathcal{C} \cup \iota_{max}$ where $\cup$ denote the element-wise union
 end for
end while
return selected set K

B.4. Table of Notations

Notation	Meaning
Variables and Sets	
A	Subset of B that stays present after deletions
a	Number of owners staying present
α	VaR or CVaR risk-aversion level
α, β	Beta-binomial distribution parameters
B	Subset of M; possible selected set
B_k	Subset of M of size k
B_t	Subset of M of size t; possible selected set in round t
C	Subset of K
C^i	i-th sampled subset for sample average approximation
D	Set of deleted owners (Sec. 2)
d	Maximum number of deletions (Sec. 2)
$\overline{\Delta}_t$	Largest marginal gain computed in the t-th round of selection (Sec. 4.4)
δ_j^i	Random variable of the marginal gain (after deletions) of adding owner j to the sampled set C^i (Sec. 4.4)
ϵ	Bound on the scaled absolute difference $\lvert u_{\text{avg}}(A) - u_{\mathbb{E}}(A)\rvert / \max_B u_{\mathbb{E}}(B)$
ε	Bounding term in Hoeffding's inequality
γ	Scale parameter in the mutual information/diversity function
$\mathbb{I}_j$	Indicator variable that indicates if j stays
$\mathbb{I}_\ell$	Indicator variable that indicates if ℓ stays
$\mathbf{i}$	Random vector of 0 or 1 that indicates if every element stays (Sec. 4.4)
$\mathcal{J}_\ell$	Subset of M (Sec. 4.4 definition of multilinear functions)
j	Data owner; element of M
K	Selected training set
K_t	Selected training set after round t
k	Maximum number of owners to select
l	Data owner; element of M
λ	Weight used to "split" the pmf at $u_{\text{VaR}_\alpha}(B)$
M	Feasible set to select owners from
n	Number of i.i.d. samples of C^i
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers
r_a	Probability exactly a owners stay out of k
$(r_a)_0^k$	Probabilities of exactly any number of owners staying out of k, i.e., $(r_0, r_1, \ldots, r_k)$
ρ	Approximation ratio for algorithm that solves the SAA problem
s	Independent staying probability
$\mathbf{s}$	Vector of independent staying probabilities
s_j	Independent staying probability of j
s_ℓ	Independent staying probability of ℓ
s_+	Independent staying probability of owners in the positive class
s_-	Independent staying probability of owners in the negative class
$\bar{s}$	Simulated staying probability (Sec. 5)
σ^2	Problem specific constant that affects the number of SAA samples needed
T	Subset of unselected owners
t	t-th round of selection
τ	Real number $\geq$ the random data utility $u(A)$ where $A \sim p_B(\cdot)$ with probability $\leq \alpha$
z	Number of deletions
Functions and Distributions	
$\lvert A\rvert$	Cardinality of A
$\texttt{BetaBin}(\cdot)$	Beta-binomial distribution
$\Delta_u(j\lvert A)$	Marginal gain in u of adding element j to set A
$\delta(\cdot)$	Dirac delta distribution
$H(B, \tau)$	Optimization objective in the alternative definition of CVaR (Eq. 3)
$\binom{k}{a}$	Number of ways to choose a elements out of k, i.e., $k!/(a!(k-a)!)$
$\mu(\mathbf{v})$	Multilinear function over vector $\mathbf{v}$
$\mathcal{O}(\cdot)$	Big O complexity
$p_B(A)$	pmf that maps each subset $A \subseteq B$ to the probability only A stays (out of B)
$p_K(C)$	pmf that maps each subset $C \subseteq K$ to the probability only C stays (out of K)
$p_k(a)$	pmf over $\{0, 1, \ldots, k\}$ that maps each a to the probability only a specific subset of size a (out of k) stays present ($= r_a / \binom{k}{a}$)
$p_M(\cdot)$	pmf that maps each subset $\cdot \subseteq M$ to the probability that only $\cdot$ stays (out of M)
$p_t(a)$	pmf over $\{0, 1, \ldots, t\}$ that maps each a to the probability only a specific subset of size a (out of t) stays present (used in round t)
$\texttt{U}(\cdot, \cdot)$	Uniform distribution over $[\cdot, \cdot]$
$u(\cdot)$	Data utility function used to evaluate set $\cdot$
$u_{\text{avg}}(\cdot)$	Mean data utility computed from n i.i.d samples of C^i
$u_{\mathbb{E}}(\cdot)$	Expected data utility function used to evaluate set $\cdot$; EDADS objective
$u_{\text{VaR}_\alpha}(\cdot)$	VaR of data utility function used to evaluate set $\cdot$
$u_{\text{CVaR}_\alpha}(\cdot)$	Risk-averse data utility function used to evaluate set $\cdot$; RA_α-DADS objective

Table 2: Summary of notations used in this paper.

C. Background and Related Works

C.1. Properties of Submodular Functions

Lemma 1 (Basic set operations preserve submodularity (Yu, 2015)). *Let the set function u that maps any subset $A \subseteq M$ to a non-negative value be (monotone) submodular. Then, for any subset $S \subseteq M$, the functions g and h that, respectively, consider the union and intersection with S, i.e.,*

$$g(B) = u(S \cup B), \qquad h(B) = u(S \cap B)$$

are (monotone) submodular.

Lemma 2 (Non-negative linear combinations preserve submodularity). *Given r (monotone) submodular functions and non-negative weights $w_1, \ldots, w_r$, the sum function f defined such that $f_\Sigma(B) = \sum_{i=1}^{r} w_i f_i(B)$ is (monotone) submodular.*

Consider that each (monotone) submodular function f_i is generated from the same underlying function f but with different parameters ξ, i.e., $f_i(B) = f(B; \xi)$. Given the distribution p_ξ over ξ, the expectation $\mathbb{E}_{\xi \sim p_\xi(\cdot)}[f(B; \xi)]$ is (monotone) submodular as the weights correspond to non-negative probabilities.

The maximization of submodular functions under cardinality constraints has been studied extensively.

Monotone submodular functions. For non-negative and monotone functions, the greedy algorithm (Nemhauser et al., 1978) iteratively selects the element with the largest marginal gain and returns a solution with a value at least $(1 - 1/e)$ of the optimal solution. The greedy algorithm has a complexity of $\mathcal{O}(k|M|)$ where k is the number of elements selected and $|M|$ is the size of the feasible set. This complexity may be computationally expensive when evaluating the set function u is costly. In practice, the *lazy greedy algorithm* (Minoux, 1978) can be an order of magnitude faster than the greedy algorithm. The lazy greedy algorithm maintains a priority queue of unselected elements. Each element's priority is its last evaluated marginal gain. At each round t, let K_{t-1} denote the set of elements selected before round t. The algorithm repeatedly dequeues the element (denoted as j) with the largest priority, evaluates its marginal gain $\Delta(j|K_{t-1})$ in round t, and re-inserts it into the priority queue. The evaluations pause when an element j_t is dequeued twice and the algorithm selects j_t (i.e., $K_t = K_{t-1} \cup \{j_t\}$). The justification is: j_t's marginal gain $\Delta(j_t|K_{t-1})$ is greater than (or equal to) the last evaluated marginal gain of every other unselected element j in $M \setminus K_{t-1}$. By the definition of submodularity, j's last evaluated marginal gain is at least its marginal gain in round t. Thus, the lazy greedy algorithm has selected the element j_t with the largest marginal gain in round t without evaluating all marginal gains. Mirzasoleiman et al. (2015) proposes the *stochastic greedy* algorithm to further reduce the number of evaluations but can only achieve a slightly weaker approximation guarantee.

Constraints. In our paper, we mainly focus on the cardinality constraint — the selected set size must not exceed k. However, it is possible that the elements (e.g., data owners) have non-uniform costs and the learner maximizes the data utility function u subject to the knapsack constraint that the total cost cannot exceed a budget.

We refer the reader to (Krause & Golovin, 2014) for a more complete survey on submodular function maximization, including algorithms to maximize non-monotone submodular functions (Feige et al., 2011; Gharan & Vondrák, 2011) and algorithms to maximize monotone submodular functions under knapsack constraints (Sviridenko, 2004).

C.2. Supervised Data Selection and Active Learning Submodular Functions

The **nearest neighbor (NN) submodular function** (Wei et al., 2015) measures the representativeness of a set S about the data partition for every class. When there is only 1 class, the NN submodular function corresponds to the **facility location function** (Mirchandani & Francis, 1990). The NN submodular function is

$$u^{NN}(S) = \sum_{y \in \mathcal{Y}} \sum_{i \in M^y} \max_{j \in S \cap M^y} w(i, j) \,,$$

where w is the similarity measure (that returns a non-negative similarity value), $\mathcal{Y}$ is the set of possible classes and M^y is the part of the feasible set of class y. For example, $w(i, j)$ can be defined as pairwise distance between (i, j) subtracted from the maximum pairwise distance. As the NN submodular function is multilinear (Özcan et al., 2021), its expectation can be computed exactly and efficiently as described in Sec. 4.4. Alternatively, when we save the maximum similarity between each data point $i \in M$ and the set K_t, $\bar{w}_{i,K_t} \coloneqq \max_{j \in K_t} w(i, j)$, we can *incrementally* compute the maximum similarity

with set $K_t \cup \{j'\}$ by taking the maximum of the pair $(w(i,j'), \bar{w}_{i,K_t})$. During sample average approximation, we can use this incremental computation trick (and vectorize over samples) for greater efficiency.

CRAIG (Mirzasoleiman et al., 2020), a method to select a weighted coreset of the training data that closely estimates the full gradient, also involves maximizing the facility location function.

The **naïve Bayes submodular function** (Wei et al., 2015) measures the diversity of the feature coverage in S and is computed solely from the frequencies of data in the feasible set with class y and j-th feature value x_j. Formally, we denote the subset of the set S with class y and j-th feature value x_j as $S^{(y,x_j)}$, then,

$$u^{NB}(S) = \sum_{y \in \mathcal{Y}} \sum_{j} \sum_{x_j \in \mathcal{X}_j} |M^{(y,x_j)}| \log |S^{(y,x_j)}| \, .$$

When each owner is equally likely to delete their data (e.g., $p_t(A) = \texttt{BetaBin}(a|t,\alpha,\beta)/\binom{t}{a}$), we can compute $\mathbb{E}_{A\sim p_B(\cdot)}\left[u^{NB}(A)\right] = \sum_{y\in\mathcal{Y}} \sum_j \sum_{x_j \in \mathcal{X}_j} |M^{(y,x_j)}| \mathbb{E}_{A\sim p_{S^{(y,x_j)}}(\cdot)}\left[\log |A|\right]$ in closed form from the counts $|S^{(y,x_j)}|$ and known probabilities (e.g., `BetaBin` pmf).

The **mutual information (diversity) function** (Mirzasoleiman et al., 2017; Sim et al., 2020) measures the diversity of S or the entropy reduction of a Gaussian Process model from observing S. The function is

$$u^{MI}(S) = \log\det(\mathbf{I} + \gamma \mathbf{K}_S) \, ,$$

where $\mathbf{K}_S$ is a principal submatrix (indexed by S) of the positive semi-definite similarity kernel $\mathbf{K}$ and $\gamma > 0$ is a scale parameter. The mutual information function is monotone submodular as the matrix $\mathbf{I} + \gamma \mathbf{K}_S$ has a minimum eigenvalue ≥ 1.

The **variance reduction function** (Krause et al., 2008; Hemachandra et al., 2023) measures the total reduction in predictive variance across a target set T after observing S.[6] Let $\sigma^2_{\mathcal{M}}(\mathbf{x}|S)$ denote the (predictive) output variance at $\mathbf{x}$ after training the model $\mathcal{M}$ on data points from the set S. The function is

$$u^{VR}(S) = \sum_{\mathbf{x}\in T} \sigma^2_{\mathcal{M}}(\mathbf{x}|\emptyset) - \sigma^2_{\mathcal{M}}(\mathbf{x}|S) \, .$$

Das & Kempe (2008) have shown that, in most cases, the variance reduction at any particular location is submodular. The function corresponds to the **expected variance with Gaussian process** (`EVGP`) criterion for neural active learning (Hemachandra et al., 2023) . `EVGP` is theoretically guaranteed to select data points which lead to trained neural networks with both good predictive performances and initialization robustness while not needing neural network training during data selection.

We approximate the `EDADS` objective for **mutual information (diversity) function** and **variance reduction function** with sample average approximation. We adopt (Chen et al., 2018b) approach of incrementally updating the Cholesky factor to efficiently compute the utility or marginal gain with a larger sampled staying subset. Moreover, we vectorize the computations over multiple staying subsets.

See (Guo et al., 2022; Zhan et al., 2022) for other supervised data subset selection and active learning functions.

C.3. Further Comments on Related Works

Our work does not fit the adaptive setting in that of Asadpour et al. (2008) and Golovin & Krause (2011). The adaptive setting differs from the non-adaptive setting as (i) it assumes that the learner will observe the state/realization of an element after it is selected, and (ii) the goal is to design an optimal, possibly non-deterministic, policy to select the next element based on the observed states so far. In our scenario, the learner will only know if any owner j deletes data after `DS` is over. Thus, the realization cannot influence the selection in the next round.

[6] If the learner does not have a target set for which the learner is particularly interested in accurate predictions, the target set T can be defined as the training set.

Our work also differs from the setting where the learner randomizes their selection. Krause et al. (2011) propose that the learner should find the optimal distribution p^* over feasible sets to maximize the *expected* worse-case objective value in adversarial environments (where the adversary has a finite number of strategies with different objective functions). In our work, we maximize the expected and conditional value-at-risk (instead of the adversarial case) based on the anticipated deletion probabilities and deterministically select a set of data owners.

D. Proofs for Sec. 4

D.1. Counterexamples

Consider a monotone submodular function u with $u(\emptyset) = 0$. We give some (extreme) counterexamples which show that flexibly/freely setting $\{p_B\}_{B \subseteq M}$ may violate monotonicity and submodularity.

Monotonicity. Let $A \subset B \subset C$. By setting

$$p_B(S) = \begin{cases} 1, & \text{if } S = B \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad p_C(S) = \begin{cases} 1, & \text{if } S = A \\ 0, & \text{otherwise} \end{cases},$$

the expected EDADS objective on the smaller subset B is not less than the larger subset C,

$$u_{\mathbb{E}}(B) = u(B) \geq u_{\mathbb{E}}(C) = u(A)\,.$$

Submodularity. Let $\{a\}$ and $\{b\}$ be singleton sets with elements (e.g., data owners) a and b, respectively. Let $C = \{a, b\}$, $u(\{b\}) = u_{\mathbb{E}}(\{b\}) = 0$ and $u(\{a\}) > 0$. By setting

$$p_{\{a\}}(S) = \begin{cases} 0.1, & \text{if } S = \{a\} \\ 0.9, & \text{if } S = \emptyset \end{cases} \quad \text{and} \quad p_C(S) = \begin{cases} 1, & \text{if } S = \{a\} \\ 0, & \text{otherwise} \end{cases},$$

(the presence of b encourages a to stay present), we have a larger marginal gain to the larger subset ($\{b\}$ vs. $\emptyset$),

$$\begin{aligned} \Delta_{u_{\mathbb{E}}}(a|\emptyset) &= u_{\mathbb{E}}(\{a\}) = 0.1 \cdot u(\{a\})\,, \\ \Delta_{u_{\mathbb{E}}}(a|\{b\}) &= u_{\mathbb{E}}(C) = u(\{a\}) > \Delta_{u_{\mathbb{E}}}(a|\emptyset)\,. \end{aligned}$$

D.2. Proof of Proposition 1

For any set $B \subseteq K$,

$$\begin{aligned} u_{\mathbb{E}}(B) &= \mathbb{E}_{A \sim p_B(\cdot)}\left[u(A)\right] \\ &= \sum_{A \subseteq B} p_B(A)\, u(A) \\ &\overset{1}{=} \sum_{A \subseteq B} \sum_{T \subseteq K \setminus B} p_K(A \cup T)\, u(A) \\ &\overset{2}{=} \sum_{C \subseteq K} p_K(C)\, u(A = (B \cap C)) \\ &= \mathbb{E}_{C \sim p_K(\cdot)}\left[u(B \cap C)\right]. \end{aligned}$$

In Step 1, we make use of $p_B(A) = \sum_{T \subseteq K \setminus B} p_K(A \cup T)$ for any $B \subseteq K$. In Step 2, both summations are combined by letting $C \coloneqq A \cup T$. The set A can then be recovered by taking the intersection of B and C.

The set function $u_{\mathbb{E}}$ is (monotone) submodular as for each C, $h_C(B) = u(C \cap B)$ is (monotone) submodular by Lemma 1.

D.3. Proof of Proposition 2

For any set B,

$$\begin{aligned} u_{\mathbb{E}}(B) &= \mathbb{E}_{A\sim p_B(\cdot)}\left[u(A)\right] \\ &= \sum_{A\subseteq B} p_B(A)\, u(A) \\ &\overset{1}{=} \sum_{A\subseteq B}\sum_{T\subseteq M\setminus B} p_M(A\cup T)\, u(A) \\ &\overset{2}{=} \sum_{C\subseteq M} p_M(C)\, u(A=(B\cap C)) \\ &= \mathbb{E}_{C\sim p_M(\cdot)}\left[u(B\cap C)\right]. \end{aligned}$$

In Step 1, we make use of $p_B(A)=\sum_{T\subseteq M\setminus B} p_M(A\cup T)$ (‡) for any $B\subseteq M$. In Step 2, by letting $C=A\cup T$, the set A can be recovered by taking the intersection of B and C.

The set function $u_{\mathbb{E}}$ is (monotone) submodular as for each C, $h_C(B)=u(C\cap B)$ is (monotone) submodular by Lemma 1.

D.4. Marginal Gain

For notational convenience, we denote $A\cup\{j\}$ as $A\cup j$.

When each data owner decides to delete their data independently, the expected marginal gain from adding j to the selected set K_t is

$$\begin{aligned} \Delta_{u_{\mathbb{E}}}(j|K_t) &= u_{\mathbb{E}}(K_t\cup j)-u_{\mathbb{E}}(K_t) \\ &= \mathbb{E}_{A\sim p_{K_t\cup j}(\cdot)}\left[u(A)\right]-\mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[u(A)\right] \\ &= \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[s_j\, u(A\cup j)+(1-s_j)\, u(A)-u(A)\right] \\ &= s_j\, \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[u(A\cup j)-u(A)\right] \\ &= s_j\, \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[\Delta_u(j|A)\right]. \end{aligned}$$

When each data owner's decision may depend on others, the expected marginal gain from adding j to the selected set K_t is

$$\begin{aligned} \Delta_{u_{\mathbb{E}}}(j|K_t) &= u_{\mathbb{E}}(K_t\cup j)-u_{\mathbb{E}}(K_t) \\ &= \mathbb{E}_{A\sim p_{K_t\cup j}(\cdot)}\left[u(A)\right]-\mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[u(A)\right] \\ &= \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[p_{j|A}\, u(A\cup j)+(1-p_{j|A})\, u(A)-u(A)\right] \\ &= \mathbb{E}_{A\sim p_{K_t}(\cdot)}\left[p_{j|A}\left[u(A\cup j)-u(A)\right]\right]. \end{aligned}$$

Note that $p_{j|A}$ may differ for A of different sizes and is equal to $\dfrac{p_{K_t\cup j}(A\cup j)}{p_{K_t}(A)}$.

D.5. Proof of Cor. 1

Direction 1: Assume $\forall B\subseteq K, p_B(A)=\sum_{T\subseteq K\setminus B} p_K(A\cup T)$. Prove $\forall B\subseteq K,\ j\in K\setminus B, p_B(A)=p_{B\cup j}(A)+p_{B\cup j}(A\cup j)$.

For any $j\in K\setminus B$,

$$\begin{aligned} p_B(A) &= \sum_{T\subseteq K\setminus B} p_K(A\cup T) \\ &= \sum_{T\subseteq K\setminus(B\cup j)} p_K(A\cup T)+\sum_{T\subseteq K\setminus(B\cup j)} p_K(A\cup j\cup T) \\ &= p_{B\cup j}(A)+p_{B\cup j}(A\cup j). \end{aligned}$$

Direction 2: Assume $\forall B \subseteq K,\ j \in K \backslash B, p_B(A) = p_{B\cup j}(A) + p_{B\cup j}(A \cup j)$. Prove $\forall B \subseteq K, p_B(A) = \sum_{T \subseteq K\backslash B} p_K(A \cup T)$ by recursively splitting each term on the RHS into two.

D.6. Proof: Beta-Binomial Distribution Satisfies Cor. 1

In Sec. 4.2, we simplify the probability of only set A of size a staying out of B_k of size k, $p_{B_k}(A) = r_{|A|}/\binom{k}{|A|}$, to $p_k(a) = r_a \cdot \binom{k}{a}^{-1}$ for notational convenience. Similarly, let $p_t(a)$ denote the probability of a subset of size a (out of t) staying present.

Let $p_{B_t}(A) = p_t(a) = \texttt{BetaBin}(a|t,\alpha,\beta)/\binom{t}{a}$. Similarly,

$$p_{B_t}(A \cup j) = p_t(a+1) = \texttt{BetaBin}(a+1|t,\alpha,\beta)/\binom{t}{a+1},$$
$$p_{B_{t-1}}(A) = p_{t-1}(a) = \texttt{BetaBin}(a|t-1,\alpha,\beta)/\binom{t-1}{a}.$$

Let $\texttt{B}$ denote the Beta function and Γ denote the gamma function, then

$$\begin{aligned}
&p_{B_t}(A) + p_{B_t}(A \cup j)\\
=&\texttt{BetaBin}(a|t,\alpha,\beta)/\binom{t}{a} + \texttt{BetaBin}(a+1|t,\alpha,\beta)/\binom{t}{a+1}\\
=&\frac{1}{\texttt{B}(\alpha,\beta)}\left[\texttt{B}(a+\alpha, t-a+\beta) + \texttt{B}(a+1+\alpha, t-a-1+\beta)\right]\\
=&\frac{1}{\texttt{B}(\alpha,\beta)\Gamma(a+\alpha+\beta)}\left[\Gamma(a+\alpha)\Gamma(t-a+\beta) + \Gamma(a+1+\alpha)\Gamma(t-a-1+\beta)\right]\\
=&\frac{\Gamma(a+\alpha)\Gamma(t-a-1+\beta)}{\texttt{B}(\alpha,\beta)\Gamma(t+\alpha+\beta)}\left[(t-a-1+\beta) + (a+\alpha)\right]\\
=&\frac{\Gamma(a+\alpha)\Gamma(t-a-1+\beta)}{\texttt{B}(\alpha,\beta)\Gamma(t+\alpha+\beta)}\left[(t-1+\beta+\alpha)\right]\\
=&\frac{\Gamma(a+\alpha)\Gamma(t-a-1+\beta)}{\texttt{B}(\alpha,\beta)\Gamma(t-1+\alpha+\beta)}\\
=&\frac{\texttt{B}(a+\alpha, t-1-a+\beta)}{\texttt{B}(\alpha,\beta)}\\
=&\texttt{BetaBin}(a|t-1,\alpha,\beta)/\binom{t-1}{a}\\
=&p_{B_{t-1}}(A).
\end{aligned}$$

D.7. Proof of Cor. 2

The proof of the case of independent decisions is reproduced from that of Özcan et al. (2021). As μ is multilinear, $\mu(\mathbf{v}) = \sum_{\ell=1}^{L}\left(c_\ell \prod_{j\in\mathcal{J}_\ell} v_j\right)$ with $\mathcal{J}_\ell \subseteq M$ for $\ell = 1,\dots,L$ for some L. Then,

$$\begin{aligned}
\mathbb{E}_{\mathbf{i}\sim\texttt{Bern}(\mathbf{s})}\left[\mu(\mathbf{i})\right] &= \sum_{\ell=1}^{L} c_\ell\, \mathbb{E}_{\mathbf{i}\sim\texttt{Bern}(\mathbf{s})}\left[\prod_{j\in\mathcal{J}_\ell} i_j\right]\\
&= \sum_{\ell=1}^{L} c_\ell \prod_{j\in\mathcal{J}_\ell} \mathbb{E}_{\mathbf{i}\sim\texttt{Bern}(\mathbf{s})}\left[i_j\right]\\
&= \sum_{\ell=1}^{L} c_\ell \prod_{j\in\mathcal{J}_\ell} s_j\\
&= \mu(\mathbf{s}).
\end{aligned}$$

The first equality is due to linearity of expectation. The second equality is because the expectation is a product of independent variables. The third equality is based on the expectation of Bernoulli random variables.

For the case of dependent decisions,

$$\begin{aligned}
\mathbb{E}_{\mathbf{i}_{B_k}\sim p_k(\cdot)}\left[\mu(\mathbf{i})\right] &= \sum_{\ell=1}^{L} c_\ell \, \mathbb{E}_{\mathbf{i}_{B_k}\sim p_k(\cdot)}\left[\prod_{j\in\mathcal{J}_\ell} i_j\right] \\
&= \sum_{\ell=1}^{L} c_\ell \, \mathbb{P}\left[\wedge_{j\in\mathcal{J}_\ell}(i_j=1)\right] \\
&= \sum_{\ell=1}^{L} c_\ell \, \mathbb{I}\left[\mathcal{J}_\ell\subseteq B_k\right] \, \mathbb{P}\left[\wedge_{j\in\mathcal{J}_\ell\cap B_k}(i_j=1)\right] \\
&= \sum_{\ell=1}^{L} c_\ell \, \mathbb{I}\left[\mathcal{J}_\ell\subseteq B_k\right] \sum_{\{\iota_{j'}\in[0,1]\}_{j'\in B_k\setminus\mathcal{J}_\ell}} \mathbb{P}\left[\wedge_{j\in\mathcal{J}_\ell\cap B_k}(i_j=1)\wedge_{j'\in B_k\setminus\mathcal{J}_\ell}(i_{j'}=\iota_{j'})\right] \\
&= \sum_{\ell=1}^{L} c_\ell \, \mathbb{I}\left[\mathcal{J}_\ell\subseteq B_k\right] \sum_{T\subseteq B_k\setminus\mathcal{J}_\ell} p_k(|\mathcal{J}_\ell|+|T|) \\
&= \sum_{\ell=1}^{L} c_\ell \, \mathbb{I}\left[\mathcal{J}_\ell\subseteq B_k\right] \, p_{|\mathcal{J}_\ell|}(|\mathcal{J}_\ell|)\,.
\end{aligned}$$

The first equality is due to linearity of expectation. The second equality is because $\prod_{j\in\mathcal{J}_\ell} i_j$ is only non-zero when every i_j is 1. The third equality is because any unselected owner $j'\in\mathcal{J}_\ell$ would make the probability $\mathbb{P}\left[\wedge_{j\in\mathcal{J}_\ell}(i_j=1)\right]=0$. Thus, the probability may only be positive when $\mathcal{J}_\ell\subseteq B_k$. The last equality is because the probability $p_{|\mathcal{J}_\ell|}(|\mathcal{J}_\ell|)$ of all $|\mathcal{J}_\ell|$ owners staying present (out of $\mathcal{J}_\ell$) is set according to Cor. 1 and Prop. 1.

E. Other Use Cases

Prop. 2 enables other unique use cases. Firstly, the learner can choose to maximize the DADS objective to actively select the set K but assume that subsequently selected points (set K') are unlikely to be deleted. This is achieved by setting $p_M(A\cup(M\setminus K))=p_k(A)$ for all subsets A of K and 0 otherwise (i.e., $p_M(B)=0$ when $(M\setminus K)\nsubseteq B$) while optimizing for K'. Points similar to K might be selected in K' (in anticipation of their deletions) but there will be less redundancy within K'.

Next, the learner can model group behavior. Suppose the M data owners can be partitioned into non-intersecting sets $\{E_i\}_{i=1}^n$ such that $\bigcup_{i=1\ldots n} E_i = M$. Suppose every owner in the same set E_i will not delete their data together and stays with probability s_{E_i}. Then, we can set the probability to 0 when A is not a union of $\{E_i\}_{i=1}^n$, i.e., $p_M(A)=0$ if $A\neq\bigcup E_i$. Else, $p_M(A)=\prod_{i:E_i\subseteq A} s_{E_i}\prod_{\ell:E_\ell\nsubseteq A}(1-s_{E_\ell})$. The other pmf's can be computed using Prop. 2.

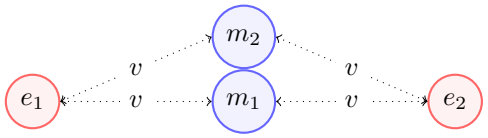
F. Example of Suboptimality of the Greedy Algorithm on EDADS vs. DS

In this section, we construct a toy dataset to understand why EDADS may be suboptimal vs. DS sometimes. We assume that every owner of a point has an independent staying probability s.

F.1. Independent Decisions

Num of deletions	Probability	$u(\{m,e\})$	$u(\{m_1,m_2\})$
0	s^2	$3+v$	$2+2v$
1	$s(1-s)$	$(2+2v)+(1+2v)=3+4v$	$(2+2v)+(2+2v)=4+4v$

Table 3: Probability of utilities when selecting 2 points: $\{m,e\}$ and $\{m_1,m_2\}$ respectively.



Subset S	$u(S)$
$\{m_1\}$, $\{m_2\}$ or $\{m_1, m_2\}$	$2+2v$
$\{e_1\}$ or $\{e_2\}$	$1+2v$
$\{e_1, e_2\}$	$2+2v$
$\{m_1, e_1\}$, $\{m_1, e_2\}$ $\{m_2, e_1\}$, $\{m_2, e_1\}$	$3+v$
$\{m_2, e_1, e_2\}$ or $\{m_2, e_1, e_2\}$ $\{m_1, m_2, e_1, e_2\}$	4

Figure 8: Let the similarity between any point and itself (and between (m_1, m_2)) be 1 and the similarity between any m and e, $w(m, e)$ is $v < 1$. Let u be the facility location utility function, $u(S) = \sum_{i \in \{m_1, m_2, e_1, e_2\}} \max_{j \in S} w(i, j)$, which is submodular. The set function is given in the table.

The greedy algorithm for DS will select $\{m, e\}$ in that order while that for EDADS will optimally select 2 similar points $\{m_1, m_2\}$ when $u_{\mathbb{E}}(\{m_1, m_2\}) > u_{\mathbb{E}}(\{m, e\})$. This condition holds when $s^2[(2+2v) - (3+v)] + s(1-s)[(4+4v) - (3+4v)]) = s^2[v-1] + s(1-s)[1] > 0$ or equivalently, the probability of staying $s < \frac{1}{2-v}$ (*) is small.

Num of deletions	Probability	$u(\{m, e_1, e_2\})$	$u(\{m_1, m_2, e\})$
0	s^3	4	$3+v$
1	$s^2(1-s)$	$2(3+v) + (2+2v) = 8+4v$	$2(3+v) + (2+2v) = 8+4v$
2	$s(1-s)^2$	$(2+2v) + 2(1+2v) = 4+6v$	$2(2+2v) + (1+2v) = 5+6v$

Table 4: Probability of utilities when selecting 3 points: $\{m, e_1, e_2\}$ and $\{m_1, m_2, e\}$ respectively.

The greedy algorithm for DS will select $\{m, e_1, e_2\}$. Given (*), the greedy algorithm for EDADS will select $\{m_1, m_2, e\}$ However, the former (i.e., set $\{m, e_1, e_2\}$) is optimal for the EDADS objective if $s^3[1-v] - s(1-s)^2[1] > 0$, i.e. $-vs^2 + 2s - 1 > 0$, or equivalently, the probability of staying is high enough $s > \frac{1-\sqrt{1-v}}{v}$ (*).

Thus, the greedy algorithm for EDADS may underperform when $s \in (\frac{1-\sqrt{1-v}}{v}, \frac{1}{2-v})$. For example, when $v = .5$, $s \in (0.568, 0.666)$. This is due to the myopic nature of greedy algorithm in introducing redundancy.

F.2. Dependent Decisions

Suppose we are selecting 3 points and only tolerant to ≤ 1 deletion.

Num of deletions	Probability	$u(\{m, e\})$	$u(\{m_1, m_2\})$
0	$1-2r$	$3+v$	$2+2v$
1	r	$(2+2v) + (1+2v) = 3+4v$	$(2+2v) + (2+2v) = 4+4v$

Table 5: Probability of utilities when selecting 2 points: $\{m, e\}$ and $\{m_1, m_2\}$ respectively.

The greedy algorithm for DS will select $\{m, e\}$ in that order while that for EDADS will optimally select $\{m_1, m_2\}$ when $u_{\mathbb{E}}(\{m_1, m_2\}) > u_{\mathbb{E}}(\{m, e\})$. This condition holds when $(1-2r)(v-1) + r > 0$ or equivalently, the probability of a subset with one deletion $r > \frac{1-v}{3-2v}$ (*). (When the probability is high (and deletions happen more often), more redundancy is preferred.)

Num of deletions	Probability	$u(\{m, e_1, e_2\})$	$u(\{m_1, m_2, e\})$
0	$1-3r$	4	$3+v$
1	r	$2(3+v)+(2+2v)=8+4v$	$2(3+v)+(2+2v)=8+4v$

Table 6: Probability of utilities when selecting 3 points: $\{m, e_1, e_2\}$ and $\{m_1, m_2, e\}$ respectively.

The greedy algorithm for DS will select $\{m, e_1, e_2\}$. Given (*), the greedy algorithm for EDADS have selected $\{m_1, m_2\}$ and will select $\{m_1, m_2, e\}$ by the last round.

However, as the set $\{m, e_1, e_2\}$ is always optimal for the EDADS objective, the greedy algorithm for EDADS may underperform when $r > \frac{1-v}{3-2v}$. For example, when $v = .5$, $r \in (.25, .333)$.

G. Connection to Cooperative Game Theory and Data Valuation

G.1. Cooperative Game Theory

Let N be a finite set of n players. A *cooperative game* on N is defined as a tuple $\langle N, v\rangle$. Here, the *characteristic function* v maps any subset (or *coalition*) C of players to a real number. The quantity $v(C)$ should interpreted as the value coalition C can achieve (in the absence of the remaining players). Let $\mathcal{G}_N$ denote the set of all cooperative games on players N. A *value* (function ϕ) on $\mathcal{G}_N$ maps each game with characteristic function v to a payoff vector $\phi[v]$ which assigns a value $\phi_i[v]$ to player i.

(Weber, 1988) defines a probabilistic value as a value that satisfies the following axioms:

- **Linearity.** For any characteristic functions v and v' and real number λ, $\phi[v+v'] = \phi[v] + \phi[v']$ and $\phi[\lambda v] = \lambda\phi[v]$;
- **Positivity.** If v is monotonic, then $\phi[v] \geq 0$;
- **Dummy player property.** A dummy player i's marginal contribution always equates its value, i.e., for any coalition $C \subseteq N \setminus i$, $v(C \cup i) - v(C) = v(i)$. If player $i \in N$ is a dummy, then $\phi_i[v] = v(i)$.

Probabilistic value can be characterized by the weighting coefficients $\{w_C^i \mid i \in N,\ C \subseteq N \setminus i\}$ and the expression

$$\forall i \in N \quad \phi_i[v] = \sum_{C \subseteq N \setminus i} w_C^i \left[v(C \cup i) - v(C)\right]$$

where the 2^{n-1} weighting coefficients for every player i must be non-negative and sum to 1, i.e., $\sum_{C \subseteq N \setminus i} w_C^i = 1$. The probabilistic value is a weighted sum of marginal contributions.

Semivalues are probabilistic values that satisfy the following additional property:

- **Anonymity.** For any permutation π that maps each player i in N to another player and permuted characteristic function πv such that $\pi v(C) = v(\{\pi(j) \mid j \in C\})$, $\phi_i[\pi v] = \phi_{\pi(i)}[v]$.

Semivalues can be characterized by fewer weighting coefficients $\{w_c\}_{c=0}^{n-1}$ and the expression

$$\forall i \in N \quad \phi_i[v] = \sum_{C \subseteq N \setminus i} w_{|C|} \left[v(C \cup i) - v(C)\right]$$

where all coalitions of a common size must share a common non-negative weight and $\sum_{c=0}^{n-1} w_c \binom{n-1}{c} = 1$. For example, the *Shapley value* (Shapley, 1953) is a semivalue with $w_c = 1/(n \times \binom{n-1}{c})$.

Multinomial probabilistic values are probabilistic values where the weighting coefficients

$$\forall i \in N,\ C \subseteq N \setminus i \quad w_C^i = \prod_{j \in C} s_j \prod_{\ell \in N \setminus (C \cup i)} (1 - s_\ell)\ .$$

Let $(w^{-j})_C^i$ denote the weighted coefficients for the cooperative game with $n-1$ players $N \setminus j$. A probabilistic value ϕ on $\mathcal{G}_N$ is *hereditary* iff

$$\forall C \subseteq N \setminus \{i,j\} \quad (w^{-j})^i_C = w^i_C + w^i_{C\cup j} \,. \tag{8}$$

The hereditary property entails that if player j is a *null* player (dummy player with no value and no marginal contribution, i.e., $v(j) = 0$), the probabilistic value and payoff of all players are not impacted by j's inclusion or exclusion. Multinomial probabilistic values satisfy the hereditary property. See (Domenech et al., 2016) for other properties of semivalues.

We observe that the hereditary property (Eq. 8) has the same structure as our recursive formula in Cor. 1.

G.2. Data Valuation

We can model the data valuation problem as a cooperative game where every data owner (or data) is a player. The characteristic function is the data utility function. For example, the value $v(C)$ can be the prediction performance (such as validation accuracy) of the model trained with data from C. A data owner's value is often set as the Shapley value (Ghorbani & Zou, 2019; Jia et al., 2019) or other semivalues such as the Beta Shapley value (Kwon & Zou, 2021). These semivalues ensure fairness — a null player will get no reward and two symmetric players with equal marginal contributions will get equal rewards.

G.3. Connection with Data Selection

At round $t+1$ of greedy data selection, we can consider a *cooperative game* from CGT with $(t+1)$ players (i.e., owners in the selected set K_t and another owner $j \in M \setminus K_t$) and define its characteristic function as the data utility function.

Conventional DS will select owner j with the largest marginal gain $\Delta_u(j|K_t)$. This corresponds to the leave-one-out value in data valuation.

Our EDADS objective with independent decisions will select owner j with the largest product of staying probability s_j and j's *multinomial probabilistic value* (Eq. 7) (Carreras & Puente, 2015). When each owner is equally likely to delete its data, our EDADS objective with dependent decisions will select owner j with the largest *semivalue* (with weight $p_t(a)$ on each subset of size a). For example, when the learner considers any number of deletions equally likely (Sec. 3.2 (**I**)), the selected owner j has the largest Shapley value (Shapley, 1953). So, our selection aligns with Ghorbani & Zou (2019) suggestion to improve a model by acquiring new data with high predicted Shapley value (an equitable data value). Ghorbani & Zou (2019) has shown that adding data with the largest Data Shapley (for the cooperative game with M players) leads to a faster increase in model performance than a random order or adding data with the highest leave-one-out value.

Differences. Our DADS approach is (i) tailored for the staying/deletion probability and (2) considers multiple games sequentially based on the selected data owners only. In Data Shapley, unselected owners (including duplicates) would affect the valuation and the selection.

Our work offers new suggestions for DV: Choose the semivalues based on the staying probabilities and consider multiple cooperative games based on the selected owners only.

On the other hand, our work can also *benefit from concepts in DV and CGT*. The ML learner can use the weights from semivalues in CGT/DV that satisfy the *hereditary property* (Domenech et al., 2016) (and hence our Cor. 1) to directly set $p_t(a)$ in the EDADS objective. For example, the learner can use the Shapley value's weight $1/\left((t+1)\times\binom{t}{a}\right)$ on each subset of size a.

H. Experiments

H.1. Experimental Details

We compare the performance of our deletion-anticipative DADS objectives vs. its corresponding conventional DS objective on a few **(submodular function-dataset)** combinations detailed below. We demonstrate the feasibility of our approach on multiple supervised data subset selection and active learning data utility functions that are easier to implement and understand. The datasets are chosen to demonstrate intended realistic applications of the DADS objectives (on health and income data) and the feasibility/performance on a large benchmark dataset.

(NN-S) the *nearest neighbor* (NN) submodular function on a 2-class Synthetic dataset. Each class has 50 points and the green class has 2 clusters (isotropic Gaussian blobs) with 35 and 15 points respectively. We design this dataset to

consider different staying probabilities across classes and observe the selection of points in the large and small clusters.

(NN-H) the NN submodular function on the combined Heart Disease dataset (Lapp, 2019) from four databases: Cleveland, Hungary, Switzerland, and Long Beach. The Heart Disease dataset consists of 1025 data points with 13 health features (e.g., chest pain type, resting blood pressure) and a target (of whether a patient has heart disease). We set aside 25% of the data as the validation set and use the remaining 75% as the feasible set (each owner owns a single datum, hence $|M| = 768$). We pre-process the data by min-max scaling to the $[0, 1]$ range. The selected subset is used to train an NN classifier with $L2$ as the distance metric.

We consider that a patient without the disease will never delete its data (i.e., staying probability $s_- = 1$) and vary the staying probability s_+ of every positive patient with the disease. The learner measures the F1 score, which balances the precision and recall of identifying heart disease patients, on the validation set.

To evaluate the $\texttt{RA}_\alpha\texttt{-DADS}$ objective, we set the number selected $k = 240$, the staying probability s_+ of a patient with the disease $= .5$ and use $n = 20000$ samples to estimate the expectation in the $\texttt{RA}_\alpha\texttt{-DADS}$ objective.

(NN-F) the NN submodular function on the benchmark Fashion MNIST (FMNIST) dataset (Xiao et al., 2017). The FMNIST dataset consists of 10 classes of clothing, each with 6000 training images and 1000 test/validation images. Each image is 28×28 pixels. We use the training images as the feasible set ($|M| = 60000$) and select a subset to train an NN classifier with $L2$ as the distance metric.

We vary the common staying probability s across all image owners in the feasible set. The learner measures the accuracy score on the validation set.

To evaluate the $\texttt{RA}_\alpha\texttt{-DADS}$ objective, we set the number selected $k = 500$, and model that the number of owners staying present out of 500 follows the distribution $\texttt{BetaBin}(n = 500, \alpha = 4, \beta = 2)$. We only include 2000 training images of each class in the feasible set ($|M| = 20000$) to reduce the computation time. We use $n = 20000$ samples to estimate the expectation in the $\texttt{RA}_\alpha\texttt{-DADS}$ objective.

(NB-A) the *naïve Bayes* (NB) submodular function on the Adults Income dataset (Becker & Kohavi, 1996). After dropping data with missing entries, the dataset consists of $|M| = 30718$ points in the feasible set and 15315 points in the validation set. Our dataset contains 10 categorical features (e.g., age, education, occupation) after removing redundant features (fnlwgt, number of years of education, native country and work country) and discretizing numerical features (e.g., capital gain, age, hours per week). The selected subset is used to train a Categorical NB model to predict whether a person has an income of at least $50k$. We fix the class prior based on the ratio in the feasible set (i.e., it does not depend on the selected set) and use a Laplace smoothing factor of .01.

We vary the common staying probability s across all image owners in the feasible set. As the dataset is imbalanced (e.g., only 7650 points of the positive class in the feasible set), the learner measures the *balanced* accuracy score on the validation set.

To evaluate the $\texttt{RA}_\alpha\texttt{-DADS}$ objective, we set the number selected $k = 140$, the staying probability s of a data owner $= .6$ and use $n = 10000$ samples to estimate the expectation in the $\texttt{RA}_\alpha\texttt{-DADS}$ objective.

(MI-S), (VR-S) the *mutual information* (MI) (diversity) submodular and *variance reduction* (VR) functions on the Singapore traffic dataset (Chen et al., 2013). The dataset includes 2506 input regions. Each region has a tuple of horizontal and vertical coordinates (x_0, x_1). Possible applications include the learner collecting data on traffic or environmental conditions (e.g., congestion, taxi demand) from owners (e.g., companies, taxi drivers) who have deployed a camera/sensor at each region. We consider a Gaussian process model with a squared exponential kernel. The lengthscales for the dimensions are set at 5 and 8, respectively. Other hyperparameters are specified in the code. We use $n = 10000$ samples to estimate the expectation in the $\texttt{EDADS}$ objective.

(MI-F) the MI (diversity) submodular function on the synthetic Friedman dataset (Friedman, 1991) with 5 input features and outputs perturbed by Gaussian noise with a standard deviation of .8. Our feasible and validation set consists of 3000 and 2000 points respectively. We standardize the output variable on the feasible set to have zero mean and unit variance.

We consider a Gaussian process model with an exponential kernel. We use a Gaussian likelihood and assume that the model hyperparameters (including separate lengthscales for each feature) are known or learned using maximum likelihood estimation. The hyperparameters are specified in the code. We use $n = 10000$ samples to estimate the expectation in the $\texttt{EDADS}$ objective.

The learner measures the log (Gaussian) predictive density (Rasmussen & Williams, 2006) on the validation set. A higher log predictive density means that it is more likely to observe the validation set given that we observe the training set. This occurs when the model makes confident and accurate predictions (with low mean squared error and predictive variance).

(VR-C) the VR function on the Californian housing dataset (Pace & Barry, 1997). The housing dataset consists of 20640 data points with 8 features (e.g., median income, latitude, longitude) and the output variable is the housing prices. To speed up computation, we select 2 sets of 6000 points as the feasible and the validation set. Each input and output variable is standardized to have zero mean and unit variance on the feasible set.

We consider a Gaussian process model and use a linear kernel for the median income feature and an exponential kernel for the rest. We use a Gaussian likelihood and assume that the model hyperparameters (including separate lengthscales for each feature) are known or learned using maximum likelihood estimation. The hyperparameters used are specified in the code. We use $n = 10000$ samples to estimate the expectation in the `EDADS` objective.

The learner measures the log (Gaussian) predictive density (Rasmussen & Williams, 2006) on the validation set.

(EVGP-C) the *expected variance with Gaussian process* (EVGP) function (Hemachandra et al., 2023) on the Californian housing dataset (Pace & Barry, 1997). The description of the dataset is the same as the previous bullet.

We consider a neural network with 2 hidden layers with 128 and 32 units and `ReLu` activation. The neural network is pre-trained on a small dataset (500 data points), simulating a historic or public dataset (Sim et al., 2020). Then, we compute its empirical neural tangent kernel to derive the `EVGP` objective. We use $n = 10000$ samples to estimate the expectation in the `EDADS` objective.

The learner measures the mean squared error on the test set achieved by the neural network after further training on data from the selected owners.

In the main paper, we compare the `DADS` objectives against the `DS` objective (as the baseline). In App. H.3, we also compared against random selection.

H.2. Computational Resources and Code

The experiments are run on a machine with Ubuntu 22.04.3 LTS, 2 x Intel Xeon Silver 4116 (2.1 GHz), and NVIDIA Titan RTX GPU (Cuda 11.7). The software environments used are Miniconda and Python. Please refer to our Github respository for the implementation details.

H.3. Independent Decisions

2D Visualization with Unequal Staying Probabilities across Classes

In Fig. 9, we mark the $k = 10$ points selected by `EDADS` and `DS` objective when u is the NN submodular function[7] and each green data owner independently decides to stay with probability s. We assume that the blue points will always stay (i.e. not be deleted).

We observe that as the staying probability s decreases, `EDADS` selects fewer of the blue data and more green data instead. This is because the additional green data can help preserve the representativeness of the green class if other selected owners delete their data as anticipated. Moreover, as the staying probability s decreases, the data selected are closer to the center of the class. As described earlier, these centers have higher net similarity to other green data and help preserve higher objective value if other selected green data owners delete their data as anticipated. This redundancy is achieved by forgoing the selection of data on the edge of the cluster (which have lower net similarity to all data).

[7]The NN submodular function (App. C.2) will actively select a subset that is *representative* of the feasible set.

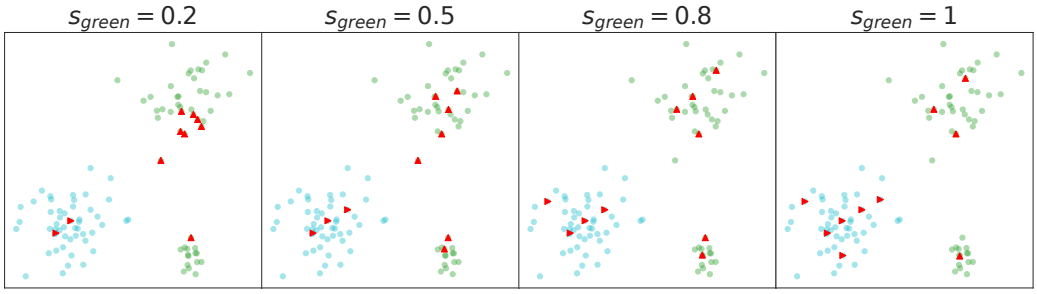


Figure 9: (NN-S) 10 data points (red) selected by EDADS with varying staying probability of the green class where $s = 1$ corresponds to the DS objective.

2D Visualization with Unequal staying probabilities across Data Owners

From Fig. 9, we observe that EDADS selects darker points with higher staying probability, thus preserving higher objective value if deletions happen as anticipated.

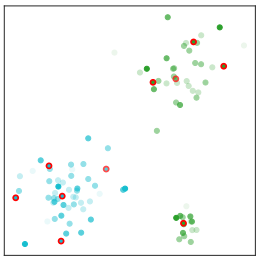


Figure 10: (NN-S) 10 data points (red) selected by EDADS with varying staying probability across data owners. Data with higher staying probability (i.e., less likely to be deleted) are plotted in a darker (more opaque) color.

Number of (NN-H) positive patients selected

From Fig. 11, we observe that EDADS selects more positive patients (with the disease) than DS and a larger number when the staying probability s_+ is lower. Thus, when deletions happen as anticipated, EDADS leads to a higher F1 score than DS (Fig. 4). We also include staying probability $s_+ = .8$ for a greater contrast against $s_+ = .5$.

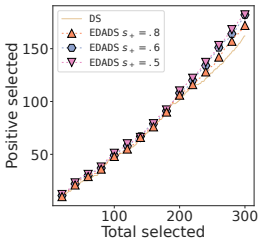
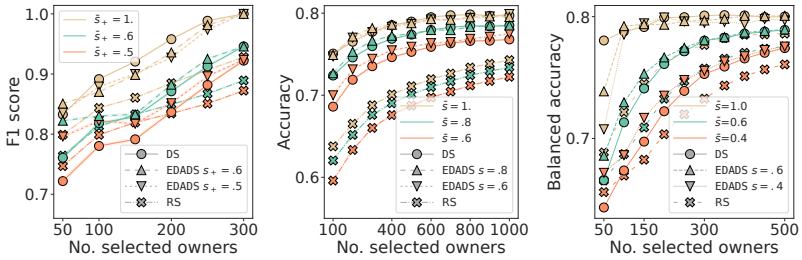


Figure 11: Number of positive patients selected vs. the total number of patients selected by DS and EDADS (under varying staying probability s_+) in the (NN-H) experiment.

Random Selection

As an additional benchmark, we compare EDADS and DS against random selection (RS) that selects the same proportion of each class as their proportion in the feasible set M. The metric score for RS is averaged over 10 random selection runs. From Fig. 12, we observe that EDADS outperforms RS when evaluated with and without deletions.



(a) (NN-H) (b) (NN-F) (c) (NB-A)

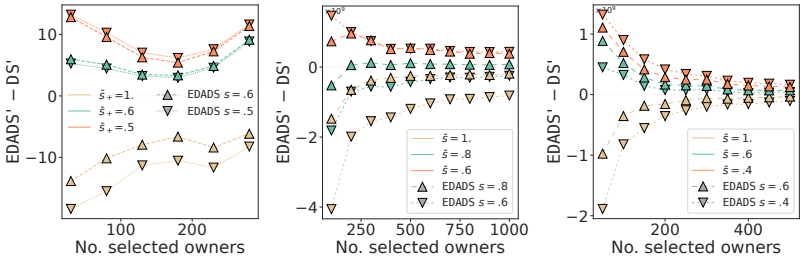
Figure 12: Graphs of mean validation set metric scores (with $2\times$ standard error (shaded) across 2500 simulations) with an increasing no. of data owners (horizontal axis) selected by DS, EDADS (with varying s) and random selection (RS) for various datasets (a-c) when owners stay with simulated staying probability $\bar{s}$.

Objective Values

In Sec. 5, we plot the validation set metrics (with and without deletions) against the number of owners selected. Now, we will verify the properties of EDADS and DS using the objective function values. In Fig. 13, we plot the difference in the expected objective value of the sets selected by EDADS and DS when points (of the positive class for (NN-H) and all for others) stay with simulated probability $\bar{s}$.

When owners stay with simulated probability $\bar{s}$, the subset selected by EDADS that optimizes for staying probability $s = \bar{s}$ achieves the highest objective value and is better than DS. This is consistent with Sec. 5.1 where EDADS outperforms DS on the validation set metrics.

When there are no deletions ($\bar{s} = 1.$), EDADS has a lower objective value than DS. This is consistent with the trade-off observed in Sec. 5.1 – EDADS achieves a poorer validation set metric than DS without deletions.



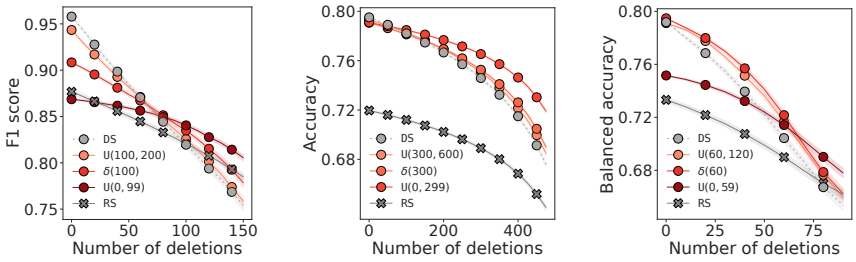
(a) (NN-H) (b) (NN-F) (c) (NB-A)

Figure 13: Difference in EDADS' (with different s) and DS's expected objective value for various datasets (a-c) when owners stay with simulated probability $\bar{s}$.

H.4. Dependent Decisions

Validation Set Metrics vs. Number of Deletions

In Fig. 14 (extension of Fig. 5), we plot the validation set metrics vs. the number of deletions when the learner considers Sec. 3.2, (I) tolerance to z deletions or (II) uncertainty in the staying probability. We observe that for different anticipated probabilities, the validation set metric falls at different rates as the number of deletions increases. DS has the steepest fall and worst metric when there are many deletions. In contrast, EDADS achieves a better metric score than DS when there are moderate and more deletions. Moreover, as the anticipated deletions increase (see darker lines corresponding to $\mathtt{U}(0,\cdot)$ and lower $\frac{\alpha}{\alpha+\beta}$ in BetaBin), the performance falls at a slower rate.



(a) (NN-H)(I), k=200 (b) (NN-F)(I), k=600 (c) (NB-A)(I), k=120

(a) (NN-H)(II), k=200 (b) (NN-F)(II), k=600 (c) (NB-A)(II), k=120

Figure 14: Graphs of mean validation set metric scores (with $2\times$ standard error (shaded) across 500 simulations) with an increasing no. of deletions (horizontal axis) for various datasets (a-c). k owners are selected by `DS` or `EDADS`. For `EDADS`, the no. of owners staying (out of k) follows (I) Uniform `U` or Dirac delta δ distribution and (II) `BetaBin` distribution.

Expected Objective Value and Metric with Deletions

Across all experiments (see Tabs. 7-10), we observe that `EDADS` achieves a higher expected objective value than `DS`. Moreover, `EDADS` achieves a better metric than `DS` most of the time. There is an exception for (NN-H) when 300 patients are selected (see Tab. 8). We hypothesize that this is due to a lower correlation between the objective value and validation set metric when many data owners have been selected. This low correlation happens because the validation set metric plateaus but slightly fluctuates after sufficient data is selected.

Distributions	`EDADS` $u_{\mathbb{E}}$	`DS` $u_{\mathbb{E}}$	`EDADS` expected F1	`DS` expected F1
$\delta(100)$	**4672.9978**	4660.7088	$\mathbf{0.8333 \pm 0.0306}$	0.8214 ± 0.0372
$\texttt{U}(100, 200)$	**4744.8274**	4742.6873	0.8829 ± 0.0415	$\mathbf{0.8877 \pm 0.0473}$
$\texttt{U}(0, 99)$	**4367.9313**	4329.2518	$\mathbf{0.7756 \pm 0.1167}$	0.7362 ± 0.1199
$\texttt{BetaBin}(200, 10, 10)$	**4666.1810**	4654.2744	$\mathbf{0.8277 \pm 0.0384}$	0.8218 ± 0.0479
$\texttt{BetaBin}(200, 4, 16)$	**4462.3355**	4412.6679	$\mathbf{0.7820 \pm 0.0663}$	0.7360 ± 0.0785
$\texttt{BetaBin}(200, 16, 4)$	**4759.7544**	4758.9481	$\mathbf{0.9027 \pm 0.0323}$	0.8996 ± 0.0351
$\texttt{BetaBin}(200, 1, 1)$	**4546.3604**	4536.9980	$\mathbf{0.8182 \pm 0.1054}$	0.8124 ± 0.1131

Table 7: (NN-H) The expected objective value $u_{\mathbb{E}}$ (exact) and F1 score (averaged over 2500 simulations) when 200 patients are selected. The number of owners staying follows the distribution column.

Distributions	EDADS $u_{\mathbb{E}}$	DS $u_{\mathbb{E}}$	EDADS expected F1	DS expected F1
$\delta(150)$	**4723.9851**	4702.4362	0.8655 ± 0.0286	$\mathbf{0.8870 \pm 0.0218}$
$\mathtt{U}(150, 300)$	**4782.2684**	4773.5301	0.9395 ± 0.0399	$\mathbf{0.9450 \pm 0.0363}$
$\mathtt{U}(0, 150)$	**4463.6384**	4439.5500	0.7992 ± 0.1072	$\mathbf{0.8087 \pm 0.0932}$
$\mathtt{BetaBin}(300, 10, 10)$	**4717.7756**	4697.3919	0.8652 ± 0.0423	$\mathbf{0.8861 \pm 0.0357}$
$\mathtt{BetaBin}(300, 4, 16)$	**4537.0203**	4501.9433	0.7974 ± 0.0577	$\mathbf{0.8046 \pm 0.0455}$
$\mathtt{BetaBin}(300, 16, 4)$	**4794.7828**	4787.6772	0.9510 ± 0.0276	$\mathbf{0.9574 \pm 0.0238}$
$\mathtt{BetaBin}(300, 1, 1)$	**4617.7797**	4607.0948	0.8552 ± 0.1032	$\mathbf{0.8757 \pm 0.0996}$

Table 8: (NN-H) The expected objective value $u_{\mathbb{E}}$ (exact) and F1 score (averaged over 2500 simulations) when k=300 patients are selected. The number of owners staying follows the distribution column.

Distributions	EDADS $u_{\mathbb{E}}$	DS $u_{\mathbb{E}}$	EDADS expected accuracy	DS expected accuracy
$\delta(300)$	$\mathbf{1.771008e12}$	$1.770023e12$	$\mathbf{0.75288 \pm 0.00692}$	0.74605 ± 0.00762
$\mathtt{U}(300, 600)$	$\mathbf{1.778268e12}$	$1.778143e12$	$\mathbf{0.77506 \pm 0.01261}$	0.77338 ± 0.01458
$\mathtt{U}(0, 299)$	$\mathbf{1.685616e12}$	$1.677902e12$	$\mathbf{0.68815 \pm 0.11339}$	0.65207 ± 0.11589
$\mathtt{BetaBin}(600, 10, 10)$	$\mathbf{1.770155e12}$	$1.769113e12$	$\mathbf{0.75304 \pm 0.01553}$	0.74384 ± 0.01952
$\mathtt{BetaBin}(600, 4, 16)$	$\mathbf{1.738350e12}$	$1.730641e12$	$\mathbf{0.70435 \pm 0.04659}$	0.65968 ± 0.05495
$\mathtt{BetaBin}(600, 16, 4)$	$\mathbf{1.779854e12}$	$1.779752e12$	$\mathbf{0.77935 \pm 0.00835}$	0.77888 ± 0.00900
$\mathtt{BetaBin}(600, 1, 1)$	$\mathbf{1.731039e12}$	$1.728106e12$	$\mathbf{0.72808 \pm 0.09720}$	0.71175 ± 0.10378

Table 9: (NN-F) The expected objective value $u_{\mathbb{E}}$ (exact) and accuracy score (averaged over 2500 simulations) when k=600 image owners are selected. The number of owners staying follows the distribution column.

Distributions	EDADS $u_{\mathbb{E}}$	DS $u_{\mathbb{E}}$	EDADS expected balanced acc.	DS expected balanced acc.
$\delta(60)$	**681161.5499**	675973.9621	$\mathbf{0.72219 \pm 0.03158}$	0.70437 ± 0.03223
$\mathtt{U}(60, 120)$	**818162.0060**	816669.6247	$\mathbf{0.76095 \pm 0.03027}$	0.75243 ± 0.03340
$\mathtt{U}(0, 59)$	**249239.7757**	244108.7184	$\mathbf{0.66561 \pm 0.06711}$	0.65098 ± 0.06470
$\mathtt{BetaBin}(120, 10, 10)$	**666858.1633**	662290.6454	$\mathbf{0.72194 \pm 0.04122}$	0.70401 ± 0.04143
$\mathtt{BetaBin}(120, 4, 16)$	**250173.9232**	241441.1907	$\mathbf{0.67165 \pm 0.05032}$	0.65067 ± 0.05264
$\mathtt{BetaBin}(120, 16, 4)$	**845700.3327**	844590.7931	$\mathbf{0.76996 \pm 0.02054}$	0.76202 ± 0.02387
$\mathtt{BetaBin}(120, 1, 1)$	**534420.8463**	532755.1257	$\mathbf{0.71145 \pm 0.07027}$	0.69995 ± 0.07383

Table 10: (NB-A) The expected objective value $u_{\mathbb{E}}$ (exact) and balanced accuracy score (averaged over 2500 simulations) when k=120 data owners are selected. The number of owners staying follows the distribution column.

2D VISUALIZATION

In Fig. 15, we mark the 10 points selected by EDADS and DS objectives when u is the NN submodular function and the number of owners staying follows some Uniform U, Dirac delta δ or BetaBin distribution. Note that across each row in Fig. 15, the weights of smaller subsets increase, i.e., more deletions are anticipated (see Fig. 16). We observe that as the weights of smaller subsets increase, the data selected are closer together and to the center of the class. As described earlier, these centers have higher net similarity to other data and help preserve higher objective value if other selected owners delete their data as anticipated. This redundancy is achieved by forgoing the selection of data on the edge of the cluster (which have lower net similarity to others).

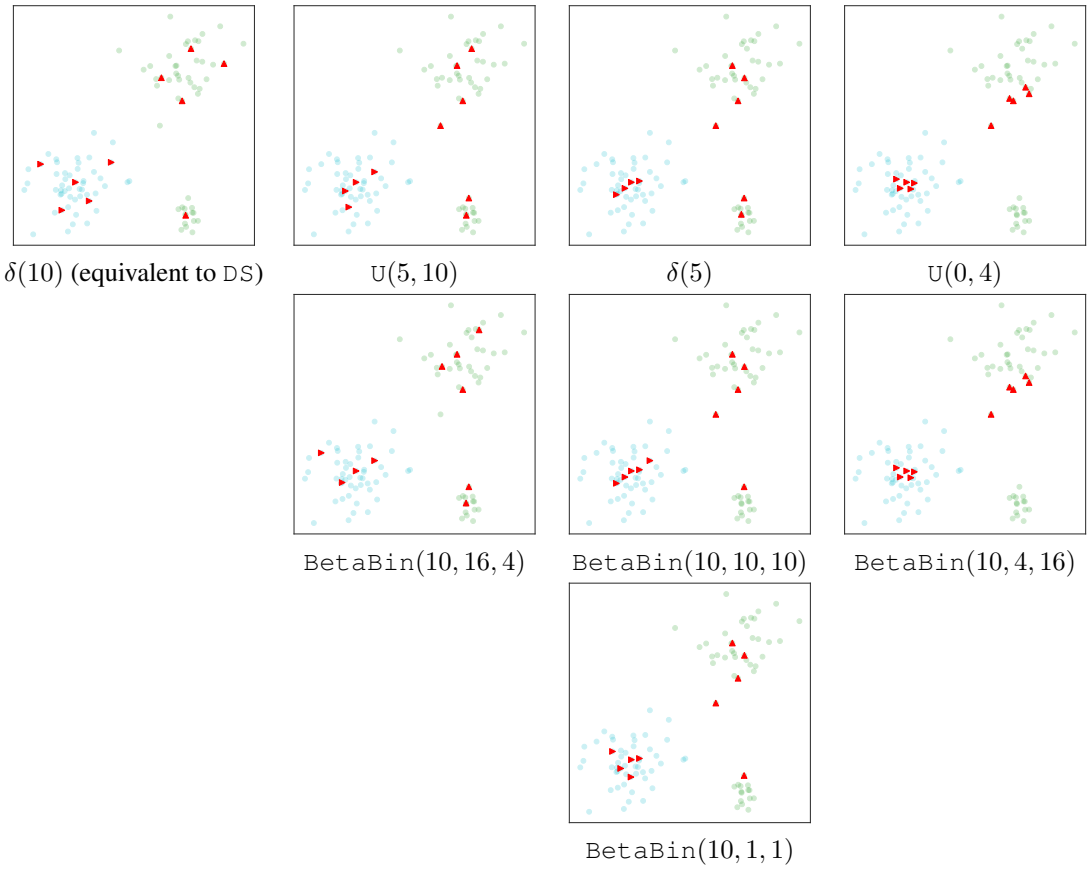


Figure 15: (NN-S) 10 data points (red) selected by EDADS when the learner considers the number of owners staying follows the distribution below each sub-figure in the EDADS objective. Across each row (towards the right), the weights on smaller subsets increase, i.e., more deletions are anticipated.

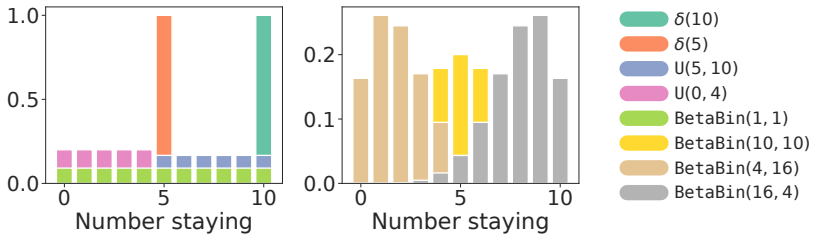


Figure 16: The learner's relative priority/weights on different numbers of data owners staying out of $k = 10$ for the distributions considered in Fig. 15.

Number of (NN-H) positive patients staying follows a different distribution

We consider that a patient without the disease will never delete its data (i.e., staying probability $s_- = 1$) while the learner is uncertain about the staying probability of positive patients. Thus, the number of positive patients staying out of t positive patients is assumed to follow BetaBin(t, α, β) for different values of (α, β) as described in Sec. 4.2.

In Fig. 17, we observe that with more anticipated deletions (lower $\frac{\alpha}{\alpha+\beta}$ in BetaBin), the metric under no deletions (leftmost) worsens but the performance falls at a slower rate as the number of deletions increases. We verify that when deletions happen according to our probability model, EDADS preserves a higher expected objective value and metric than DS in Tab. 11.

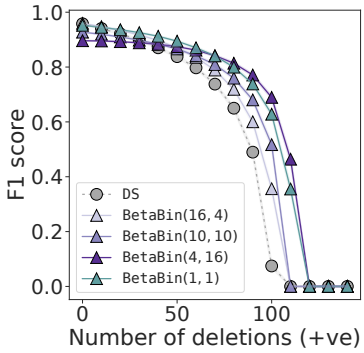


Figure 17: Graphs of mean F1 score achieved by DS and EDADS (when no. of positive patients staying follows different BetaBin distributions) vs. no. of deletions by positive patients.

(α, β)	EDADS $u_{\mathbb{E}}$	DS $u_{\mathbb{E}}$	EDADS expected F1	DS expected F1
$(10, 10)$	**4745.669953**	4739.439621	$\mathbf{0.84771 \pm 0.04650}$	0.83040 ± 0.06256
$(4, 16)$	**4654.436398**	4625.224034	$\mathbf{0.72210 \pm 0.11058}$	0.60889 ± 0.14199
$(16, 4)$	**4787.392586**	4786.883940	$\mathbf{0.93048 \pm 0.02706}$	0.92041 ± 0.02959
$(1, 1)$	**4690.061959**	4680.885183	$\mathbf{0.81030 \pm 0.18192}$	0.76795 ± 0.20792

Table 11: (NN-H) The expected objective value $u_{\mathbb{E}}$ (exact) and F1 score (averaged over 2500 simulations) when k=200 patients are selected. The number of positive patients staying follows a BetaBin distribution with parameters in the first column.

H.5. Risk-Averse DADS

EXPERIMENT ON (NB-A)

We select $k = 140$ data owners and consider the case of independent decisions with $s = .6$. It can be verified that smaller α leads to better balanced accuracy in the worse cases with more deletions.

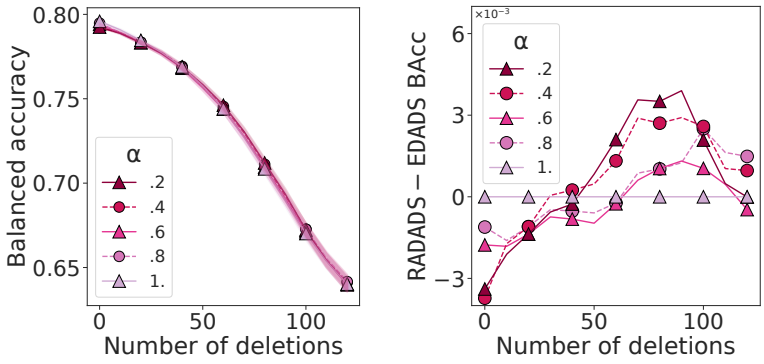


Figure 18: Graphs of mean validation set metrics obtained from the selections of $k = 140$ data owners by EDADS and RA_α−DADS (with varying α) (and additionally their differences) across 500 simulations with an increasing no. of deletions for (NB-A).

CVAR OBJECTIVE VALUE AND METRIC WITH DELETIONS

Across all experiments (see Tabs. 12 and 13), we observe that RA_α−DADS achieves a higher CVaR_α objective value and metric than EDADS.

α	RA$_\alpha$-DADS u_{CVaR_α}	EDADS u_{CVaR_α}	RA$_\alpha$-DADS CVaR$_\alpha$ F1	EDADS CVaR$_\alpha$ F1
0.2	$\mathbf{4756.3682 \pm 0.1186}$	4756.1135 ± 0.1641	$\mathbf{0.8411 \pm 0.0008}$	0.8404 ± 0.0007
0.4	$\mathbf{4760.0607 \pm 0.1092}$	4759.9601 ± 0.1242	$\mathbf{0.8571 \pm 0.0006}$	0.8566 ± 0.0009
0.6	$\mathbf{4762.6318 \pm 0.0967}$	4762.6228 ± 0.1038	$\mathbf{0.8695 \pm 0.0003}$	0.8678 ± 0.0004
0.8	$\mathbf{4764.9332 \pm 0.0959}$	4764.9111 ± 0.0952	0.8762 ± 0.0004	$\mathbf{0.8770 \pm 0.0004}$

Table 12: (NN-H) The CVaR$_\alpha$ objective value and F1 score (averaged over 50 runs, CVaR$_\alpha$ computed with 10000 samples) when k=240 patients are selected and the staying probabilities are $s_+ = .5, s_- = 1$.

α	RA$_\alpha$-DADS u_{CVaR_α}	EDADS u_{CVaR_α}	RA$_\alpha$-DADS CVaR$_\alpha$ acc	EDADS CVaR$_\alpha$ acc
0.2	$\mathbf{5.63728e11 \pm 7.13815e7}$	$5.63562e11 \pm 7.15768e7$	$\mathbf{0.7294 \pm 0.0008}$	0.7220 ± 0.0008
0.4	$\mathbf{5.65367e11 \pm 3.88493e7}$	$5.65303e11 \pm 4.38197e7$	$\mathbf{0.7411 \pm 0.0005}$	0.7365 ± 0.0005
0.6	$\mathbf{5.66309e11 \pm 2.85203e7}$	$5.66287e11 \pm 3.26483e7$	$\mathbf{0.7480 \pm 0.0004}$	0.7448 ± 0.0004
0.8	$\mathbf{5.67001e11 \pm 2.39820e7}$	$5.66994e11 \pm 2.60371e7$	$\mathbf{0.7523 \pm 0.0003}$	0.7507 ± 0.0003

Table 13: (NN-F) The CVaR$_\alpha$ objective value and accuracy score (averaged over 50 runs, CVaR$_\alpha$ computed with 10000 samples) when k=500 owners are selected. The number of owners staying present follows $\texttt{BetaBin}(500, 4, 2)$.

α	RA$_\alpha$-DADS u_{CVaR_α}	EDADS u_{CVaR_α}	RA$_\alpha$-DADS CVaR$_\alpha$ bal acc	EDADS CVaR$_\alpha$ bal acc
0.2	$\mathbf{2.17679e6 \pm 4.35455e2}$	$2.17663e6 \pm 4.55329e2$	$\mathbf{0.7162 \pm 0.0005}$	0.7131 ± 0.0006
0.4	$\mathbf{2.18866e6 \pm 3.34645e2}$	$2.18858e6 \pm 3.61646e2$	$\mathbf{0.7292 \pm 0.0004}$	0.7266 ± 0.0004
0.6	$\mathbf{2.19707e6 \pm 2.91878e2}$	$2.19704e6 \pm 3.06494e2$	0.7350 ± 0.0003	$\mathbf{0.7355 \pm 0.0003}$
0.8	$\mathbf{2.20447e6 \pm 2.52436e2}$	$2.20446e6 \pm 2.71833e2$	0.7425 ± 0.0003	$\mathbf{0.7427 \pm 0.0003}$

Table 14: (NB-A) The CVaR$_\alpha$ objective value and balanced accuracy score (averaged over 50 runs, CVaR$_\alpha$ computed with 10000 samples) when k=140 owners are selected. Each owner decides to stay present (independently) with probability $s = .6$.

2D VISUALIZATION

In Fig. 19, we mark the $k = 10$ points selected by RA$_\alpha$-DADS with different α. Note that $\alpha = 1$ corresponds to the EDADS objective. We set the staying probability s of every owner $= .5$ and u as the NN submodular function.

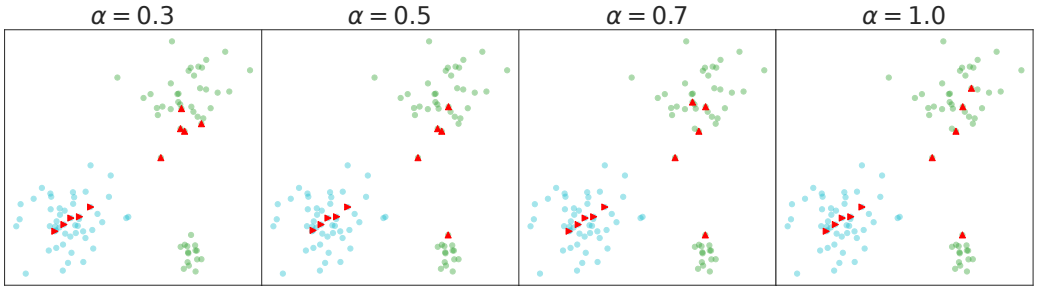


Figure 19: (NN-S) 10 data points (red) selected by RA$_\alpha$-DADS when varying α where $\alpha = 1$ corresponds to the EDADS objective. Each owner stays with probability $s = .5$.

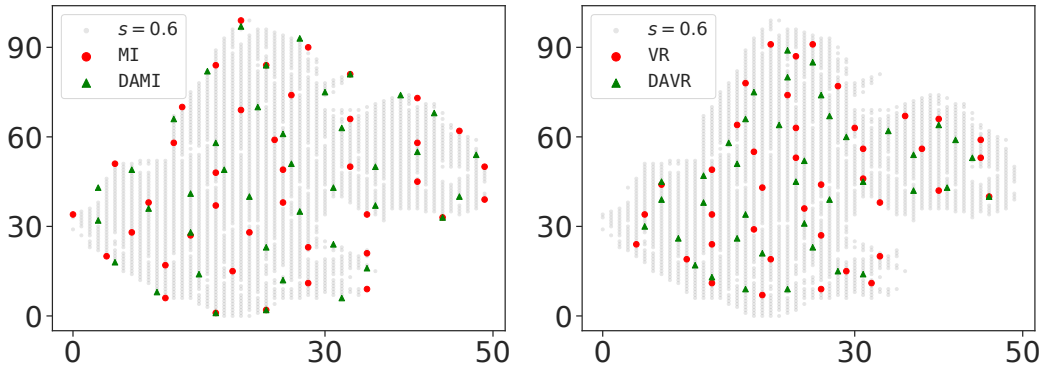
We observe that as α decreases (risk-aversion level increases), the data selected move slightly closer together (the difference is less drastic than changing the staying probability s). When $\alpha = 0.3$, RA$_\alpha$-DADS will forego the smaller green cluster to cover more of the larger green cluster.

H.6. Active Learning

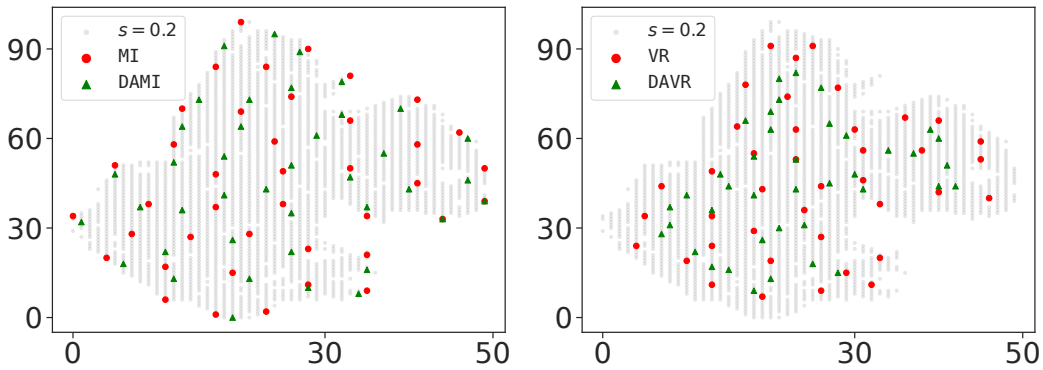
In this section, we consider the use of two active learning (AL) data utility functions: the mutual information and the variance reduction submodular functions (see App. C.2). To distinguish between both functions, we let MI and VR denote respective DS objectives and DAMI and DAVR denote the corresponding deletion-anticipative DADS objectives.

2D VISUALIZATION FOR (MI-S), (VR-S) WITH EQUAL STAYING PROBABILITIES FOR ALL INPUT REGIONS

- (MI vs. DAMI). In Figs. 20a&c, we do not observe a significant difference in the points selected by MI vs. DAMI for both choices of staying probabilities. This may be because MI seeks to maximize the diversity of the points (instead of the representativeness of the feasible set considered by the NN submodular function). Thus, when all points are equally likely to stay or be deleted, they affect the diversity metric equally.
- (VR vs. DAVR). In Figs. 20b&d, we do not observe a significant difference in the points selected by VR vs. DAVR for $s = .6$. However, when the staying probability is lower (for $s = .2$), DAVR seems to select regions closer to each other and the center. This is expected as the variance reduction function cares about the variance reduction at all input regions.



(a) $s = .6$, MI vs. DAMI (b) $s = .6$, VR vs. DAVR



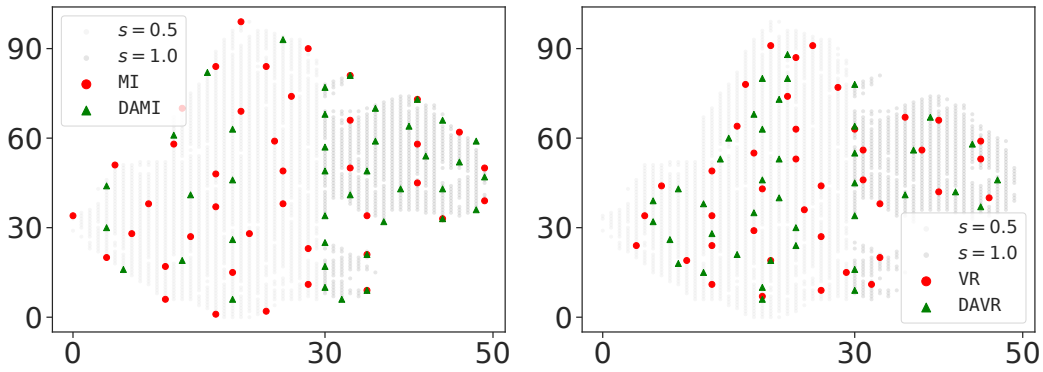
(a) $s = .2$, MI vs. DAMI (b) $s = .2$, VR vs. DAVR

Figure 20: 40 regions selected by MI or VR and DAMI or DAVR under different staying probability s.

Next, we will observe that the DADS objectives have a greater impact on the selection using these AL submodular functions when the probabilities *vary* across input regions.

2D VISUALIZATION FOR (MI-S), (VR-S) WITH UNEQUAL STAYING PROBABILITIES ACROSS ALL INPUT REGIONS

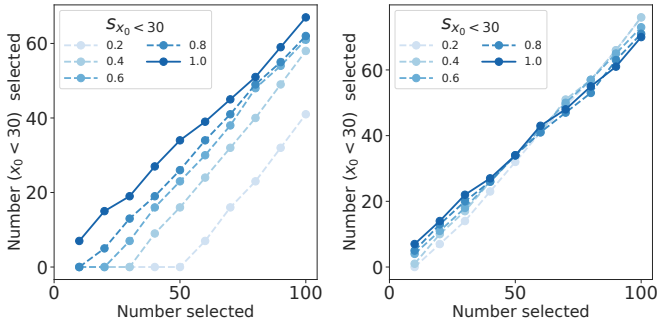
From the visualization in Fig. 21, we observe that both DAMI and DAVR prefer selecting owners with data in the right region. Both objectives select more regions along the border where $x_0 = 30$ to "cover" (reduce the output variance of) the region with $x_0 < 30$ with higher probability. Regions with $20 < x_0 < 30$ are almost never selected.



(a) MI vs. DAMI (b) VR vs. DAVR

Figure 21: 40 regions selected by MI or VR and DAMI or DAVR when the staying probabilities differ across regions. The region on the right with $x_0 \geq 30$ will not be deleted with certainty.

Number of Selected Points from Region with Lower Staying Probability in (MI-S) and (VR-S)



(a) DAMI (b) DAVR

Figure 22: Number of regions with $x_0 < 30$ (of staying probability s) selected vs. the total number of regions selected for the DA objective while varying s. Note that $s = 1$ corresponds to the non-anticipative MI/VR objective. The regions on the right will not be deleted with certainty.

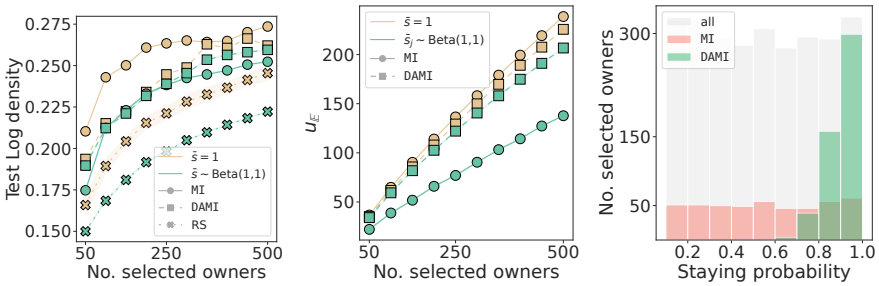
From Fig. 22, we observe when the staying probability of the left regions s decreases (lighter lines):

- DAMI will select fewer of the left regions. This may be because the mutual information function u^{MI} in App. C.2 only depends on S (the selected set that stays undeleted) and does not depend on the feasible set M. A region with higher staying probability will result in a larger staying subset S, contribute a > 1 eigenvalue in $\mathbf{I} + \gamma \mathbf{K}_S$) and increase the mutual information more frequently.
- DAVR will select fewer of the left regions initially (when the total selected is < 60) but more subsequently (more than VR from $s = 1$). The explanation is: Initially, regions with higher staying probability are more likely to contribute to variance reduction of neighbouring points. However, when those output variances are sufficiently low, DAVR selects inputs from the lower staying probability region as they contribute a larger variance reduction if they stay.

Experiment on (MI-F)

We randomly assign each owner j (of a datum) a staying probability s_j by sampling from the distribution $\mathtt{U}(0, 1)$. From Fig. 23a&b, we observe that when owners stay according to our probability model (green line), our DAMI objective outperforms MI on both the log predictive density and objective function value. However, we observe that the advantage comes with a trade-off — when there are no deletions (beige lines), MI has a higher validation set metric and performance.

In addition, from Fig. 23c, we observe that unlike MI which selects an equal number of owners with different staying probabilities, DAMI strongly prefers owners with higher staying probabilities. However, DAMI may select some owners with lower staying probability, e.g., 0.7, as they may contribute a larger gain in mutual information.



(a) (b) (c)

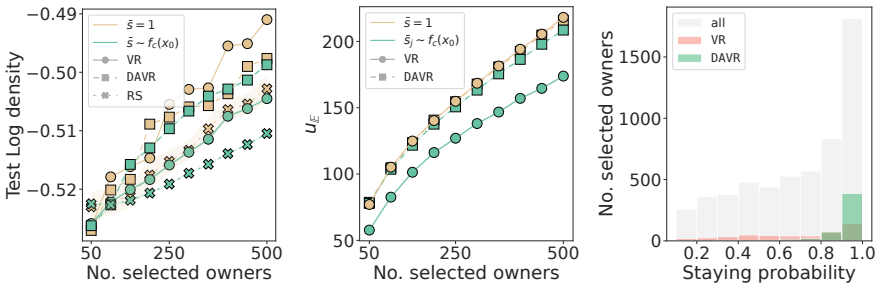
Figure 23: Graphs of (a) mean validation set metric scores (with $2\times$ standard error (shaded) across 2500 simulations), (b) mean objective function value (approximated by u_{avg}) with an increasing no. of data owners (horizontal axis) selected by MI and DAMI. Each data owner j's staying probability s_j is drawn the uniform distribution $\mathtt{U}(0,1)$ and the simulated staying probabilities $\bar{s}_j$ include 1 and s_j. (c) Histogram of the staying probability of the owners selected by MI and DAMI.

Experiment on (VR-C)

We assign each owner j (of a datum) a staying probability s_j dependent on the min-max scaled version of its median income (the 0th feature), x_0^j, as follows: $s_j = f_c(x_0) = .5 - .5\cos(4\pi x_0)$. This staying probability model encodes that owners with the smallest, largest or middle income are less likely to stay while owners with the lower and upper quartile incomes are more likely to stay.

From Fig. 24a&b, we observe that when owners stay according to our probability model (green line), our DAVR objective outperforms VR on both the log predictive density and objective function value. However, we observe that the advantage comes with a trade-off — when there are no deletions (beige lines), VR usually gives a higher validation set metric and performance.

In addition, from Fig. 24c, we observe that unlike VR which selects the same proportion of owners with different staying probabilities (as in M), DAVR more strongly prefers owners with higher staying probabilities. However, DAVR may select some owners with slightly lower staying probability as they may contribute a larger variance reduction.



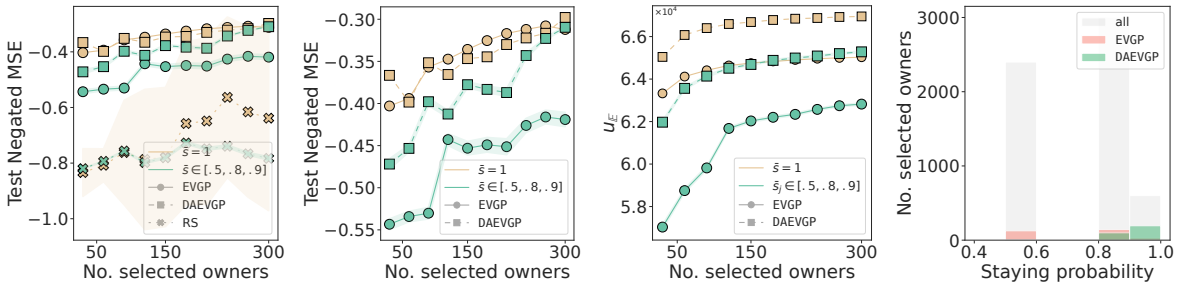
(a) (b) (c)

Figure 24: Graphs of (a) mean validation set metric scores (with $2\times$ standard error (shaded) across 2500 simulations), (b) mean objective function value (approximated by u_{avg}) with an increasing no. of data owners (horizontal axis) selected by VR and DAVR. Each data owner j's staying probability s_j is $f_c(x_j)$ and the simulated staying probabilities $\bar{s}_j$ include 1 and s_j. (c) Histogram of the staying probability of the owners selected by VR and DAVR.

EXPERIMENT ON (EVGP-C)

We assign 50%, 40% and 10% of the data owners with a staying probability of .8, .5 and .9, respectively. From Fig. 25a&b, we observe that when owners stay according to our probability model (green line), our DAEVGP objective outperforms EVGP on both the negated mean squared error (MSE) and objective function value. When there are no deletions (beige lines), EVGP may sometimes give a higher test negated MSE (i.e., lower MSE).

In addition, from Fig. 25c, we observe that DAEVGP more strongly prefers owners with higher staying probabilities. However, DAEVGP may select some owners with slightly lower staying probability (0.8), as they may contribute a larger variance reduction.



(a) (b) (c)

Figure 25: Graphs of (a) mean validation set metric scores (with 2× standard error (shaded) across 2000 simulations), (b) mean objective function value (approximated by u_{avg}) with an increasing no. of data owners (horizontal axis) selected by EVGP and DAEVGP. Each data owner j's staying probability s_j is either .5, .8 or .9 and the simulated staying probabilities $\bar{s}_j$ include 1 and s_j. (c) Histogram of the staying probability of the owners selected by EVGP and DAEVGP.

I. Other Questions

1. **When are the DADS objectives most useful over DS?**

   We believe that DADS objective is especially useful over DS when the number of points selected is moderate (i.e. not large, see Fig. 4).

   This may be because (i) for monotone submodular functions, more points may lead to a higher submodular curvature, thus the approximation guarantee of the greedy algorithm for DADS is worse and (ii) with more points, the accuracy or other validation set metric will saturate at the best possible value (thus, even DS is tolerant to a few deletions beyond the saturation point). Thus, DADS's higher data utility values may not translate to a higher validation set metric score.

2. **Are there parts of the work that are unique to ML applications?**

   The use cases in Sec. 3 are specifically for data acquisition problems. We also empirically demonstrate the results of using our DADS objectives on the validation set metric of ML models in Sec. 5.

   In Sec. 4.4, we highlighted that (1) in DS applications, it is best to only resample the inclusion of the newly added points as we can compute the marginal gain more efficiently incrementally and (2) some data utility set functions used in ML are a linear combination of multilinear functions and can be computed exactly and efficiently.

3. **How would the learner decide the staying/deletion probability in advance?**

   See Fig. 7 for an overview.

   As explained in Sec. 3.1, the learner can survey each owner j indirectly on its privacy preferences to predict the staying probability; query the data deletion history on trusted data sharing platforms; or sign binding contracts with data owners to enforce the staying probability.

   Moreover, in Sec. 3.2, we propose how the learner can set the probability distribution over staying subsets less precisely with less accurate knowledge. For example, the learner can decide that it only wants to tolerate z deletions or capture the uncertainty in the staying probability s_j using a Beta distribution whose prior depends on the observations/survey of a small set of data owners. This also allows some small errors in estimating the probabilities.

Lastly, the learner can instead define owner j's staying probability as the probability that j's data is credible. The probability corresponds to the probability that the learner and future data auditors (instead of owner j) would not request deletion/unlearning of j's data. This interpretation is useful when it is easy for the learner to identify how anomalous data is (e.g., by high Z-score for some features or low entropy of the image pixel values) but the learner can only accurately tell if the data is useful for predictions after training the ML model and evaluating the predictive performance. Thus, our DADS objectives can be used to balance between selecting data that are more credible vs. data with higher utility values.

4. **Why does Sec. 3.2 suggest that the learner decide p_K instead of p_M (over the feasible set)?**

   Firstly, if p_M is used, the learner has to set more probability parameters and more computation is needed to evaluate $\mathbb{E}_{p_M}$. Next, it may be less intuitive to decide the probability (or its relative concern) for deletions in M as it includes some unselected data. How should duplicate data be handled? If the learner selects a subset B, is the probability of B staying just $p_M(B)$?

   However, in App. E, we gave a use case example where the learner sets p_M instead.

5. **What are the limitations of a simple extension of existing DS methods, such as multiplying the marginal gain by the staying probability s_j?**

   There are three limitations:

   - The trivial approach would not work when each data owner's decision to stay present or delete their data depends on others' decisions (as in Sec. 4.2).
   - The learner has to optimize and choose between different formulas, e.g., multiplying by s_j or s_j^2.
   - The learner does not always prefer selecting the owner with a higher staying probability. Additionally, the learner needs to consider the expected marginal contribution (i.e., probabilistic value) (see Sec. 4.1) of each owner, which depends on the probability of deletions of *others* and the objective function.

     For example, from the histogram in Fig. 23c, we note that the deletion-anticipative objective did not select all the data of the highest staying probability.

     In Fig. 11, we observe that the learner selects more data of the lower staying probability for the nearest neighbor submodular function. However, in Fig. 22, we observe that the learner selects more data of the higher staying probability for the mutual information function.

6. **Can our method handle deletions that happen consecutively (separately) instead of happening together in a batch?**

   Yes, as mentioned in Sec. 2, the learner is subsequently free to use any unlearning technique and handle deletions consecutively or in batch mode.

   Proactively optimizing the DADS objectives in the active learning or data acquisition stage averts a future greater loss in model performance from deletions.

7. **How does our work relate to existing work on data valuation?**

   See Sec. G.

8. **Do we assume that the data utility function u is (monotone) submodular?**

   No, the aim of our work is to prove or construct set functions for the DADS objectives such that they preserve monotonicity, submodularity and weak submodularity (Chen et al., 2018a), etc. Our work realizes this aim by ensuring that expectation is over the same distribution (thus a positive weighted sum of functions that share the same property).

9. **Using sample average approximation (SAA) with n samples, the cost of evaluating the DADS objectives are n times the cost of evaluating the DS objective. Is the computation cost still feasible? Can the computation cost be reduced in practice?**

   Yes, the data utility functions for different sampled staying subsets can be evaluated in parallel.

   In particular, there are a few data utility functions (e.g., nearest neighbor submodular function (Wei et al., 2015), diversity function (Sim et al., 2020), variance-reduction function and EV-GP criterion (Krause et al., 2008; Hemachandra et al., 2023)) which are (i) *training-free* (i.e., do not require training of ML models) during data selection and (ii) lend themselves to efficient incremental updates that can be vectorized across samples. Thus, in our experiments, we can simultaneously compute the utility of many sampled staying subsets with matrix operations on GPU.