Heavy Sets with Applications to Interpretable Machine Learning Diagnostics

Dmitry Malioutov
Scarsdale

Sanjeeb Dash
IBM Research

Dennis Wei
IBM Research

Abstract

ML models take on a new life after deployment and raise a host of new challenges: data drift, model recalibration and monitoring. If performance erodes over time, engineers in charge may ask what changed – did the data distribution change, did the model get worse after retraining? We propose a flexible paradigm for answering a variety of model diagnosis questions by finding heaviest-weight interpretable regions, which we call heavy sets. We associate a local weight describing model mismatch at each datapoint, and find a simple region maximizing the sum (or average) of these weights. Specific choices of weights can find regions where two models differ the most, where a single model makes unusually many errors, or where two datasets have large differences in densities. The premise is that a region with overall elevated errors (weights) may discover statistically significant effects despite individual errors not standing out in the noise.

We focus on interpretable regions defined by sparse AND-rules (conjunctive rules using a small subset of available features). We first describe an exact integer programming (IP) formulation applicable to smaller datasets. As the exact IP is NP-hard, we develop a greedy coordinate-wise dynamic-programming based formulation. For smaller datasets the heuristic often comes close to the IP in objective, but it can scale to datasets with millions of examples and thousands of features. We also address statistical significance of the detected regions, taking care of multiple hypothesis testing and spatial dependence challenges that arise in model diagnostics. We evaluate our proposed approach both on synthetic data (with known ground-truth), and on well-known public ML datasets.

Proceedings of the 26[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

1 INTRODUCTION

Developing the best-performing model on a fixed dataset is the focus of much of academic ML research, but in practice it is only the first step in the life-cycle of an ML model. In this work we focus on model monitoring after the model is deployed or used on a new domain, and develop tools to help analysts gain interpretable insight into what changed (NMD+21; DB18; PWKC19). Our goal is to find simple interpretable regions in the feature space where either the data or the model changed the most. Finding a single datapoint with the largest error or prediction most different from a baseline is not difficult, but it may not carry much information, especially in noisy settings common with tabular or time-series data. Finding a larger region with overall elevated differences is often far more informative. In order to make such analysis insightful and actionable (and to help control overfitting), we need regions that are simple and interpretable – so in this work we focus on simple AND-rule regions using a small subset of features.

We model a variety of model-diagnostic problems using the same mathematical primitive: finding a simple region in feature space which has the highest weight. By appropriately defining weights at each datapoint we can address a variety of questions: finding regions where two models (regression or classification) disagree the most, regions where one model makes unusually many errors or where it is especially uncertain, regions with the largest changes in data-density or with pockets of strong correlation. The same framework could also be used to diagnose causal treatment effect models or model fairness. We allow arbitrary weights, including real-valued weights with mixed signs, positive weights, or binary weights. Finally, to identify the most salient region, we find the one with the highest sum (or average) of weights, optionally subject to cardinality constraints [1].

We first describe an integer programming (IP) formulation to find maximum-weight sets, applicable to smaller datasets. As the problem is NP-hard (BDT16), in order to address larger datasets we propose a greedy coordi-

[1]The formulation can be extended to more general knapsack-style settings, where one would like to optimize "reward-weights" subject to constraints on "resource-weights".

nate descent (CD) heuristic that optimally solves 1D (or 2D) subproblems for individual features via dynamic programming (DP). We use the 1-dimensional setting to illustrate interesting statistical challenges with heavy sets. A naive formulation is ineffective in accurate region localization due to long expected run-lengths of sums of iid random variables. We address this via cardinality constraints (equivalently, instance-wise penalties), and describe connections to classical cumulative-sum (CUSUM) statistics for change-point detection (Pag54).

Furthermore, we consider statistical significance of the detected region. In the context of model diagnostics, there are two major challenges. First, we face a severe case of multiple-hypothesis testing, as we are optimizing over a large family of regions. Second, ML models for natural phenomena tend to be slowly-varying and have strong spatial correlation in their errors or differences. A simple iid errors assumption would grossly overstate the number of independent observations and statistical significance. To mitigate these issues, we adapt permutation tests and a procedure motivated by classical clustered standard errors. We use clustered standard errors in a novel way: traditionally they are used with hand-picked categorical variables, i.e., the clusters are known a-priori, whereas we leverage recent research allowing unknown data-driven clusters (BCL20).

In terms of applicability, we do not expect to always be able to describe differences between two flexible complex models using simple interpretable regions. For example, differences between two powerful black-box models trained on the same dataset will likely be scattered and non-interpretable. In contrast, practical dataset shift can indeed often be interpretable (for example, a population ageing over time, or growing income levels), and these shifts will be apparent even after training complex classifiers on the data. We pursue this latter class of problems, where we believe and see evidence that rule-based interpretable regions can indeed be helpful in explaining model changes.

Related work. Our work has connections to several threads in the ML literature. In *group anomaly detection*, *scan statistics*, and *spatio-temporal hot-spot detection* (Nei09; SSMIN16), the goal is to find a group of datapoints that looks unusual compared to the background. Much of this work is spatio-temporal, but some look at higher dimensions with arbitrary categorical features. A growing body of work on ML model diagnostics (ZN16; PWKC19; PdAB21) finds poor performing regions in ML models but is predominantly focused on categorical features as well (and (PWKC19) is limited to single slices of features). In contrast, we focus on continuous features, and develop both exact and approximate solutions. Furthermore, some papers in group-anomaly detection discuss statistical significance of the detected regions (KHF18) but under the iid errors assumption. We address this limitation.

In (DB18; NMD+21) the authors use different techniques (variable importance and distillation into rule sets, respectively) to find differences between two models. (TCHL18) distills using generalized additive models (GAMs) to explain differences between a model and ground truth outcomes. Our formulation encompasses both of these problems as well as others described in Section 2. We also focus on finding the region with the greatest differences.

The interpretable rule-learning literature (ALSA+18; GR14; MVED17; EKG19; WDGG19; YHY+21) has considered IP formulations or relaxations for classification and regression problems. In particular, our maximum-sum-of-weights problem (but not the maximum-average) is similar to the rectangular maximum agreement subproblem of (EKG19), who also use a dynamic-programming relaxation to solve it but for the task of learning boosted regressors. Here we focus on a different problem with its own challenges, notably the statistical aspects of localization and significance that were not considered by (EKG19).

Outline. We introduce the heavy set formulation for model diagnostics in Section 2. We describe the exact IP in Section 3. We present the exact 1-dimensional DP-formulation and use it to develop approximate solutions for multi-dimensional problems in Section 4. Finally, we discuss statistical significance of detected regions in Section 5 and present experimental results in Section 6.

2 HEAVY SET PROBLEM FOR ML DIAGNOSTICS

We first motivate the abstract heavy set problem, the basic computational primitive that we use to address a variety of ML diagnostic applications. Heavy sets with sparse-rule regions are presented in Section 2.1.

Suppose that we have a collection of datapoints $x_t \in \mathbb{R}^N$, $t \in \mathcal{T} = \{1, ..., T\}$, that are associated with weights w_t. We consider general real-valued weights with mixed signs, but special cases of non-negative or binary weights are also of interest. These weights will encode various local metrics of fit / error / density that one may want to optimize, as we explain below. We also define a family $\mathcal{R}$ of simple regions $R \subset \mathbb{R}^N$: in the paper we focus on sparse AND-rule regions, see Section 2.1, but one could also imagine using alternative families such as k-nearest neighborhoods or radius-r balls. The goal is to find the heaviest-weight simple region from $\mathcal{R}$, optionally subject to region-size constraints (i.e., restricting the number of datapoints falling into R):

$$R^* = \arg\max_{R \in \mathcal{R}} \sum_{t | x_t \in R} w_t \,, \quad \text{such that} \quad |R| \leq K, \quad (1)$$

where we define $|R| = |\{t \mid x_t \in R\}|$, i.e. the number of points x_t that fall inside the region R. In addition to sum-

of-weights, we also consider the more challenging average-of-weights with upper and lower bounds on region size:

$$\arg\max_{R\in\mathcal{R}} \frac{1}{|R|}\sum_{t|x_t\in R} w_t\ , \quad K_{\min}\le |R| \le K_{\max}. \qquad (2)$$

Before delving into specific families of simple regions, we motivate applications of heavy sets. Suppose that we have datasets $D_A = \{x_t^A, y_t^A\}_{t=1}^{T_A}$ and $D_B = \{x_t^B, y_t^B\}_{t=1}^{T_B}$ where x's are the features and y's are (binary / multi-class / real-valued) labels. For example, dataset D_A could be collected pre-deployment, and D_B a few months post-deployment, over the same feature space. Suppose we also have models m_A trained on dataset D_A, and m_B on D_B. One may be interested in a variety of model diagnostic problems (we list a few here):

Model diagnostics using heavy sets:

- **Model changes.** Regions of D_B where predictions of model m_B deviate the most from m_A. For example, for regression problems, we could use $w_t = |m_B(x_t) - m_A(x_t)|^p$, $p = 1, 2$. For binary or multiclass classification we could sum the differences in class-probabilities. Signed weights $w_t = m_B(x_t) - m_A(x_t)$ can show where model B exceeds A.
- **High-error regions.** For example, $w_t = |m_A(x_t) - y_t|^p$, $p = 1, 2$ for regression, or a suitable loss for classification.
- **High-variance regions.** $w_t = m_A(x_t)^2$, using the max-avg formulation in (2) and de-meaning. Similarly, in lieu of a fitted model, weights could capture high-variance regions of the target variable in the data.
- **High/Low-correlation regions.** $w_t = m_A(x_t) * m_B(x_t)$, with max-avg formulation in (2). This is really a high inner product region, but, assuming both models were normalized, this could approximate inter-model correlations.
- **Density drift.** $w_t = \begin{cases} -\frac{1}{T_A} & \text{for } t \text{ in } D_A \\ \frac{1}{T_B} & \text{for } t \text{ in } D_B. \end{cases}$ Here we combine the two data-sets $D = D_A \cup D_B$.
- Applications in *causal inference*, e.g. a region with elevated individual treatment effects (or model differences), applications to fairness / bias.

2.1 Heavy-weight sparse AND-rules

Now we focus on heavy sets with simple regions defined in terms of sparse AND rules. We consider continuous (and ordinal) features first, and leave categorical features to Section 4.5. Suppose that we have N features $x = [x_{[1]}, ...x_{[N]}]$. The region is defined based on a small subset of features $\mathcal{I} \subset [1, ..., N]$, and a pair of lower and upper bounds for each variable: AND-rule $\triangleq \{(i, m_i, M_i)\}$ for $i \in \mathcal{I}$. We allow the lower bound m_i to be a real number or $-\infty$, and the upper bound M_i a real or $+\infty$, i.e., the regions could be one or two-sided.

$$\text{AND-rule-region} = \{x \mid m_i \le x_{[i]} \le M_i\ ,\ i \in \mathcal{I}\} \qquad (3)$$

As a motivating example, a region of this form in a medical context might look like: BodyMassIndex ≥ 30 and SystolicBloodPressure ≥ 120 and $20 \le$ age ≤ 40, where the three features are a subset of all available features. Such rule-based regions are considered highly interpretable[2]. The number of possible bounds for a single continuous feature can be assumed to be $O(T)$, as both m_i and M_i can be limited to distinct values of $x_{[i]}$ seen in the dataset. One can further limit the set of bounds or thresholds by partitioning the range of $x_{[i]}$ into disjoint intervals (e.g. based on 5%-quantile intervals). We assume that $B_i = (m_1, \ldots, m_{|B_i|})$ is the ordered set of thresholds (or bin boundaries) for feature $x_{[i]}$ with $-\infty < m_1 < m_2 < \cdots < m_{|B_i|} < \infty$. Having defined the regions, we'd like to find a sparse AND-rule region which includes datapoints with the most weight. Next, in Section 3, we define an exact integer programming formulation appropriate for small datasets and present a greedy scalable heuristic in Section 4.4.

3 HEAVY AND-RULES: INTEGER PROGRAMMING FORMULATION

We start off with a basic mixed-integer program (MIP) to obtain a max-weight sparse AND-rule. This MIP has binary variables z_l that specify the AND-rule, and other binary variables α_t specifying which points satisfy the AND-rule. We assume that the upper and lower bounds in the AND-rule in (3) come from the list B_i. We will create a model with $2|B_i|$ variables per feature $x_{[i]}$. Half of the variables (call this set of variables U_i) indicate whether the condition $x_{[i]} < m_k$ for some $m_k \in B_i$ is a part of the AND-rule. The other half indicate whether $x_{[i]} > m_k$ for some $m_k \in B_i$ (call this set of variables L_i). Let J stand for the set of indices of feature-threshold pairs. Suppose $l \in J$, and let $z_l = 1$ correspond to a condition of the form $x_{[i]} < m_k$ or $x_{[i]} > m_k$. Let a_{tl} be a 0-1 constant indicating whether the tth data point satisfies the condition encoded by z_l. Binary variables α_t indicate whether point

[2] While we don't pursue it here, more complex interpretable regions can be constructed by combining such AND-regions (i.e. a DNF formula, a rule-list or a weighted combination (MVED17)).

t satisfies the AND-rule. The basic model is as follows:

$$\max \sum_{t\in\mathcal{T}} w_t\alpha_t \tag{4}$$

$$\text{s.t. } \alpha_t + z_j \le 1, \qquad \forall t\in\mathcal{T}, j\in J : a_{tj}=0 \tag{5}$$

$$\alpha_t + \sum_{j:a_{tj}=0} z_j \ge 1 \qquad \forall t\in\mathcal{T} \tag{6}$$

$$\sum_{j\in J} z_j \ge 1 \tag{7}$$

$$\alpha_t \in \{0,1\} \qquad \forall t\in\mathcal{T} \tag{8}$$

$$z_j \in \{0,1\} \qquad \forall j\in J \tag{9}$$

Solving this basic model gives the maximum weight AND-rule that is non-empty (i.e., the rule is defined by at least one bound constraint). To see this consider an optimal solution $\bar{z}$ of this model. By constraint (5) all points x_t that do not satisfy a selected condition indicated by $\bar{z}_j = 1$ (i.e., points for which $a_{tj} = 0$) must have $\alpha_t = 0$. On the other hand, constraints (6) ensure that if all the conditions violated by a point x_t are not present in the AND-rule (so x_t satisfies the resulting AND-rule), then $\alpha_t = 1$ (and the weight of the point is added in the objective). The model above can be strengthened by introducing additional constraints. We use this strengthened model in our experiments, and discuss it in Appendix A.

We now give a formulation to compute the maximum average weight AND-rule, assuming the average is positive and at least σT points satisfy the AND-rule where $0 < \sigma < 1$. It consists of all constraints from the previous model, but has a different objective and some additional constraints that enable the computation of the average. Let $w_{\max}$ be the maximum weight of a data point (it is an upper bound on the maximum average weight). We use a variable β that represents the average weight of the chosen points. In that case $\beta = \sum_{t\in\mathcal{T}} w_t\alpha_t / \sum_{t\in\mathcal{T}} \alpha_t$. This is a nonlinear constraint but can be linearized as follows. Assume we have a variable β_t to represent the product $\beta\alpha_t$. Then the formulation we use is:

$$\max\ \beta \tag{10}$$

$$\beta_t \le \beta \qquad \forall t \tag{11}$$

$$\beta_t \le w_{\max}\alpha_t \qquad \forall t \tag{12}$$

$$\beta + w_{\max}\alpha_t \le \beta_t + w_{\max} \qquad \forall t \tag{13}$$

$$\beta_t \ge 0 \qquad \forall t \tag{14}$$

$$\beta \ge 0 \tag{15}$$

$$\sum_{t\in\mathcal{T}} w_t\alpha_t = \sum_{t\in\mathcal{T}} \beta_t \tag{16}$$

$$\sum_{t\in\mathcal{T}} \alpha_t \ge \sigma T \tag{17}$$

$$\text{Constraints}(5)-(9)$$

When $\alpha_t = 0$, the constraints (12) and (14) force β_t to be 0; the constraints (11) and (13) are redundant in this case. On the other hand, when $\alpha_t = 1$, (11) and (13) together force β_t to equal β, and the constraints (12) and (14) are redundant. Thus β_t is constrained to equal $\beta\alpha_t$ when α_t is 0 or 1. Equation (16) enforces the condition $\beta\sum_{i\in\mathcal{T}}\alpha_t = \sum_{i\in\mathcal{T}} w_t\alpha_t$ when all α_t variables are set to 0/1 values. Finally, constraint (17) enforces the condition that at least a fraction σ of all points are required to satisfy the AND-rule. Without this constraint, an AND-rule that only includes a highest-weight point may be chosen. This model can be strengthened by constraints (27)-(33) as in the case of the max-sum model described earlier.

4 EFFICIENT SOLUTION VIA DP-HEURISTIC

The exact IP formulations in Section 3 can be used to find heavy sets in smaller datasets (with at most a few thousand datapoints and dozens of features, and coarse quantization). We now pursue an efficient approximate solution that scales to millions of datapoints and thousands of features and does not require quantization. First, in Section 4.1 we focus on the 1-dimensional setting, and describe an exact efficient dynamic-programming formulation. We point out and address statistical issues in using it for heavy sets in Sections 4.2 and 4.3. Finally we propose a heuristic based on exactly solving 1D subproblems in Section 4.4.

4.1 Optimal heavy set solution in 1D

In 1D, the heavy set problems in (1) and (2) can be efficiently solved in $O(T)$.[3] Consider the unconstrained 1D max-sum heavy set problem first:

$$w* = \max_{t_m, t_M} \sum_{t=t_m}^{t_M} w_t \quad (P1) \tag{18}$$

Here, we assume without loss of generality that points are indexed (i.e., ordered) on $t = [1, .., T]$, and have weights w_t. This version of the problem is known as maximum (contiguous) sub-array sum, and can be solved in $O(T)$ by an elegant dynamic-programming formulation proposed by Jay Kadane (Ben84). A simplified algorithm is listed below for convenience, with array with elements $A[i]$:

```
max_sofar=0; max_here = 0;
for i in [0,..,T-1]:
    max_here  = max(max_here + A[i], 0)
    max_sofar = max(max_sofar, max_here)
```

As we describe in Section 4.2, the unconstrained formulation in (P1) has undesirable statistical properties (poor localization), and in particular, it becomes meaningless when the weights are non-negative: the trivial full-array solution

[3] Or $O(T_U)$, the number of unique points (after binning).

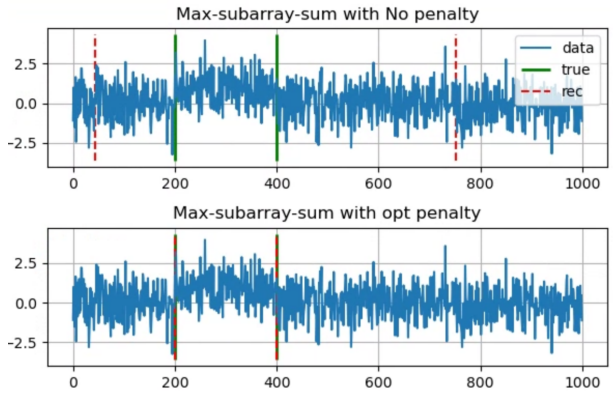

Figure 1: Max-subarray localization of planted regions. Green vertical lines show the true planted region, and dashed red lines the estimated region. (top) no-penalty (bottom) mean/2 penalty.

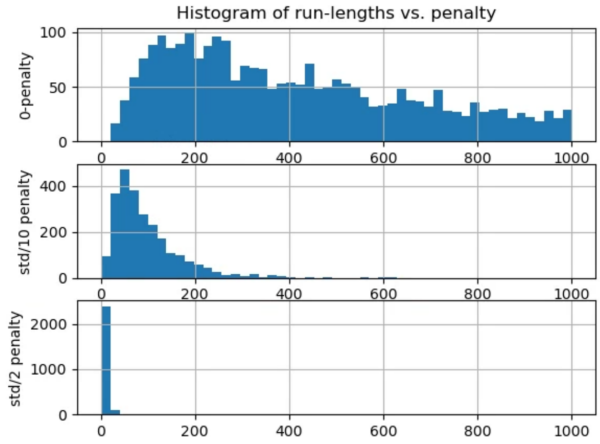

Figure 2: Run-lengths of max subarray-sum regions under null-hypothesis (0-mean), $T = 1000$. (top) 0-penalty (middle) std/10 penalty (bottom) std/2 penalty.

always achieves the maximal weight. Instead we consider (P1) subject to region-size constraints $|R| \leq K$:

$$w^*(K) = \max_{t_m, t_M} \sum_{t=t_m}^{t_M} w_t, \;\; t_M - t_m + 1 \leq K \quad (P2) \quad (19)$$

There exists an $O(T)$ efficient algorithm for length-constrained max subarray sum, (CL05), but in order to generalize it to higher-dimensions we instead use a Lagrangean formulation. Adding a Lagrangean penalty on the region-size constraint and simplifying, we have:

$$w^*(\delta) = \max_{t_m, t_M} \sum_{t=t_m}^{t_M} (w_t - \delta) \qquad (P3) \qquad (20)$$

To find a solution with desired cardinality, we use bisection search to find min δ that satisfies $t_M - t_m + 1 \leq K$. Within each bisection iteration we have to solve the unconstrained problem (P1) with modified weights $\tilde{w}_t = w_t - \delta$. We note that unlike convex-optimization problems (where under some technical conditions strong duality holds), for this discrete optimization, the set of solutions $\{w^*(K)\}$ is not equivalent to $\{w^*(\delta)\}$. We discuss in Appendix C that $\{w^*(\delta)\}$ includes those solutions from $\{w^*(K)\}$ that lie at the corner points of the convex-hull of all solutions in the (weight, cardinality) space. Despite this "loss of resolution" we rely on formulation (P3), as it can be conveniently extended to the multi-dimensional setting in Section 4.4. One could argue that the points on the convex hull have a particularly good trade-off of region-weight vs cardinality, but ideally we would like to be able to reach all the solutions. This is the subject of ongoing research on strengthening heavy set approximations.

Max-average (max-density) version in 1D An efficient $O(T)$ solution is also available for the 1D max-average problem in (2) with lower and upper bounds on region size, and allowing more general knapsack constraints. It was proposed under the name "maximum-density segments" in the biological sequence analysis literature (CL05).

4.2 Run-lengths of 1D heavy sets without penalties

To gain intuition into max-subarray-sum for detecting planted heavy sets, we consider the 1D case and take w_t initially to be an i.i.d. zero-mean Gaussian $\mathcal{N}(0, 1)$ vector, and plant a small region where we increase the mean by $\mu > 0$. Naively, one can attempt to recover the region by solving the maximum-weight subarray problem. However, even with μ visibly standing out in the noise, the recovered regions are off, see Figure 1 (top). The reason is that even under the null hypothesis of zero-mean, the estimated heavy sets have expected length of $O(T)$, as we can see in Figure 2(top). Hence, accurate localization of the planted region is essentially impossible, as errors are of the order of the length of the array! By adding an appropriate small penalty δ, with weights $w_t - \delta$ in (P3), these run-lengths dramatically shrink, and now allow accurate localization of the planted region, Figure 1(bottom) and 2(middle,bottom). We describe a statistical interpretation of this penalty next.

4.3 Statistical interpretation

We show that problem (20) can be alternatively derived from a statistical localization perspective in the Gaussian setting of the preceding example. This provides the statistical interpretation of the penalty δ mentioned above.

We assume that $w_1, \ldots, w_T$ are independent Gaussian random variables. For an unknown interval $t_m, \ldots, t_M$, the mean is $\mu > 0$, and elsewhere the mean is zero. The variance is the same σ^2 throughout. The joint probability den-

sity of $w_1, \ldots, w_T$ is therefore

$$f(w_1, \ldots, w_T) = \prod_{t=t_m}^{t_M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(w_t-\mu)^2}{2\sigma^2}\right) \times \prod_{t \notin \{t_m, \ldots, t_M\}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{w_t^2}{2\sigma^2}\right). \quad (21)$$

Given a realization of $(w_1, \ldots, w_T) = w$, we localize the interval by estimating t_m, t_M via maximum likelihood. From (21), the log-likelihood can be written as

$$\ell(t_m, t_M \,|\, w) = \frac{\mu}{\sigma^2} \sum_{t=t_m}^{t_M} \left(w_t - \frac{\mu}{2}\right) - \frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T} w_t^2. \quad (22)$$

We observe that the second line in (22) does not depend on t_m, t_M and can thus be dropped from the maximization. The remaining quantity in the first line is then proportional to the objective function in (20), if we identify δ with $\mu/2$.

We can now interpret problems (P2) (19) and (P3) (20) as representing different forms of prior knowledge. If we have prior knowledge that the interval has mean μ or greater, or approximately so, then this provides a setting of δ as $\mu/2$ in (20). Note that knowledge of σ^2 is not necessary because it appears only in the constant of proportionality and in the second line in (22). If instead we have prior knowledge (or desire) to have the detected region be of size K, then we can search for δ using bisection search to get a solution with cardinality close to the desired K in (19).

The above derivation has close parallels to CUSUM statistics for online change-point detection in process control (Pag54) (tutorial (Gra14) is helpful in seeing this). The problems are different however: CUSUM is used for *online detection* (i.e., from observations arriving sequentially) of a *single*[4] change-point, and if a change is detected, it is also localized. In contrast, our problem is *offline*, only addresses localization, and is *two-sided* with two boundaries.

4.4 Greedy approach for N-dim heaviest AND-rule

As the exact IP formulation in N dimensions in Section 3 is limited to small problems, we pursue an approximate coordinate descent solution. We use Kadane's 1D dynamic-programming algorithm as a subroutine.

The high level idea is to build up our subset $\mathcal{I}$ of active features (recall (3)) one at a time, where we compute the marginal gain γ_i of including each feature i, and add the feature that maximizes the marginal gain. Marginal gain is defined as the increase in the weight of the region after we add the feature. This can be viewed as a Gauss-Southwell version of coordinate descent (picking coordinate with the largest gain). At each step, we decide whether including the new feature substantially improves the weight, and if not we stop. Similar to what we have done in the 1D version, we use the Lagrangean formulation of the heavy set problem with a Lagrange multiplier λ to convert the inequality constraint: $R^* = \arg\max_{R\in\mathcal{R}} \sum_{t|x_t\in R} w_t - \lambda|R|$. This is equivalent to maximizing:

$$R^* = \arg\max_{R\in\mathcal{R}} \sum_{t|x_t\in R} (w_t - \lambda). \quad (23)$$

For a fixed λ, we run Algorithm 1, with weights $w_t - \lambda$. Let $J(\lambda)$ be the max value corresponding to (23). To enforce cardinality constraints we use a binary search over λ to find

$$\max_{\lambda} J(\lambda) \text{ s.t. } |R| \le K. \quad (24)$$

We distinguish two versions of the algorithm: (i) pure-greedy never re-visits a coordinate after selecting it once. (ii) Coordinate-descent (CD), allows revisiting already-selected coordinates and updating their bounds (m_i, M_i). If we impose a budget on the number of active coordinates, CD is only allowed to revisit already selected coordinates after reaching the budget. Pure-greedy (non-revisit) has finite termination, while the CD solution (revisit) tends to furnish better approximations to the IP.

Algorithm 1: Unconstrained multi-dim heavy set.

```
// λ set outside (bisection search)
function CoordDescHeavySet(X, w_t)
    INIT. Active set I = {}.
    for i ← 0 to max_steps do
        marg. gain γ_i ← solve (P3) w. x^t_[i], wts w_t − λ.
        i* = arg max γ_i. I = I ∪ i*, update m_{i*}, M_{i*}
```

Stopping criteria. To keep regions interpretable, we may decide to stop adding new features to the AND-rule after reaching a predefined budget (say 3 or 4) of active features. Alternatively, one could use the t-test to check if the weight increase due to the recent update is statistically significant. A naive definition of region-weight t-statistics based on an i.i.d assumption is easy to use, but may be overly crude for ML diagnostics. We discuss a refined solution in Section 5.

4.5 AND-rules with categorical features

Our focus in the paper is on continuous (or ordinal) features. However, unordered categorical features (with small number of categories) can be incorporated using a simple extension: using an arbitrary subset instead of the contiguous sub-array. There is an extensive literature on group

[4] Multiple changes can be detected by resetting the algorithm.

anomaly detection with categorical variables, especially the subset-scanning literature (SSMIN16; ZN16) which allows fast scanning with a general class of linear-time-subset-scan (LTSS) objective functions. Also, more complex schemes that include graph or topological category priors are possible, or regularization that prefers "simple" subsets of categories, but we do not pursue them in this paper. In the context of the Lagrangean formulation in (20), our simple solution selects exactly those categories with positive aggregate weights $w_t - \lambda$. We treat ordinal variables (e.g. binned age) as continuous.

5 REGION STATISTICAL SIGNIFICANCE

We now discuss statistical significance of regions found using heavy sets (either IP in Section 3 or DP-heuristic in Section 4.4). In the case of a fixed region with i.i.d. points, the classical t-test can decide if the region mean is higher than the background. However, since our regions are the result of optimization, their weights will generally be higher than background, and we face a multiple-hypothesis testing bias. Furthermore, the i.i.d. assumption is generally a poor model for weights arising from ML diagnostics: ignoring spatial correlation leads to over-counting the effective number of independent observations and hence significance.

We suggest the permutation test for the first bias. We randomly permute weights w_t (dispersing the alleged high-weight region), find an optimal region with permuted weights, and repeat. The distribution of permuted region weights captures the null-hypothesis. If the weight of the detected region (w.o. permutation) is indeed anomalous it should stand well outside the null-hypothesis distribution.

For the spatial-correlation bias we suggest applying a version of clustered standard errors. Classical clustered standard errors (AAIW17) assume known clusters, where the errors are correlated within the cluster, but independent across clusters. More recent research addresses the case of unknown clusters (BCL20). We simply use a generic clustering algorithm (e.g. k-means or spectral clustering) to define local clusters whose average size roughly matches the spatial auto-correlation length of the data[5], and apply clustered standard errors. This provides a reasonable first-order correction to the iid assumption. We illustrate the approach for statistical significance in experiments in Section 6.

6 EXPERIMENTAL RESULTS

We now evaluate the proposed heavy set approach experimentally on both simulated examples with ground-truth, and well-known ML datasets.

[5]For multi-dimensional data we can measure empirical auto-correlation as a function of the number k of nearest neighbors.

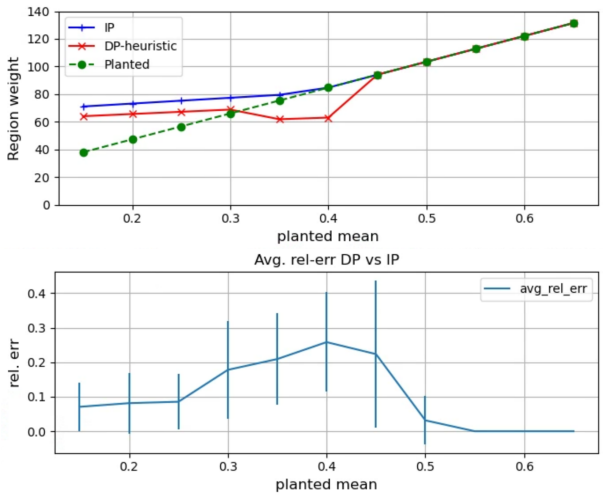

Figure 3: IP vs DP-heuristic. (top) weight of planted and recovered regions (y-axis) vs. the planted mean μ (x-axis). Below some noise floor the planted region becomes submerged in noise, and suboptimal in terms of weight. DP-heuristic comes close to IP, and matches it at higher levels of planted mean. (bottom) Relative error of DP-heuristic w.r.t. IP vs planted mean μ. Averaged over 5 trials.

6.1 Simulated data: planted heavy sets

Efficacy of the DP-heuristic vs IP First, we compare the DP-heuristic solution in Section 4.4 to the exact IP from Section 3. We use simulated data with known ground-truth: we generate a 10-dimensional iid Gaussian $\mathcal{N}(0,1)$ dataset with 1500 datapoints, and plant a heavy set region R^* with mean $\mu > 0$ that uses 3 of the features and covers around 10% of the data. We compare weights of estimated regions $\hat{R}$ found by IP and DP-heuristic (we use the CD version for experiments) as a function of the planted mean. Results appear in Figure 3(top). We see that IP recovers the planted region at higher levels of μ, while for small μ, the planted region is submerged in noise, and IP finds a higher-weight solution. Also, the DP-heuristic provides a reasonable approximation to the IP over the entire range of planted means and is able to match it exactly for larger means. In Figure 3(bottom) we plot average relative error (IP-DP)/IP over 5 trials.

Regression tree baseline A simple natural baseline to find interpretable rule-based heavy-sets is to use a regression tree to predict the weights, and then to find a leaf (or a sub-tree) that has the heaviest aggregate weight. Note, however, that the tree is trained with the goal of accurately predicting the weights, and not optimizing the weight of the subset. The latter is done simply as a post-processing step. In contrast, our DP-based solution does attempt to go after the correct objective function. To evaluate the tree heuristic we train tree regressors with various depths, and find the

Table 1: Max-sum vs max-avg IPs

planted μ:	0.2	0.3	0.4	0.5	0.6
mx-sum #pts	139	139	163	163	163
mx-avg #pts	75	78	78	79	79
mx-sum: avg-wt	0.378	0.400	0.423	0.523	0.623
mx-avg: avg-wt	0.523	0.545	0.569	0.647	0.747

leaf that maximizes the sum of the weights over all depths. While this approach does provide a useful solution, its average relative error is significantly higher, at 21%, whereas our DP-heuristic is more accurate with an 8% error. The average is over trials and planted means as in Figure 3(bottom). The under-performance of the regression tree can likely be attributed to the mismatch of its objective functions and the heavy-set problems.

Max-sum vs max-average IPs Next, we briefly compare max-sum vs max-average IP formulations for a planted region over a range of planted means μ. We use $|R| \leq 163$ (size of the ground-truth planted region) for max-sum and $|R| \geq 75$ (i.e., $\geq 5\%$ of points) for max-average (recall that max-avg needs a lower bound to avoid a trivial solution). R^* with size 163 is feasible for both. The results appear in Table 1. We report the average-weight for the regions found by both formulations (it is not optimized by max-sum IP). Max-sum tends to find solutions with size close to the upper bound, while max-average is closer to the lower bound. While the planted heavy set is a feasible solution, max-average instead selects a smaller region with higher average weight.

Region and feature detection results. Next, we leave IP behind to focus on larger problems, and study the performance of the DP-heuristic in recovering planted regions and identifying the active features. We generate T i.i.d. datapoints x_t uniformly over $\mathbb{R}^N$, and generate weights w_t as i.i.d. Gaussians. We then pick an AND-rule involving K features, and assign lower and upper bounds at random, making sure that the bounds cover between 25 and 75 % of the range of the feature. We consider how well we are able to identify these planted regions in terms of feature recovery (fraction of true active features identified) and region overlap, i.e. $\frac{\hat{R} \cap R^*}{\hat{R} \cup R^*}$. In Figure 4 we plot these two metrics as a function of number of samples T, the planted mean μ and the number of active variables K. With higher T and μ the problem becomes easier, but with more active variables it gets harder. The variables that are not varying are fixed at $T = 5000$, $N = 10$, $\mu = 1.25$, $K = 3$ and planted regions include roughly 5% of points (250 points).

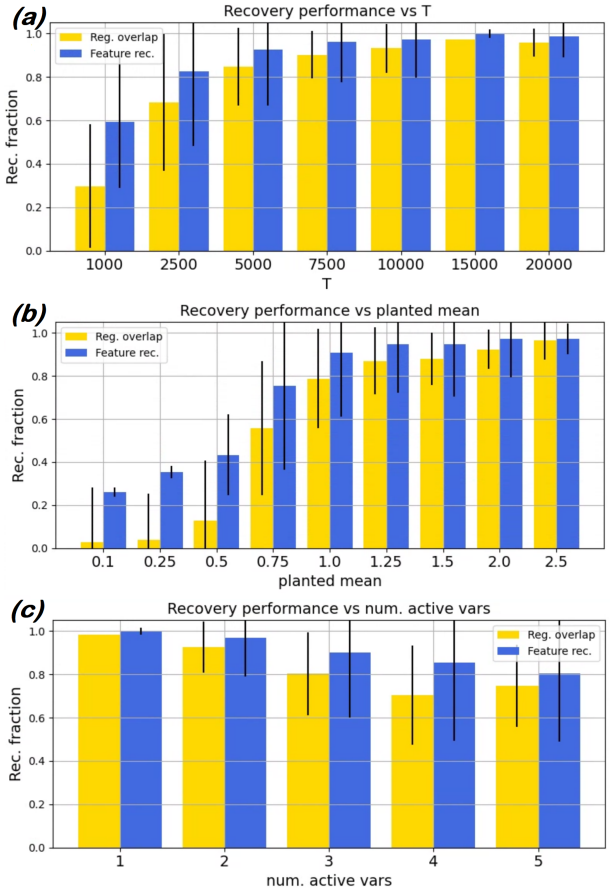

Figure 4: Planted region recovery: feature recovery and region overlap. (a) vs T (b) vs mean μ (c) vs # active vars.

6.2 Model diagnostics examples

Folktables. We use the folktables dataset (DHMS21), a newer version of the widely-used Adult census dataset refreshed with data from years 2014 and 2018, split by state, and extended with additional targets. It is particularly well-suited for studying data drift and model fairness questions. Pre-processing details are in Appendix D. First we use heavy sets to identify an interpretable sub-population which has unusually high levels of the target variable (INCOME $\geq$ 50000\$). Here we use heavy sets directly on data without a trained model. Our weights are simply defined as the binary target values. Out of 1.7M observations across all the states for 2018 we identify a sub-population of roughly 10% in size, where over 84.2% of target values are positive vs 39% in the baseline. The 3-feature AND-rule identifying this sub-population is:[6]

```
SCHL : Bach. degree <= x <= Doct. degree
WKHP : 40-50 <= x <= 80-90
AGE  : 30-50 <= x <= 70+
```

Next we train separate logistic-regression models for MA

[6] SCHL=education level, WKHP=weekly work hours.

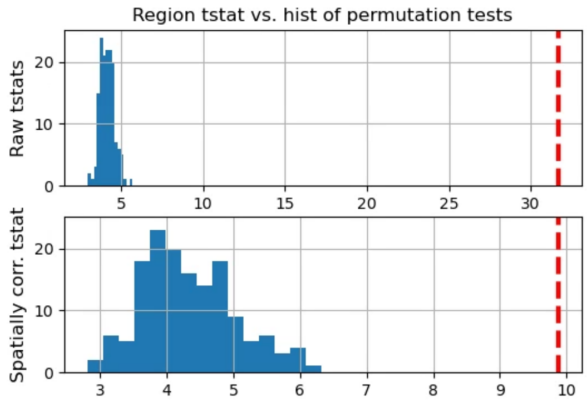

Figure 5: T-stat of discovered region (dashed red line) vs. histogram of permutation-test t-stats. (top) Raw (bottom) spatially corrected.

and TX, and compare them on withheld data from TX. We use a 50-50 train / test split. We take the weights to be the differences in predicted probabilities of the positive class between the two models. We find a region with 13.7% of datapoints where the average predicted probability difference between the two models is 17.2 %, whereas the baseline is 10.2%. The AND-region identifying this set is:

```
COW: [Loc. gov empl., NO-pay, family biz,
     State gov empl., private for-profit,
     private non-profit]
WKHP: 40-50 <= x <= 70-80
SCHL: <=8 Grades <= x <= Assoc. degree
```

We see that the differences found between the two models are still readily interpretable, although a bit more complex than that of high-target-value regions for a single data-set.

In Appendix E, we present examples of heavy AND-regions on two additional datasets.

Region statistical significance. Next we consider the UCI electricity dataset, with 38.4k observations, 8 features, and a binary target. We train logistic and RF models on a training set (50%) and compare them on the test set. We define weights w_t as differences in predicted class-1 probabilities. Using heavy sets we find a region (AND-rule with 3 variables) covering 2760 out of 19237 test datapoints, where the mean difference exceeds the baseline by 0.22 (0.524 for $\hat{R}$ vs. 0.306 for the full test-set). The baseline is positive since RF performs better than logistic. Naive t-stat calculation shows overwhelming statistical significance, with t-stat 32.67 for the region. However, this is largely due to ignoring spatial correlation, which inflates the number of independent observations. We plot in Figure 5 the naive and spatially corrected t-stats for the estimated region superimposed on a null-hypothesis histogram of t-stats of regions found after randomly permuting the weights. After spatial correction (see Section 5) with avg. cluster-size 50 roughly equal to the correlation length, the region is still statistically significant but with a more modest t-stat= 9.87. The [1,99]%-percentile range of the permutation test t-stats is $[3.04, 5.96]$. Note that the permutation test t-stats are little impacted by spatial correction since by design their weights lack spatial correlation.

7 CONCLUSIONS AND FUTURE WORK

We presented a flexible framework to investigate model change and other ML model diagnostic questions based on finding heaviest-weight interpretable regions in the data, which we call *heavy sets*. The weights at each datapoint characterize model error or mismatch, and we aim to find interpretable regions in the data (characterized by AND-rules) with the heaviest sum or average of these weights. We propose both exact (integer programming) and approximate (dynamic-programming) approaches to search for the maximal AND-rule regions, and discuss the evaluation of statistical significance of the discovered regions.

Future work. In the paper we focused on interpretable heavy-sets described by sparse AND-rules. Alternative definitions of heavy-sets are also of interest: for example heavy-sets based on K-nearest-neighbors or radius-r balls around datapoints, or heavy sets based on decision trees can also be readily interpreted. Furthermore, one could look for heavy-weight neighborhoods in a neighborhood graph (perhaps using graph neural nets). We are also interested in theoretical guarantees for the proposed approach. For example, in the rule-learning context, (YHY+21) developed performance guarantees using results from submodular optimization. Related analysis may offer guarantees on performance of greedy heuristics to find heavy-sets when all the weights are non-negative.

Acknowledgements

We would like to thank Skyler Speakman, Adebayo Ayomide Oshingbesan, and Kush Varshney for helpful discussions. We thank Tanya Akumu for providing the preprocessed folktables dataset https://github.com/tanya-akumu2/folktables_scan.

References

[AAIW17] A. Abadie, S. Athey, G.W. Imbens, and J.. Wooldridge. When should you adjust standard errors for clustering? *National Bureau of Economic Research. w24003*, 2017.

[ALSA+18] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

[BCL20] J. Bai, S.H. Choi, and Y. Liao. Standard errors for panel data models with unknown clusters. *Journal of Econometrics*, 2020.

[BDT16] A. Backurs, N. Dikkala, and C. Tzamos. Tight hardness results for maximum weight rectangles. *43rd Intl Colloquium on Automata, Languages and Programming (ICALP 2016)*, 2016.

[Ben84] Jon Bentley. Programming pearls: Algorithm design techniques. *Communications of the ACM*, 27(9):865–873, 1984.

[CL05] Kai-Min Chung and Hsueh-I Lu. An optimal algorithm for the maximum-density segment problem. *SIAM Journal on Computing*, 24(2):373–387, 2005.

[DB18] Jaka Demšar and Zoran Bosnić. Detecting concept drift in data streams using model explanation. *Expert Systems with Applications*, 92:546–559, 2018.

[DHMS21] F. Ding, M. Hardt, J. Miller, and L.. Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[EKG19] J. Eckstein, A. Kagawa, and N. Goldberg. REPR: Rule-enhanced penalized regression. *INFORMS Journal on Optimization*, 1(2):143–163, 2019.

[GR14] S. T. Goh and C. Rudin. Box drawings for learning with imbalanced data. In *Proc. 20th ACM SIGKDD*, pages 333–342, 2014.

[Gra14] Pierre Granjon. The CUSUM algorithm – a small review. https://hal.archives-ouvertes.fr/hal-00914697/document, 2014.

[KHF18] S. Kijung, B. Hooi, and C. Faloutsos. Fast, accurate, and flexible algorithms for dense subtensor mining. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(3):1–30, 2018.

[MVED17] D. Malioutov, K. Varshney, A. Emad, and S. Dash. Learning interpretable classification rules with boolean compressed sensing. *Transparent Data Mining for Big and Small Data*, pages 95–121, 2017.

[Nei09] Daniel B. Neill. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting*, 25(3):498–517, 2009.

[NMD+21] R. Nair, M. Mattetti, E. Daly, D. Wei, O. Alkan, and Y. Zhang. What changed? interpretable model comparison. *IJCAI*, pages 2855–2861, 2021.

[Pag54] Ewan S. Page. Continuous inspection schemes. *Biometrika*, 41(1):100–115, 1954.

[PdAB21] E. Pastor, L. de Alfaro, and E. Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proc. 2021 International Conference on Management of Data*, 2021.

[PWKC19] N. Polyzotis, S. Whang, T.K. Kraska, and Y. Chung. Slice finder: Automated data slicing for model validation. *Proc. IEEE Intl. Conf. on Data Engineering (ICDE)*, 2019.

[SSMIN16] S. Speakman, S. Somanchi, E. McFowland III, and D. B. Neill. Penalized fast subset scanning. *Journal of Computational and Graphical Statistics*, 25(2):382–404, 2016.

[TCHL18] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, page 303–310, 2018.

[WDGG19] Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Günlük. Generalized linear rule models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6687–6696, 09–15 Jun 2019.

[YHY+21] F. Yang, K. He, L. Yang, H. Du, J. Yang, B. Yang, and L. Sun. Learning interpretable decision rule sets: A submodular optimization approach. *Advances in Neural Information Processing Systems*, 34:27890–27902, 2021.

[ZN16] Zhe Zhang and Daniel B. Neill. Identifying significant predictive bias in classifiers. *arXiv preprint arXiv:1611.08292*, 2016.

Heavy Sets with Applications to Interpretable Machine Learning Diagnostics: Supplementary Materials

In the supplementary material we describe how to strengthen the heavy set IP formulations in Section A, show timing experiments for larger-scale problems in Section B, and illustrate the observation we made regarding the Lagrangean relaxation and the convex-hull of optimal IP-solutions in 1D in Section C. In Section D we summarize the pre-processing of the folktables dataset used in the paper, and in Section E we examine two additional datasets.

A STRENGTHENING HEAVY SET IP

We described basic IP formulations for max-sum and max-average heavy sets in Section 3. Here we explain how to improve (strengthen) these formulations to help IP-solvers reach the optimal solution faster (the optimal solution itself is not changed). We use CPLEX for this project, but the techniques are applicable across solvers.

The basic max-sum and max-average models in Section 3 can be strengthened in a number of ways. First, note that in a max-sum AND-rule that has as few conditions as possible, one cannot have two upper bound conditions for the same feature (as one of them will be redundant) or two lower bound conditions. This leads to the following inequalities.

$$\sum_{k\in L_i} z_k \le 1, \qquad \forall i \in \{1,\dots,N\}, \tag{25}$$

$$\sum_{k\in U_i} z_k \le 1 \qquad \forall i \in \{1,\dots,N\}. \tag{26}$$

Using the above costraints, we can strengthen (5):

$$\alpha_t + \sum_{k\in L_i: a_{tk}=0} z_k \le 1, \quad \forall t \in \mathcal{T}, i \in \{1,\dots,N\} \tag{27}$$

$$\alpha_t + \sum_{k\in U_i: a_{tk}=0} z_k \le 1, \quad \forall t \in \mathcal{T}, i \in \{1,\dots,N\} \tag{28}$$

We can strengthen (25) and (26) by noticing that the conditions $x_{[i]} < m_k$ and $x_{[i]} > m_l$ cannot be simultaneously active when $k \le l$ and the AND-rule has at least one point satisfying the rule. if $k \in L_i$, then $z_k = 1$ implies that the condition $x_{[i]} < m_l$ for some $m_l \in B_i$ is part of the output AND-rule; we define a function $th(k) \to m_l$. Thus the following set of constraints is valid for a nonempty AND-rule

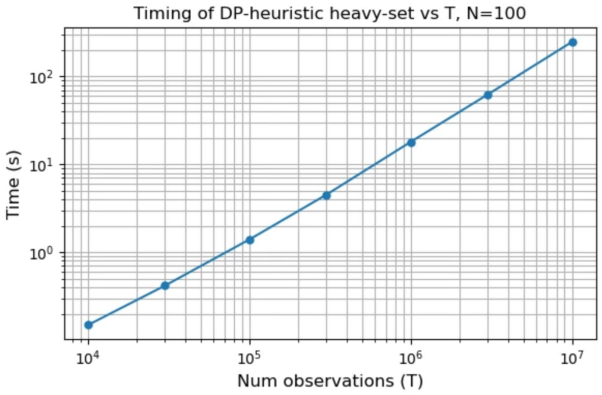

Figure 6: Timing of the DP-heuristic vs number of observations T, log-log scale. $N = 100$ features.

with fewest possible conditions.

$$\sum_{k\in L_i, th(k)\le\nu} z_k + \sum_{l\in U_i, th(l)\ge\nu} z_l \le 1, \quad \forall i \in \{1,\dots,N\}, \nu \in B_i \tag{29}$$

Finally, to constrain the number of features, we add the following constraints:

$$\sum_{k\in L_i} z_k \le y_i, \qquad \forall i \in \{1,\dots,N\} \tag{30}$$

$$\sum_{k\in U_i} z_k \le y_i, \qquad \forall i \in \{1,\dots,N\} \tag{31}$$

$$\sum_{i=1,\dots,N} y_i \le F \tag{32}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,N\} \tag{33}$$

Here each variable y_i represents whether feature i is active or not. The constraints (30) and (31) force y_i to be one, if any upper or lower bound condition associated with feature i is active in the solution. Finally, inequality (32) bounds the number of active features in the AND-rule to F. We note that inequalities (30) and (31) imply, respectively, the inequalities (25) and (26) as y_i is a binary variable, and we thus do not need to include the latter two inequalities in the model. Our final model consists of the constraints in the basic model along with the constraints (27)-(33).

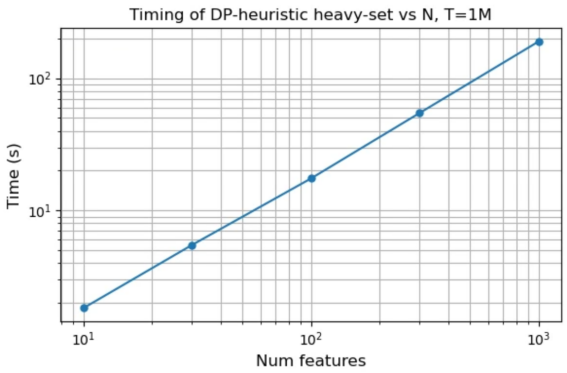

Figure 7: Timing of the DP-heuristic vs number of features N, log-log scale. T =1M data-points. Scaling is roughly linear in the number of features.

B TIMING EXPERIMENT

To demonstrate the scalability of the DP-heuristic for heavy set problems, we include some timing results. In the first experiment we fix the number of available features to be $N = 100$, and vary the number of observations T from 10K to 10M. The results appear in Figure 6 on log-log scale and corresponding numeric values in Table 2. We use continuous (non-discretized) feature values[7]. We use the non-revisit version of the DP-heuristic (to avoid random variation in the number of iterations), and use a bound of 3 active features motivated by interpretability. The coordinate-descent version that allows to revisit features typically takes no more than 3 times of the non-revisit. We use a basic Google cloud platform instance with 16Gb of RAM, and limit computation to 1 CPU core.

In the second experiment we set the number of observations to $T = 1$M, and vary the number of features from $N = 10$ to $N = 1000$. Results appear in Figure 7 and Table 3.

In comparison, the exact IP formulation for the experiment in Section 6 with $T = 1500$ points and $N = 10$ features ranged from several minutes to 30 minutes in the worst case. The running time of the IP does not depend solely on the problem dimension, but also on the data: problems with multiple competing solutions (around the noise-floor) can take significantly longer than problems with a salient planted region with large mean. The proposed DP-heuristic allows to dramatically expand the applicability of heavy-sets to much larger datasets.

C LAGRANGEAN CONVEX HULL OF MAX-SUBARRAY

In Section 4.1 we described an optimal size-constrained solution of the max-subarray-sum problem in (P2), and compared it to the Laragrangean formulation (P3). Problem (P2) produces a family of solutions $\{w^*(K)\}$ parameterized by K, bound on the region size. Problem (P3) produces its own family of solutions $\{w^*(\delta)\}$, parameterized by penalty δ. If we were in the convex optimization setting, then under certain technical conditions, one could show that strong duality holds, and the two sets would be equivalent. However, we are dealing with Lagrangean relaxations of integer optimization problems, so the story is more interesting. Here, we illustrate that while each solution in $\{w^*(\delta)\}$ indeed corresponds to $w^*(K)$ for some K, the converse is not true, and the two sets are not equivalent. Some $w^*(K)$ can not be obtained by $w^*(\delta)$ with any setting of δ. The subset of $\{w^*(K)\}$ that can be reached can be nicely characterized as the corner-points on the convex-hull of the solution path of $\{w^*(\delta)\}$. We illustrate it in Figure 8. Here, $T = 500$, the optimal subarray with $\delta = 0$ has length 416, $\{w^*(K)\}$ (labeled exact) is shown with $K = [1, .., 500]$ in steps of 1. $\{w^*(\delta)\}$ is computed with a very fine grid of δ, and lack of other solutions was confirmed by bisection search.

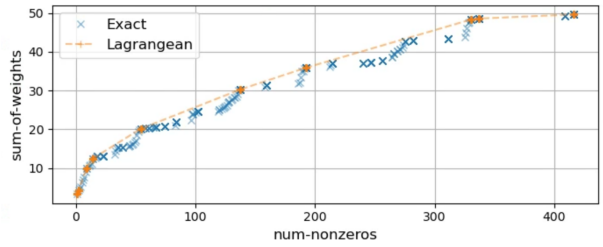

Figure 8: Lagrangean relaxation (P3) is able to find a subset of solutions of (P2) that lie on the convex hull.

Table 2: Timing of DP-heuristic vs. num observations T. $N = 100$ features.

T	10K	30K	100K	300K	1M	3M	10M
time(s)	0.15	0.42	1.40	4.52	18.0	62.3	247.5

Table 3: Timing of DP-heuristic vs. num features N. T =1M datapoints.

N	10	30	100	300	1000
time (sec)	1.83s	5.49s	17.46s	54.53s	189.29s

Recall that solutions $\{w^*(\delta)\}$ have a statistical motivation, described in Section 4.3, so it would be interesting to develop further intuition of how are they different from those solutions in $\{w^*(K)\}$ that can not be reached via Lagrangean relaxation. We hypothesize that they provide a better balance of weight vs. sparsity.

[7] For most practical purposes, discretization with 100 to 1000 levels would be sufficient.

D PREPROCESSING OF FOLKTABLES

We describe the pre-processing that was done on folktables dataset used in experiments in Section 6. We used the provided feature set that is used to predict income[8], which includes 10 features:

- 'AGEP' (age)
- 'COW' (employer type: gov/private/public/self...),
- 'SCHL' (educational level),
- 'MAR' (marital status),
- 'OCCP' (occupation),
- 'POBP' (place of birth, either US state or foreign country),
- 'RELP' (relation of the person responding to survey to the survey subject),
- 'WKHP' (weekly hours worked),
- 'SEX' (gender),
- 'RAC1P' (race).

We discarded 3 high-cardinality categorical features 'OCCP', 'POBP', 'RELP' (with resp. 570, 215, and 18 categories). Without further domain-aware category aggregation (coarsening) these features are not useful for our analysis, and likely to produce highly over-fitted results in heavy sets, since we do not employ category subset regularization. Furthermore, for the 'SCHL' feature, we merged rare categories (e.g. final level of education is 3rd-grade) keeping the following ordered list of categories (in this order):

```
['<=8 Grades', 'Some high school',
'High school', 'GED or alt cred',
'Incomplete college', 'Associate degree',
'Bachelors degree', 'BS + Prof degree',
'Masters degree', 'Doctorate degree']
```

Observations with missing values were discarded (handling them in alternative ways is outside the scope of this paper).

E ADDITIONAL DATASETS

We test our approach on two additional well-known datasets: lending-club and recidivism, which we borrow from (Ribeiro et. al, 2018). We follow the pre-processing and data-cleaning steps as described in the paper and the corresponding GitHub repository `https://github.com/marcotcr/anchor-experiments`.

[8]Namely the indicator "PINCP" that $income \geq 50000\$$.

In contrast to folktables, these two datasets are static[9], so to simulate dataset shift we pick an attribute and split the dataset based on this attribute. Similar to folktables in Section 6.2, we train a logistic regression model on each split and use heavy-sets to find interpretable differences between the two models.

Lending club. The lending-club dataset aims to predict whether a loan on the lending-club website will default, please see (Ribeiro et. al, 2018) for details and preprocessing. There are 9 features, and 11K examples. We use the loan-amount feature to split the dataset: loans below median-size (below 10000$) are used to train model A, and loans above or equal to median size to train model B. The loan-amount feature is then removed from both sets. We follow the steps in Section 6.2 for folktables: we train separate logistic models on training sets for small and large loans, and evaluate both on test-set for large loans. The datapoint weights are set to the differences of predicted probabilities between the two models. The proposed approach finds a region with 17.5% of the data, with 11.3% average difference in predicted default probabilities, while the baseline is 5.1%. The AND-region for the set is:

```
last_fico_range_hi :  584.0 <= x <= 644.0
inq_last_6mths     :    0.0 <= x <=   1.0
revol_util         : -999.0 <= x <=  99.3
```

Recidivism. Recidivism dataset aims to predict whether a person released from prison will be imprisoned again, please see (Ribeiro et. al, 2018) for details and preprocessing. There are 15 features, and 7K examples. We use the educational level to split the dataset: people with less than the median years of education (10 yrs) are used to train model A, and the remaining people to train model B. The educational level is then removed from both sets. We follow the same procedure as for folktables and lending-club above. The proposed approach finds a region with 12.5% of the datapoints, where the average difference in predicted recidivism probabilities is +4.4%, while the baseline is -7.6%. The AND-region identifying the set is:

```
Age          : 39.0 <= x <= 76.0
Alcohol      : False
YearsSchool  : 0.0 <= x <= 10.0
```

[9]Popular datasets focused on dataset shift, such as WILDS, https://wilds.stanford.edu/datasets, are mostly focused on computer vision examples, and ignore interpretability, so are beyond the scope of the paper. Tabular datasets focused on dataset shift are less prevalent in public domain, so we use the attribute split heuristic to emulate dataset shift.

References

M. T. Ribeiro, S. Singh, C. Guestrin (2018). Anchors: High-precision model-agnostic explanations. In *Proc. AAAI conf. on artificial intelligence.*, vol. 32, No. 1.