Data-driven subgroup identification for linear regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Medical studies frequently require to extract the relationship between each covariate and the outcome with statistical confidence measures. To do this, simple parametric models are frequently used (e.g. coefficients of linear regression) but always fitted on the whole dataset. However, it is common that the covariates may not have a uniform effect over the whole population and thus a unified simple model can miss the heterogeneous signal. For example, a linear model may be able to explain a subset of the data but fail on the rest due to the nonlinearity and heterogeneity in the data. In this paper, we propose DDGroup (data-driven group discovery), a data-driven method to effectively identify subgroups in the data with a uniform linear relationship between the features and the label. DDGroup outputs an interpretable region in which the linear model is expected to hold. It is simple to implement and computationally tractable for use. We show theoretically that, given a large enough sample, DDGroup recovers a region where a single linear model with low variance is well-specified (if one exists), and experiments on real-world medical datasets confirm that it can discover regions where a local linear model has improved performance. Our experiments also show that DDGroup can uncover subgroups with qualitatively different relationships which are missed by simply applying parametric approaches to the whole dataset.

1 Introduction

In scientific and medical analyses, simple parameteric models are frequently fit to data to draw qualitative or quantitative conclusions about the relationships between different variables of interest. Typically, a single interpretable model is fit on the entire dataset, implicitly assuming that there are uniform relationships between the covariates and target variable across the whole population. In practice, the data may instead come from a heterogeneous population, where different *subgroups* of the population may obey qualitatively different trends.

For example, suppose we fit a linear model with features including several patient biomarkers, as well as blood concentration of a particular drug, to predict blood pressure. After fitting the model to the whole dataset, we find that there is a statistically signficant negative coefficient on the drug concentration. We may be tempted to conclude that this drug should be administered to a general patient in order to reduce blood pressure. However, there may be a small subgroup in the data (say, patients over the age of 80) for whom the drug actually *increases* blood pressure. In this case, naively fitting a single model to the entire dataset not only reduces our predictive accuracy, it also leads to adverse outcomes for this subgroup of the population.

Modern high-capacity models such as neural networks can help to avoid this problem as they represent a much richer function class. However, these models are often inherently difficult to interpret, making them unsuitable if the primary goal is to draw scientific or clinical conclusions about the data rather

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

than simply having good predictive performance. This motivates our desire to find interpretable regions in the data where interpretable models (such as linear regression) perform well. We call this the *subgroup selection* problem.

1.1 Our contributions

In this work, we consider a flexible formalization of the subgroup selection problem. We propose an general algorithmic framework and a specific instantiation, DDGroup (data-driven group discovery), for data-driven subgroup selection. We prove that DDGroup has desirable theoretical properties, and results on synthetic and real data show the effectiveness of DDGroup in practice.

1.2 Related work

Subgroup identification is an important topic in biostatistics [20]. Here, the main focus is on identifying subsets of the population with a significant beneficial treatment effect from a new drug or procedure. Common approaches include *global outcome modeling*, in which the user models the patient response with and without treatment separately, then reconstructs the treatment effect from these models; *global treatment modeling*, in which the user models the treatment effect directly; and *local modeling*, where the user tries to identify a region with a strong positive treatment effect. Of these approaches, our method is most closely related to the local modeling approach. However, existing local modeling methods typically use tree-based greedy approaches to region selection which do not come with any guarantees [20].

Piecewise linear regression is an existing method for adding flexibility to linear models while preserving interpretability. Here, the assumption is that the response is a piecewise linear function of the covariates. Early works focused on the one-dimensional covariate case [28], and recently methods have been proposed for piecewise linear regression in higher dimensions [26, 11]. Unlike the piecewise linear setting, we make no assumptions on the regression function outside of the "good" region which we are trying to detect.

Lastly, as we seek to learn a subset of the data on which we are willing to make predictions, our work is connected to the literature on learning with rejection [9] or learning to defer [21, 23, 19], in which a model is given the option not to make a prediction. These works focus primarily on classification and decide whether or not to make a prediction on individual data point via thresholding model confidence. While this implicitly defines a subgroup on which we expect the model to perform well—namely, the points for which the model does not defer—, this subgroup will typically be uninterpretable (if the model is a neural network). If logistic regression is used, the deferred subgroup will be a slab between two parallel hyperplanes, which may be considered interpretable but is fairly inflexible in terms of the region selected. In our setting, we focus on the regression problem and on explicitly defining an interpretable region in which we will not defer.

Our problem framework has connections to list-decodable learning [7, 17, 25] and conditional linear regression [16, 5]. For the sake of brevity, we discuss these and other related works in the appendix.

2 Problem setup

The general subgroup selection problem can be formulated as follows. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ denote the sample space, $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ denote a class of functions (e.g. linear regression models), and let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function measuring the performance of our model. We will always have $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$. Our goal is to find a (interpretable, large as possible) region $R \subseteq \mathcal{X}$ of the *feature space* where

$$\operatorname*{argmin}_{f \in \mathcal{F}} \mathbb{E}[\ell(y, f(x)) \mid x \in R]$$

is small. In order to satisfy the interpretability criterion, we will primarily consider regions R which are axis-aligned boxes. This corresponds to a subgroup where each feature lies within a specified range (corresponding to the sides of the axis-aligned box).

For this paper, we will specify the function class $\mathcal{F}$ to be linear models and the loss $\ell(y, \hat{y}) = (y - \hat{y})^2$ to be the squared loss. For our theoretical results, we will assume that there exists a "good" region $R^* \subseteq \mathcal{X}$ where the linear model is well-specified with low conditional variance of $y|x$. In this case, the goal will be to recover R^*.

3 Algorithmic framework

We introduce an algorithmic framework with three distinct phases. In Phase 1, we find a "core set" of points which should belong to the good region, then fit a model to these points. In Phase 2, we reject points in the larger dataset which cannot feasibly follow the same model as the core group based on a hypothesis test. Finally, in Phase 3, we find a large region which contains only non-rejected points.

Phase 1: Given a choice of core group size k, for each datapoint, we fit a local model to that point's k nearest neighbors. We then select the group of points with the lowest "training error" of its local model as the core group. The pseudocode for selecting the core group is provided in Algorithm 1 in the appendix.

Phase 2: We use the threshold

$$\rho^{\text{grow}}_{\alpha,k,n}(x_i) = \underbrace{\sigma\lambda_{\min}^{-1}\|x_i\|\sqrt{\frac{d\log\frac{4d}{\alpha}}{k}}}_{\text{core model error}} + \underbrace{\sigma\sqrt{2\log\frac{4n}{\alpha}}}_{\text{random fluctuations}}. \tag{1}$$

This threshold is derived from a hypothesis test; for more details, see the appendix. Here k is the size of the data used to fit the model $\hat{\beta}$ and n is the size of the training set. The inclusion labels ℓ_i are then computed as $\ell_i = \mathbb{1}\{|y_i - \hat{\beta}^\top x_i| \geq \rho_{\alpha,n,N}(x_i)\}$. We define the set of *rejected points* $X_{\text{rej}} = \{x_i \in X \mid \ell_i = 1\}$.

Phase 3: Roughly speaking, we "grow" a hyperrectangle which contains the core points. The sides of the hyperrectangle expand until each side "hits" a rejected point, at which point we stop growing the region in this direction. Once all sides are supported by either a rejected point or a larger bounding region for all of the data, we stop and return this region. A more thorough explanation (along with pseudocode) can be found in the appendix.

Combining Phases 1-3 gives DDGroup, an algorithm for automatic subgroup selection. Our main theoretical result shows that DDGroup precisely recovers R^* given sufficient data.

Theorem 1. *main Assume that there is an axis-aligned box R^* in which the linear model is well-specified, that the variance of $y|x$ inside R^* is not too large, and that the variability of $y|x$ outside of R^* is large enough. Then as $n, k \to \infty$ with $k = o(n)$, there exist selections of the hyperparameters for DDGroup such that it returns $\hat{R}$ with $R^* \subseteq \hat{R}$ with probability at least $1 - \alpha$. Furthermore, $\text{vol}(\hat{R} \setminus R^*) \to 0$ for any fixed $\alpha > 0$.*

4 Experiments

In this section, we evaluate the performance of DDGroup on real-world medical datasets. Additional details plus experiments on synthetic data can be found in the appendix.

Methods for comparision We compare DDGroup with both standard linear regression and linear model tree (LMT). 1) The standard linear regression — a linear model fit to the whole dataset — is used as a baseline comparison. It is equivalent to the situation where the selected region includes all of the data and it is the method employed by the original medical studies on the real-world datasets we consider. 2) We further compare DDGroup with linear model tree — decision tree with a linear regression model in each leaf [29, 24]. Though LMT is not designed for subgroup identification, we can still use its decision path as a way to select cohorts. In order to identify the most coherent subgroup, we pick the leaf of LMT with the smallest MSE.

Datasets We evaluate our method on five real-world medical related datasets, where linear coefficients were used for interpretation in their original publications: Brazil Health [6], China Glucose [30], China HIV [32], Dutch Drinking [3], and Korea Grip [31].

Performance evaluation DDGroup correctly identifies a subgroup on which the linear model has low test error and consistently outperforms the baseline methods on all five real-world medical

Table 1: Performance of baseline (linear regression model on the whole data), linear tree model and DDGroup on the real-world datasets. Here d denotes the dimension of the features and subgroup size is the proportion of selected datapoints. We average the results for 10 runs of different random splits.

Dataset	Task	d	Test MSE			Subgroup Size	
			Baseline	LTM	DDGroup	LTM	DDGroup
Brazil	HF	6	0.80 ± 0.06	0.18 ± 0.01	**0.04 ± 0.00**	13% ± 1%	6% ± 0%
Health	stroke	6	1.14 ± 0.22	0.17 ± 0.01	**0.06 ± 0.00**	15% ± 1%	6% ± 0%
China	SUA-F	11	0.83 ± 0.02	0.72 ± 0.03	**0.69 ± 0.06**	33% ± 8%	21% ± 3%
Glucose	SUA-M	11	0.94 ± 0.01	0.81 ± 0.03	0.81 ± 0.04	20% ± 5%	8% ± 1%
China HIV	stigma	27	0.84 ± 0.01	0.88 ± 0.04	**0.69 ± 0.04**	39% ± 7%	21% ± 3%
Dutch Drinking	inh	16	0.64 ± 0.01	0.50 ± 0.03	0.50 ± 0.02	23% ± 7%	11% ± 2%
	wm	16	0.71 ± 0.01	**0.56 ± 0.01**	0.57 ± 0.02	20% ± 3%	9% ± 1%
	sha	16	0.64 ± 0.01	0.46 ± 0.02	**0.42 ± 0.02**	13% ± 2%	10% ± 1%
Korea Grip	strength	11	0.71 ± 0.02	0.88 ± 0.07	**0.69 ± 0.04**	36% ± 7%	20% ± 3%

datasets (Table 1). We demonstrate that there exists subgroups within the real-world population where linear model is a good proxy and should be used to enhance interpretability. Our current method focuses on finding the most coherent region within the dataset, thus it always identifies small subgroups with the strongest signal. If a larger subgroup is desired, one may enforce this by selecting the best region which includes e.g. at least a certain fraction of the validation set. In our case, we required that at least 5% of validation was selected. We also remark that DDGroup is computationally efficient. The average runtime for Algorithm 3 across one run of each dataset was 1.98 seconds on an AMD 7502 CPU, and no individual dataset took longer than 10 seconds.

Case study Here we use China HIV Datase to illustrate how DDGroup can enhance understanding of the data. The original study analyzes how different HIV infection routes affect the internalized stigma by fitting a multivariate linear regression model with confounders [32]. In their main results, the blood transfusion route is found to have positive effect on internalized stigma (coefficient β larger than zero), but in low confidence with large p value. In our analysis, we observed similar behavior: after data standardization, the linear model on whole dataset predicts blood transfusion route to have positive effect on internalized stigma with β of 0.12, but low confidence level with p value of 0.67. In this case, DDGroup identifies a subgroup of 21% participants where blood transfusion route has the opposite effect on stigma ($\beta = -1.71$) with strong signal (p value = 0.006). The selected subgroup consists of younger participants with lower self-esteem, lower anxiety level, and less social support. The result indicates that while blood transfusion route seems not to associate with internalized stigma in the general population living with HIV, it is coherently associated with lower stigma in a certain subpopulation. This seems plausible, as the other infection routes include sex with stable partners, sex with casual partners, sex with commercial partners, and injecting drug use. Younger participants may have stronger feelings of shame associated with these activities than older participants. In general, interpretation of the learned selection rules could be of great interest in real applications.

5 Conclusion

In this paper, we considered a flexible formalization of the cohort selection problem. We proposed a general algorithmic framework and a specific instantiation, DDGroup, for solving the problem, and we proved that DDGroup recovers the correct subgroup given sufficient data. Experiments on both synthetic and real data verify our theory and show the practical usefulness of DDGroup.

There are a number of important open questions which remain to be addressed. If a hyperparameter search is used with DDGroup to train a linear model (as we did with our real data experiments), further analysis is needed to give meaningful (but valid) p-values for the resulting model coefficients. (For any extensive hyperparameter search, a naive Bonferroni correction is likely to be too conservative.) Another important question is how to extend our framework to classification and survival analysis data.

References

[1] Jonathan Backer and J Mark Keil. The mono-and bichromatic empty rectangle and square problems in all dimensions. In *Latin American Symposium on Theoretical Informatics*, pages 14–25. Springer, 2010.

[2] Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight hardness results for maximum weight rectangles. *arXiv preprint arXiv:1602.05837*, 2016.

[3] Sarai R Boelema, Zeena Harakeh, Martine JE Van Zandvoort, Sijmen A Reijneveld, Frank C Verhulst, Johan Ormel, and Wilma AM Vollebergh. Adolescent heavy drinking does not affect maturation of basic executive functioning: longitudinal findings from the trails study. *PloS one*, 10(10):e0139186, 2015.

[4] Julii Brainard, Rob D'hondt, Engy Ali, Rafael Van den Bergh, Anja De Weggheleire, Yves Baudot, Frederic Patigny, Vincent Lambert, Rony Zachariah, Peter Maes, et al. Typhoid fever outbreak in the democratic republic of congo: Case control and ecological study. *PLoS neglected tropical diseases*, 12(10):e0006795, 2018.

[5] Diego Calderon, Brendan Juba, Sirui Li, Zongyi Li, and Lisa Ruan. Conditional linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 2164–2173. PMLR, 2020.

[6] Denise de Fátima Barros Cavalcante, Valéria Silva Cândido Brizon, Livia Fernandes Probst, Marcelo de Castro Meneghim, Antonio Carlos Pereira, and Gláucia Maria Bovi Ambrosano. Did the family health strategy have an impact on indicators of hospitalizations for stroke and heart failure? longitudinal study in brazil: 1998-2013. *PLoS One*, 13(6):e0198428, 2018.

[7] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60, 2017.

[8] Steven A Cohen, Mary L Greaney, and Natalie J Sabik. Assessment of dietary patterns, physical activity and obesity from a national survey: Rural-urban health disparities in older adults. *PLoS One*, 13(12):e0208268, 2018.

[9] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016.

[10] Ilias Diakonikolas, Daniel Kane, Ankit Pensia, Thanasis Pittas, and Alistair Stewart. Statistical query lower bounds for list-decodable linear regression. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] Ilias Diakonikolas, Jerry Li, and Anastasia Voloshinov. Efficient algorithms for multidimensional segmented regression. *arXiv preprint arXiv:2003.11086*, 2020.

[12] David P Dobkin, Herbert Edelsbrunner, and Mark H Overmars. Searching for empty convex polygons. In *Proceedings of the fourth annual symposium on Computational geometry*, pages 224–228, 1988.

[13] Adrian Dumitrescu and Minghui Jiang. On the largest empty axis-parallel box amidst n points. *Algorithmica*, 66(2):225–248, 2013.

[14] Maria Angeles Garcia-Leon, Maria Isabel Peralta-Ramirez, Laura Arco-Garcia, Borja Romero-Gonzalez, Rafael A Caparros-Gonzalez, Noelia Saez-Sanz, Ana Maria Santos-Ruiz, Eva Montero-Lopez, Andres Gonzalez, and Raquel Gonzalez-Perez. Hair cortisol concentrations in a spanish sample of healthy adults. *PloS one*, 13(9):e0204807, 2018.

[15] Nicole A Haberland, Christine A Kelly, Drosin M Mulenga, Barbara S Mensch, and Paul C Hewett. Women's perceptions and misperceptions of male circumcision: a mixed methods study in zambia. *PloS one*, 11(3):e0149517, 2016.

[16] Brendan Juba. Conditional Sparse Linear Regression. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[17] Sushrut Karmalkar, Adam Klivans, and Pravesh Kothari. List-decodable linear regression. *Advances in neural information processing systems*, 32, 2019.

[18] Anshul Kastor and Sanjay K Mohanty. Disease-specific out-of-pocket and catastrophic health expenditure on hospitalization in india: do indian households face distress health financing? *PloS one*, 13(5):e0196106, 2018.

[19] Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 154–165, 2021.

[20] Ilya Lipkovich, Alex Dmitrienko, and Ralph B D'Agostino Sr. Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials. *Statistics in medicine*, 36(1):136–196, 2017.

[21] David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in Neural Information Processing Systems*, 31, 2018.

[22] Bidhubhusan Mahapatra and Niranjan Saggurti. Exposure to pornographic videos and its effect on hiv-related sexual risk behaviours among male migrant workers in southern india. *PloS one*, 9(11):e113599, 2014.

[23] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

[24] Duncan Potts and Claude Sammut. Incremental learning of linear model trees. *Machine Learning*, 61(1):5–48, 2005.

[25] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180. SIAM, 2020.

[26] Ali Siahkamari, Aditya Gangrade, Brian Kulis, and Venkatesh Saligrama. Piecewise linear regression via a difference of convex functions. In *International Conference on Machine Learning*, pages 8895–8904. PMLR, 2020.

[27] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[28] Elisabeth Vieth. Fitting piecewise linear regression functions to biological responses. *Journal of applied physiology*, 67(1):390–396, 1989.

[29] Yong Wang and Ian H Witten. Induction of model trees for predicting continuous classes. 1996.

[30] Yunyang Wang, Jingwei Chi, Kui Che, Ying Chen, Xiaolin Sun, Yangang Wang, and Zhongchao Wang. Fasting plasma glucose and serum uric acid levels in a general chinese population with normal glucose tolerance: A u-shaped curve. *PLoS One*, 12(6):e0180111, 2017.

[31] Lihui Wen, Min-Ho Shin, Ji-Hyoun Kang, Yi-Rang Yim, Ji-Eun Kim, Jeong-Won Lee, Kyung-Eun Lee, Dong-Jin Park, Tae-Jong Kim, Sun-Seog Kweon, et al. Association between grip strength and hand and knee radiographic osteoarthritis in korean adults: Data from the dong-gu study. *PLoS One*, 12(11):e0185343, 2017.

[32] Chen Zhang, Xiaoming Li, Yu Liu, Shan Qiao, Liying Zhang, Yuejiao Zhou, Zhenzhu Tang, Zhiyong Shen, and Yi Chen. Stigma against people living with hiv/aids in china: does the route of infection matter? *PloS one*, 11(3):e0151078, 2016.

A Related work cont'd.

Our problem framework also has connections to list-decodable learning [7], specifically list-decodable linear regression [17, 25]. In the list-decodable setting, we assume that an α fraction of the data come from a "trusted" source which we are trying to model; this would correspond to the subset of our data belonging to the good region. The goal is to output a small list (polynomial in α^{-1}) which contains a model that will perform well on the trusted data. While an algorithm for the list-decodable linear regression problem will return a model that performs well for the good region, it does not directly solve the problem of actually finding the good region itself.

Our work is also similar in spirit to previous works on conditional linear regression [16, 5]. In this setting, the goal is also to find the largest possible subset of the data for which there is an accurate linear model. However, the subgroup identification is made in terms of *pre-defined* binary features, which are assumed to be provided with the data in addition to the regressor variables. While one could instantiate our problem by defining the binary inclusion variables as indicators of whether or not each regressor is above or below a certain threshold, doing so would result in exponentially many possible selection rules and will therefore be computationally intractable for our setting. One can also view our work as finding data-driven binary inclusion labels for the conditional linear regression problem.

A core element of our problem setting is in selecting a region which avoids certain "bad" points. Related problems have been extensively studied in the computational geometry community [12, 1, 13], but even approximate algorithms for solving related problems are not practical for high dimensions, and indeed even some seemingly simple region selection problems can be shown to be NP hard [2]. We propose tractable alternatives and show that they have desirable properties both theoretically and empirically.

B Detailed algorithm descriptions

Here we provide the pseudocode for DDGroup and its components. We denote a dataset $D = (X, Y)$ to be a collection of n feature vectors (collected in $X \in \mathbb{R}^{n \times d}$) and corresponding labels (collected in $Y \in \mathbb{R}^n$). Here $\text{KNN}(x, k, D)$ denotes the k nearest neighbors of x (and their corresponding labels) in the dataset D, $\text{OLS}(D)$ denotes the output of ordinary least squares on feature matrix X and response vector Y, and $\text{MSE}(\hat{\beta}, D)$ denotes the mean squared error of linear model $\hat{\beta}$ on the data X, Y. In addition, $\lambda_{\min}(M)$ denotes the minimum eigenvalue of the PSD matrix M. Note that if the variance σ^2 is not known, we can replace it with a standard unbiased estimate computed on the core group.

Algorithm 1 Core group selection

procedure CoreGroup(k, D)
  $\text{MSE}^* \leftarrow \infty$
  for $(x, y) \in D$ **do**
    $D_{\text{nbhd}} = (X_{\text{nbhd}}, Y_{\text{nbhd}}) \leftarrow \text{KNN}(x, k, D)$
    $\hat{\beta} \leftarrow \text{OLS}(X_{\text{nbhd}}, Y_{\text{nbhd}})$
    if $\text{MSE}(\hat{\beta}, D_{\text{nbhd}}) < \text{MSE}^*$ **then**
      $D_{\text{core}} \leftarrow D_{\text{nbhd}}$
      $\text{MSE}^* \leftarrow \text{MSE}(\hat{\beta}, D_{\text{nbhd}})$
    end if
  end for
  return D_{core}
end procedure

The GrowBox algorithm is somewhat opaque, so we offer additional explanation below. Let $U \subseteq \mathbb{R}^d$. We define the *directed infinity norm* $\|x\|_{U,\infty}$ by

$$\|x\|_{U,\infty} = \max_{u \in U} x^\top u.$$

We note that for many sets U, $\|\cdot\|_{U,\infty}$ may not be a norm, nor even a seminorm. In what follows, U will initially be defined as $U = \{\pm e_i\}_{i=1}^d$, in which case $\|\cdot\|_{U,\infty} = \|\cdot\|_\infty$ coincides with the

Algorithm 2 Growing box

procedure GROWBOX(c, X_{rej}, U)
  $X_{\text{rej}} \leftarrow X_{\text{rej}} + \{-c\}$ ▷ Center the points at c. $+$ denotes Minkowski sum.
  $\hat{R} \leftarrow \emptyset$
  while $X_{\text{rej}} \neq \emptyset$ **do**
    $x^* \leftarrow \text{argmin}_{x \in X_{\text{rej}}} \{\|x\|_{U,\infty}\}$
    $a^* \leftarrow \|x^* + c\|_{U,\infty}$ ▷ Define the constraints w.r.t. the uncentered points
    $u^* \leftarrow \text{argmax}_{u \in U} \{u^\top x^*\}$ ▷ u^* is the next support direction for the polytope
    Add (u^*, a^*) to $\hat{R}$
    Remove u^* from U
    $X_{\text{rej}} \leftarrow \{x \in X_{\text{rej}} \mid x^\top u^* < a^*\}$
  end while
  return $\hat{R}$
end procedure

Algorithm 3 Data-driven subgroup selection

procedure DDSUBGROUP(α, k, U, D)
  $D_{\text{core}} \leftarrow \text{COREGROUP}(k, D)$ ▷ Phase 1: Find a core group and fit a coarse model.
  $\hat{\beta} \leftarrow \text{OLS}(D_{\text{core}})$

  $\lambda_{\min} \leftarrow \lambda_{\min}(\frac{1}{k} X_{\text{core}}^\top X_{\text{core}})$ ▷ Phase 2: Label which points should be excluded.
  for $i = 1, \ldots, n$ **do**
    $\ell_i \leftarrow \mathbb{1}\{|y_i - \hat{\beta}^\top x_i| \geq \rho_{\alpha,k,n}(x_i)\}$
  end for
  $X_{\text{rej}} \leftarrow \{x_i \in X \mid \ell_i = 1\}$

  $c \leftarrow \text{MEAN}(X_{\text{core}})$ ▷ Phase 3: Approximate the good region.
  $\hat{R} \leftarrow \text{GROWBOX}(c, X_{\text{rej}}, U)$
  return $\hat{R}$
end procedure

usual infinity norm on $\mathbb{R}^d$. We will then gradually remove directions which are no longer relevant to consider.

The region will be described in terms of linear constraints. We will overload notation and use a set $R = \{(u_i, a_i)\}_{i=1}^m$ of constraint directions and values to denote the region $R = \{x \in B : x^\top u_i \leq a_i\}$.

The pseudocode for the growing box is provided in Algorithm 2. When $U = \{\pm e_i\}_{i=1}^d$, Algorithm 2 begins expanding an ℓ_∞ ball centered at c with each side growing at an equal rate. Whenever one of the sides runs into a rejected point, we add the corresponding linear constraint and continue growing the other sides of the box. (The directed infinity norm is what we use to measure which point will collide with the box next.) This continues until all sides of the box have a support point, or there are no points left to constrain the box.

Note that the set U simply specifies the normal vectors to the sides of the constraint polytope. The lengths of these vectors effectively determine the speed at which the constraint region will grow in that direction. Thus, by changing U, this method can select polytopes of any desired shape. Since axis-aligned boxes provide easily interpretable inclusion criteria, we use such regions for all of our experiments.

C Omitted proofs

We restate the lemmas and theorems here for convenience. We make the additional assumption here that $y|x$ is $O(1)$ sub-Gaussian for all x. Let τ be a uniform upper bound on the sub-Gaussian parameter of $y|x$, independent of x. We make the following assumptions.

1. The samples $(x_i, y_i) \overset{\text{iid}}{\sim} \mathcal{P}$ for a probability distribution $\mathcal{P}$ on $\mathcal{Z}$.
2. The marginal distribution of x has a density f with respect to the Lebesgue measure.
3. There is a region R^* with $\text{vol}(R^*) > 0$ such that $f(x) \geq \delta > 0$ for all $x \in \mathcal{X}$.
4. Conditional on $x \in R^*$, y is generated according to the linear model.
5. there exists a constant $\sigma_0 > \sigma\sqrt{2}$ such that, for any collection of n independent data, we have

$$\mathbb{E}\left[\max_{1\leq i\leq n} |y_i - \mathbb{E}[y_i|x_i]| \,\middle|\, x_1, \ldots, x_n\right] \geq \sigma_0\sqrt{\log n}, \tag{2}$$

$$\text{var}\left(\max_{1\leq i\leq n} |y_i - \mathbb{E}[y_i|x_i]| \,\middle|\, x_1, \ldots, x_n\right) \leq \frac{\sigma_0^2}{n}. \tag{3}$$

Assumptions 2 and 3 ensure that the samples will cover the sample space (so that we can detect R^*), and Assumption 4 ensures that our model is well-specified on R^*. Finally, Assumption 5 ensures that R^* is in fact the "best" region for us to select, namely, there is no other region where we can have better predictive power. It can be thought of as a "super-Gaussian" assumption on the noise in $y_i|x_i$, and it will hold e.g. if $y_i|x_i$ is Gaussian with variance $C\sigma_0^2$ for some sufficiently large absolute constant C. This condition ensures that the random fluctuations in y_i are large enough to be detected by the test. Also, note that (2) and (3) still hold if $\mathbb{E}[y_i|x_i]$ in each inequality is replaced by $x_i^\top \hat{\beta}$ for any fixed $\hat{\beta}$.

Lemma 2. *Let Z_i be independent random variables with uniformly bounded fourth moments. Then*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n Z_i - \mathbb{E}Z_i\right| \geq n^{-1/8}\right) = \mathcal{O}(n^{-3/2}).$$

Proof. This is just a generalization of the standard Chebyshev inequality, and the proof proceeds in the same way. By Markov's inequality, we have

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n Z_i - \mathbb{E}Z_i\right| \geq t\right) = \mathbb{P}\left(\left(\sum_{i=1}^n Z_i - \mathbb{E}Z_i\right)^4 \geq n^4t^4\right) \leq \frac{\mathbb{E}[(\sum_{i=1}^n Z_i - \mathbb{E}Z_i)^4]}{n^4t^4}.$$

Expanding $(\sum_{i=1}^n Z_i - \mathbb{E}Z_i)^4$ and taking expectation, by linearity of expectation and independence of the Z_i, the only terms which do not vanish are of the form $(Z_i - \mathbb{E}Z_i)^4$ and $(Z_i - \mathbb{E}Z_i)^2(Z_j - \mathbb{E}Z_j)^2$. There are $\mathcal{O}(n^2)$ of all of these terms, each with expectation bounded by $\mathcal{O}(1)$, so we obtain

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n Z_i - \mathbb{E}Z_i\right| \geq t\right) = \mathcal{O}\left(\frac{1}{n^2t^4}\right).$$

Substituting $t = n^{-1/8}$ completes the proof. □

The following lemma shows that the core group selection contains only "good" points.

Lemma 3. *Under Assumptions 1-5, the core group selected by Algorithm 1 has $X_{core} \subseteq R^*$ with probability approaching 1 as $n, k \to \infty$ with $k = o(n)$.*

Proof. Consider any group of k neighboring points. We apply Lemma 2 conditional on $x_1, \ldots, x_k$. (Note that we must assume bounded 8th moments on $y_i|x_i$.) Since the data are independent, with

probability at least $1 - \mathcal{O}(k^{-3/2})$, we have

$$\begin{aligned}\frac{1}{k}\sum_{i=1}^{k}(x_i^\top\beta - y_i)^2 &= \frac{1}{k}\sum_{i=1}^{k}(x_i^\top\beta - \mathbb{E}[y_i|x_i])^2 + \frac{2}{k}\sum_{i=1}^{k}(x_i^\top\beta - \mathbb{E}[y_i|x_i])(\mathbb{E}[y_i|x_i] - y_i) \\ &\quad + \frac{1}{k}\sum_{i=1}^{k}(y_i - \mathbb{E}[y_i|x_i])^2 \\ &\geq \frac{1}{k}\sum_{i=1}^{k}(y_i - \mathbb{E}[y_i|x_i])^2 + \frac{2}{k}\sum_{i=1}^{k}(x_i^\top\beta - \mathbb{E}[y_i|x_i])(\mathbb{E}[y_i|x_i] - y_i) \\ &\geq \frac{1}{k}\sum_{i=1}^{k}\mathrm{var}(y_i|x_i) - k^{-1/8} - 2k^{-1/8}. \end{aligned} \tag{4}$$

The final inequality holds because $\mathbb{E}[(y_i - \mathbb{E}[y_i|x_i])^2|x_1, \ldots, x_k] = \mathrm{var}(y_i|x_i)$ and

$$\mathbb{E}[(x_i^\top\beta - \mathbb{E}[y_i|x_i])(\mathbb{E}[y_i|x_i] - y_i)|x_1, \ldots, x_k] = 0.$$

Furthermore, note that this lower bound is independent of β.

If we take $k = n^{3/4}$, then a simple union bound over (at most) n different neighborhoods of k points shows that (4) holds for all of these neighborhoods simultaneously with probability at least $1 - \mathcal{O}(n^{-1/8})$. Henceforth we will assume that we are on this high probability "good" event.

Assumptions 1-3, along with $k = o(n)$, imply that with probability approaching 1 as $n \to \infty$, there exists a neighborhood of k points all of which lie in the interior of R^* (say $x_1, \ldots, x_k$). For this group of points, we have $\mathbb{E}[y_i|x_i] = \beta^\top x_i$. Thus, setting $\beta = \beta$ and using logic similar to the derivation of (4), we have

$$\frac{1}{k}\sum_{i=1}^{k}(x_i^\top\beta - y_i)^2 = \sigma^2 + k^{-1/8}.$$

Finally, fix $\varepsilon > 0$ and consider a set of k nearest neighbors $x_1, \ldots, x_k$, at least εk of which do not belong to R^* (WLOG $x_1, \ldots, x_{\varepsilon k}$). By (4), for such a group, we have

$$\frac{1}{k}\sum_{i=1}^{k}(x_i^\top\beta - y_i)^2 \geq \varepsilon\sigma_0^2 + (1-\varepsilon)\sigma^2 - 3k^{-1/8} = \sigma^2 + \varepsilon(\sigma_0^2 - \sigma^2) - 3k^{-1/8} > \sigma^2 + k^{-1/8}$$

for k sufficiently large. Since Algorithm 1 returns a group with the smallest MSE, such a group will not be selected. We conclude that as $n, k \to \infty$, Algorithm 1 will return a group in which all but $o(k)$ points belong to R^*.

□

The next result states that if we have selected a good core group, then with high probability, we will not erroneously reject any points that actually belong to R^*.

Lemma 4. *Suppose that all of the core points belong to R^*. Then with probability at least $1 - \alpha$, none of the points in X_{rej} belong to R^*.*

Proof. First, we show that $\hat{\beta}$ is close to β with high probability. Let $X \in \mathbb{R}^{k\times d}$ denote the design matrix for the core group and $Y \in \mathbb{R}^k$ denote the response vector. Since all of the core points belong to R^*, we have $Y = X\beta + E$, where $E \sim \mathcal{N}(0, \sigma^2 I_k)$. It follows that

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y = (X^\top X)^{-1}X^\top(X\beta + E) = \beta + (X^\top X)^{-1}\sum_{i=1}^{k}\varepsilon_i x_i,$$

where ε_i are the individual error terms collected in E. It therefore follows that

$$\|\hat{\beta} - \beta\| \leq \left\|\left(\frac{1}{k}X^\top X\right)^{-1}\right\|\left\|\frac{1}{k}\sum_{i=1}^{k}\varepsilon_i x_i\right\| = \sigma\lambda_{\min}^{-1}\left\|\frac{1}{k}\sum_{i=1}^{k}g_i x_i\right\|,$$

where $g_i \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and $\lambda_{\min} = \lambda_{\min}(\frac{1}{k}X^\top X)$. It remains to bound $\left\|\frac{1}{k}\sum_{i=1}^k g_i x_i\right\|$ with high probability. Observe that

$$\mathbb{P}\left(\left\|\frac{1}{k}\sum_{i=1}^k g_i x_i\right\| \geq t\right) \leq \sum_{j=1}^d \mathbb{P}\left(\left|\frac{1}{k}\sum_{i=1}^k g_i x_{ij}\right| \geq \frac{t}{\sqrt{d}}\right).$$

Standard Gaussian concentration results (see e.g. [27]) show that the RHS is bounded by $2d\exp\left(\frac{-ct^2k}{d}\right)$ for some universal constant c. Setting this bound equal to $\alpha/2$ and solving for t, we see that $\|\hat{\beta} - \beta\| \leq C\sigma\lambda_{\min}^{-1}\sqrt{\frac{d\log\frac{4d}{\alpha}}{k}}$ with probability at least $1-\alpha/2$ for some universal constant C.

Next, we look at $|y_i - x_i^\top\hat{\beta}|$ for a point $x_i \in R^*$. In this case, applying the triangle inequality and Cauchy-Schwarz, we have

$$|y_i - x_i^\top\hat{\beta}| = |x_i^\top\beta - x_i^\top\hat{\beta} + \varepsilon_i| \leq \|\beta - \hat{\beta}\|\|x_i\| + |\varepsilon_i|.$$

Since our dataset contains n points, there are at most n points in R^*. Thus again by standard Gaussian concentration results and a union bound, we have that $|\varepsilon_i| \leq \sigma\sqrt{2\log\frac{4n}{\alpha}}$ for all $x_i \in R^*$ simultaneously with probability at least $1-\alpha/2$. A final union bound shows that

$$|y_i - x_i^\top\hat{\beta}| \leq C\sigma\lambda_{\min}^{-1}\sqrt{\frac{d\log\frac{4d}{\alpha}}{k}} + \sigma\sqrt{2\log\frac{4n}{\alpha}}$$

for all $x_i \in R^*$ with probability at least $1-\alpha$, as desired. □

We remark that although Lemma 3 does not guarantee that *all* of the core points will belong to R^*, the threshold used for Lemma 4 will remain the same up to lower order correction terms. This can be shown in a straightforward manner by comparing

$$\left(\frac{1}{k}\sum_{i:x_i\in R^*} x_i x_i^\top + \frac{1}{k}\sum_{i:x_i\notin R^*} x_i x_i^\top\right)^{-1} \quad \text{vs.} \quad \left(\frac{1}{k}\sum_{i:x_i\in R^*} x_i x_i^\top\right)^{-1}$$

via the Sherman-Morrison formula. Using the closed-form expression for $\hat{\beta}$ will then show that we get the same error as in Lemma 4 up to a bias of order $\mathcal{O}((\#\text{ core points not in } R^*)/k) = o(1)$. For brevity, we omit the complete details.

Theorem 5 (Formal version of Theorem 1)**.** *Assume that Assumptions 1-5 hold and further suppose that R^* is an axis-aligned box. Then as $n, k \to \infty$ with $k = o(n)$, there exist positive scalars $\{s_j^\pm\}_{j=1}^d$ (which may depend on the dataset) such that with $U = \{s_j^+ e_j, -s_j^- e_j\}_{j=1}^d$, Algorithm 3 returns $\hat{R}$ with $R^* \subseteq \hat{R}$ with probability at least $1-\alpha$. Furthermore, $\text{vol}(\hat{R}\setminus R^*) \to 0$ for any fixed $\alpha > 0$.*

Proof. By Lemma 3, all but an $o(1)$ fraction of the core points belong to R^*, thus their average (and therefore the point from which we begin growing the box in Algorithm 2) lies in the interior of R^*. Let c be the average of the core point features, and let ∂R^* denote the boundary of R^*. For each $j = 1, \ldots, d$, denote by $\partial R^*_{j,+}$ the face of ∂R^* which upper bounds the j-th dimension, and let $\partial R^*_{j,-}$ be the opposite face which lower bounds the j-th dimension. Let $s_j^\pm = d(c, \partial R^*_{j,\pm})$ be the distance from the center to the appropriate face of R^*. Note that Algorithm 2 with these speeds and this center is equivalent to running the algorithm from the origin and with uniform speeds, after shifting the data so that c lies at the origin and then rescaling each axis by $s_j^\pm$. In this case, R^* is transformed into a ℓ_∞ ball of radius 1 centered at the origin.

By Lemma 4, R^* contains no rejected points with probability at least $1-\alpha$. (Note that the transformations we performed above preserve this fact.) Since the region returned by Algorithm 2 returns a region which contains the largest centered ℓ_∞ ball with no rejected points in it, and R^* is a centered ℓ_∞ ball with no rejected points, we must have $R^* \subseteq \hat{R}$ as desired.

Since we have assumed that R^* is an axis-aligned box, we can write $R^* = \{x \mid \ell_j < x_j < u_j\}$ for some lower and upper bounds ℓ_j, u_j, $j = 1, \ldots, d$. Fix $\varepsilon > 0$ and let

$$\partial R^*_{\varepsilon,j,+} = \{x \mid u_j \le x_j \le u_j + \varepsilon,\ \ell_m < x_m < u_m, m \ne j\}$$
$$\partial R^*_{\varepsilon,j,-} = \{x \mid \ell_j - \varepsilon \le x_j \le \ell_j,\ \ell_m < x_m < u_m, m \ne j\}.$$

(These are just the sets of points which are at most ε "above" the upper dimension j face of R^* and "below" the lower dimension j face of R^*, respectively.) By Assumptions 1-3, there is some constant $c_\varepsilon > 0$ (depending on ε) such that at least $c_\varepsilon n$ points lie in ∂R^*_ε with probability approaching 1 as $n \to \infty$.

Apply the conditions to the $c_\varepsilon n$ points in $\partial R^*_{\varepsilon,j,\pm}$. Chebyshev's inequality implies that

$$\max_{x_i \in \partial R^*_{\varepsilon,j,\pm}} |x_i^\top \hat{\beta} - y_i| \ge \sigma_0 \sqrt{\log c_\varepsilon n} - \frac{\sigma_0}{\sqrt{c_\varepsilon n}} t$$

with probability at least $1 - 1/t^2$. Setting $t = \sqrt{n}$, and since $\sigma_0 > \sigma\sqrt{2}$, for n large enough we have

$$\sigma_0 \sqrt{\log c_\varepsilon n} - \frac{\sigma_0}{\sqrt{c_\varepsilon}} > C\sigma\lambda_{\min}^{-1} \sqrt{\frac{d \log \frac{4d}{\alpha}}{k}} + \sigma \sqrt{2 \log \frac{4n}{\alpha}}$$

provided that $\lambda_{\min}^{-1} k^{-1/2} = o(\log n)$ and for any fixed α, with probability at least $1 - 1/n$. The means that Algorithm 2 will stop growing the $(j, \pm)$ side of $\hat{R}$ at some point in $\partial R^*_{\varepsilon,j,\pm}$. It follows that $\hat{R} \subseteq R^*_\varepsilon$. Since ε was arbitrary (but can be made smaller as $n \to \infty$), this completes the proof. □

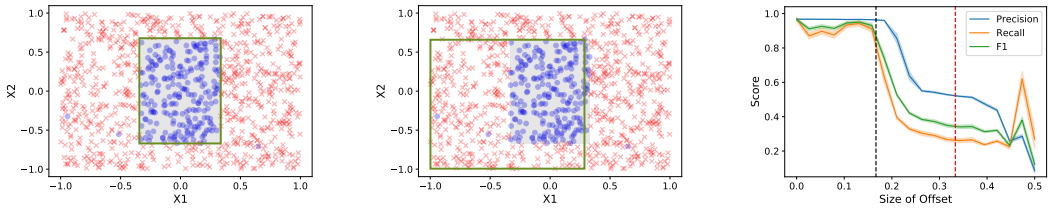
D Experiments cont'd.

D.1 Experiment details for the real datasets

Below, we give a more detailed explanation of the real datasets we used.

1. Brazil Health Dataset [6] is from a longitudinal ecological study for 645 municipalities in the state of São Paulo, Brazil. The study uses a linear model to identify key features for hospitalization of heart failure (HF) and strokes.
2. China Glucose Dataset [30] consists of 5,726 female (F) and 5,457 male (M) Chinese individuals with normal glucose tolerance. The study uses linear model to describe the relationship between fasting plasma glucose and serum uric acid levels (SUA).
3. China HIV Dataset [32] consists of 2,987 participants living with HIV from Guangxi province, China. The study uses linear regression to study how routes of HIV infection affects the HIV internalized stigma scale, adjusted by patients characteristics.
4. Dutch Drinking dataset [3] consists of the individual life survey data of alcohol use among 2,230 Dutch adolescents. The study uses linear regression to analyze how drinking affects adolescents' inhibition (inh), working memory (wm) and shift attention (sha).
5. Korea Grip Dataset [31] is for the Dong-gu study of 2,251 Korean adults with osteoarthritis (OA). The study uses linear regression to explore the associations between grip strength and individual radiographic feature scores of OA.

Experiment setup For the real-world datasets, we randomly split them into training, test and validation set, with ratio 50%, 30% and 20%. In the experiment, we fit the models on the training set with a grid search over hyperparameters and select the region with lowest validation MSE. We then refit the linear model on the training points in the selected region and evaluate its performance on the test set. For DDGroup, the value $\lambda_{\min}$ in (1) was very small in many of the real-world datasets, leading to a very high rejection threshold. Instead, we used a more general form of the threshold $\rho_{\gamma_1,\gamma_2}(x_i) = \sigma\gamma_1\|x_i\| + \sigma\gamma_2$ and tuned γ_1 and γ_2 as additional hyperparameters. Specifically, the algorithm works well by simply setting $\gamma_2 = 0$ and tuning $\gamma_1 \in \{2^{-4}, 2^{-3}, \ldots, 2^5\}$,. We also set the size k of the core group equal to p times the size of the training set, where p was selected from within $\{0.01, 0.05, 0.1, 0.15, 0.2\}$. We also tried two different "speed" settings for Algorithm 2: the sides of the box either grow all at the same rate, or each side grows at a rate proportional to the length of the bounding box in that dimension. For LMT, the tree depth is an important paramter and is scanned from 1 to the dimension of the data for the best performance.



(a) Region selected by DDGroup. (b) Region selected by LMT. (c) Robustness of DDGroup.

Figure 1: Demonstration on synthetic dataset. (a, b) The region selected by (a) DDGroup and (b) linear model tree. The grey shaded area denotes the correct subgroup and the green box corresponds to the learned boundary. Here the depth of LMT is searched from 1 to 10, and the best performance is reported in (b). (c) Robustness of DDGroup to core group misspecification. Shaded region shows standard error of the mean over 50 trials. The black dashed line denotes the point at which "bad" points are included in the core region. The red dashed line denotes the point at which the center of the supplied core set is outside of R. The y-axis records precision, recall, and F1 score (higher is better).

D.2 Evaluation on synthetic data

To visualize our method and test its performance in a well-specified setting, we construct a synthetic dataset where the desired region to be selected is known. Let $B \subseteq \mathbb{R}^d$ be the feature space, and let $R^* \subseteq B$ be the "true" region that we wish to recover. The data are generated as follows. We first sample the features $x \sim \text{Unif}(B)$. If $x \in R^*$, set $y = \beta^\top x + \varepsilon_{\text{in}}$. Else if $x \notin R^*$, set $y = \varepsilon_{\text{out}}$. Here $\beta \neq 0 \in \mathbb{R}^d$ are the fixed true model weights for the region R^*. The error terms ε_{in} and ε_{out} follow $\varepsilon_{\text{in}} \sim \mathcal{N}(0, \sigma_{\text{in}}^2)$ and $\varepsilon_{\text{out}} \sim \mathcal{N}(0, \sigma_{\text{out}}^2)$ with $\sigma_{\text{in}} < \sigma_{\text{out}}$. We set the dimension $d = 3$ so that the selected region can be easily visualized. (The third dimension just allows us to incorporate a bias term, so we will only visualize two dimensions.) We define the bounding box for the features $B = [-1, 1]^2 \times \{1\}$ and the true region $R = [-1/3, 1/3] \times [-2/3, 2/3] \times \{1\}$, and we generate $n = 1000$ data points.

Figure 1a shows the results of running Algorithm 3 on this synthetic data. The gray shaded region is R^*. The red "x" (resp. blue "o") markers denote points that were rejected (resp. not rejected) by the threshold (1), and the red rectangle shows the boundary of $\hat{R}$ returned by DDGroup. There is a nearly perfect overlap between R^* and $\hat{R}$, meaning DDGroup is able to precisely recover the true region. In contrast, the red rectangle in Figure 1b shows the region selected by LMT. While LMT doesnt select a region which contains R^*, it suffers from much lower precision than DDGroup and erroneously selects points outside of R^* as well.

Figure 1c shows the robustness of DDGroup to a misspecified core group. We replace the output of Algorithm 1 with a manually supplied set of points. We start by providing a core group whose center coincides with that of R^*. The x-axis of the plot denotes the offset of this initial core group: at position x on the plot, the center of the core group has been shifted by (x, x). Because we grow the sides of $\hat{R}$ at the same speed, it becomes harder to recover the full R^* when the center of the core group is closer to the edge of R^* (larger x value on the plot). The horizontal dashed black line denotes the point at which the core group starts to include points which do not belong to R^*. The horizontal dashed red line denotes the point at which the center of the core group (and thus the base point from which we grow $\hat{R}$) lies outside of R^*. We see that DDGroup is quite robust to the location of the core group within R^*. However, once "bad" points are included in the core group, the performance (in particular the recall) begins to drop sharply. The precision is more robust to core group misspecification, remaining well above the baseline of 0.22 (which is equivalent to selecting the whole region) even when the core group is more than 50% misspecified.