Improving Decision Sparsity

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sparsity is a central aspect of interpretability in machine learning. Typically, sparsity is measured in terms of the size of a model globally, such as the number of variables it uses. However, this notion of sparsity is not particularly relevant for decision making; someone subjected to a decision does not care about variables that do not contribute to the decision. In this work, we dramatically expand a notion of *decision sparsity* called the *Sparse Explanation Value* (SEV) so that its explanations are more meaningful. SEV considers movement along a hypercube towards a reference point. By allowing flexibility in that reference and by considering how distances along the hypercube translate to distances in feature space, we can derive sparser and more meaningful explanations for various types of function classes. We present cluster-based SEV and its variant tree-based SEV, introduce a method that improves credibility of explanations, and propose algorithms that optimize decision sparsity in machine learning models.

1 Introduction

The notion of *sparsity* is a major focus of interpretability in machine learning and statistical modeling [Tibshirani, 1996, Rudin et al., 2022]. Typically, sparsity is measured *globally*, such as the number of variables in a model, or as the number of leaves in a decision tree. Global sparsity is relevant in many situations, but it is less relevant for individuals subject to the model's decisions. Individuals care less about, and often do not even have access to, the global model. For them *local* sparsity, or **decision sparsity**, meaning the amount of information critical to *their own* decision, is more consequential.

An important notion of decision sparsity been established in the work of Sun et al. [2024], who defined the Sparse Explanation Value (SEV), in the context of binary classification, as the number of factors that need to be changed to a reference feature value in order to change the decision. In contrast to SEV, counterfactual explanations tend not to be *sparse* since they require small changes to many variables in order to reach the decision boundary [Sun et al., 2024]. Instead, SEV provides sparse explanations: consider a loan application that is denied because the applicant has many delinquent trades. In that case, the decision sparsity (that is, the SEV) would be 1 because only a single factor was required to change the decision, overwhelming all possible mitigating factors. The framework of SEV thus allows us to see sparsity of models in a new light.

Prior to this work, SEV had one basic definition: it is the minimal number of features we need to set to their reference values to flip the sign of the prediction. The reference values are typically defined as the mean of the instances in the opposite class. This calculation is easy to understand, but somewhat limiting because the reference could be far in feature space from the point being explained and the explanation could land in a low density area where explanations are not credible. As an example, for loan decisions, SEV could create a counterfactual such as "Changing the applicant's 3-year credit history to 15 years would change the decision." While this counterfactual is valid, faithful, and sparse, if the applicant is only 21 years old, it is *not close* because the distance between the query point and the counterfactual is so large (3 years to 15 years). In addition, this explanation is not *credible* because the proposed changes to the features lead to an unrealistic circumstance – 6-year-olds do not typically have credit. That is, the counterfactual does not represent a typical member of the opposite class. Lack of credibility is a common problem for many counterfactual explanations [Mothilal et al., 2020, Wachter et al., 2017, Laugel et al., 2017, Joshi et al., 2019]. Therefore, in this work, we propose to augment the SEV framework by adding two practical considerations, *closeness* of the reference point to the query, and *credibility* of the explanation, while also optimizing *decision sparsity*.

We propose three ways to create close, sparse and credible explanations. The first way is to create multiple possibilities for the reference, one at the center of each cluster of points (Section 4.1). Having a finite set of references keeps the references *auditable*, meaning that a domain expert can manually check the references prior to generating any explanations. By creating references spread throughout the negative class, queries can be assigned to closer references than before. Second, we allow the references to be flexible, where their position can be shifted slightly from a central location in order to reduce the SEV (Section 4.4). The third way pertains to decision tree classifiers, where a reference point is placed on each opposite-class leaf, and an efficient shortest-path algorithm is used to find the nearest reference (Section 4.2). Table 1 shows a query at the top, and some SEV calculations from our methods below, showing feature values that were changed within the explanation.

Table 1: An example for a query in the FICO Dataset with different kinds of explanations, SEV^1 represents the SEV calculation with one single reference using population mean, SEV$^{©}$ represents the cluster-based SEV, SEV^F represents the flexible-based SEV. The columns are four features.

	EXTERNAL	NUMSATIS-	NETFRACTION	PERCENTTRADES
	RISKESTIMATE	FACTORYTRADES	REVOLVINGBURDEN	NEVERDELQ
Query	69.00	10.00	117.01	90
SEV^1	**72.65**	**21.47**	**22.39**	90
SEV^F	**78.00**	10.00	**9.00**	90
SEV$^{©}$	**81.00**	**26.00**	**12.00**	90
SEV^T	69.00	10.00	117.01	**100**

In addition to developing methods for calculating SEV, we propose two algorithms to optimize a machine learning model to reduce the number of points that have high SEV without sacrificing predictive performance in Section 5, one based on gradient optimization, and the other based on search. The search algorithm is exact. It uses an exhaustive enumeration of the set of accurate models to find one with (provably) optimal SEV.

Our notions of decision sparsity are general and can be used for any model type, including neural networks and boosted decision trees. Decision sparsity can benefit any application where individuals are subject to decisions made from predictive models – these are cases where decision sparsity is more important than global sparsity.

2 Related Work

The concept of SEV revolves around finding models that are simple, in that the explanations for their predictions are sparse, while recognizing that different predictions can be simple in different ways (i.e., involving different features). In this way, it relates to (i) globally sparse models, (ii) local classification methods, which predict the outcomes of different units using local models, and (iii) black box explanation methods, which seek to explain predictions of complex models. We further comment on these below.

Instance-wise Explanations. Prior work has developed methods to explain predictions of black boxes [e.g., Guidotti et al., 2018, Ribeiro et al., 2016a, 2018, Lundberg and Lee, 2017, Baehrens et al., 2010] for individual instances. These explanations are designed to estimate importance of features, are not necessarily faithful to the model, and are not associated with sparsity in decisions, so they are fairly distant from the purpose of the present work. Our work is on tabular data; there is a multitude of unrelated work on explanations for images [e.g., Apicella et al., 2019, 2020] and text [e.g., Lei et al., 2016, Li et al., 2016, Treviso and Martins, 2020, Bastings et al., 2019, Yu et al., 2019, 2021]. More closely related are *counterfactual explanations*, also called inverse classification [e.g., Mothilal et al., 2020, Wachter et al., 2017, Lash et al., 2017, Sharma et al., 2022, Virgolin and Fracaros, 2023, Guidotti et al., 2019, Poyiadzi et al., 2020, Russell, 2019, Boreiko et al., 2022, Laugel et al., 2017, Pawelczyk et al., 2020]. Counterfactual explanations are typically designed to find the closest instance to a query point with the opposite prediction, without considering sparsity of the explanation. However, extensive experiments [Delaney et al., 2023] indicate that these "closest counterfactuals" tend to be unnatural for humans because the decision boundary is typically in a region where humans have no intuition for why a point belongs to one class or the other. For SEV, on the other hand, reference values represent the population commons, so they are intuitive. Thus, SEV has two advantages over standard counterfactuals: its references are meaningful because they represent population commons, and its explanations are *sparse*.

Local Sparsity Optimization Models While there are numerous prior works on developing post-hoc explanations, limited attention has been paid to developing models that provide sparse explanations. We are aware of only one work on this, namely the Explanation-based Optimization (ExpO) algorithm of Plumb et al. [2020] that used a neighborhood-fidelity regularizer to optimize the model to provide sparser post-hoc LIME explanations. Experiment in Appendix K in our paper shows that ExpO is both slower and provides less sparse predictions than our algorithms.

3 Preliminaries and Motivation

The Sparse Explanation Value (SEV) is defined to measure the sparsity of individual predictions of binary classifiers. The point we are creating an explanation for is called the *query*. The SEV is the smallest set of feature changes from the query to a reference that can flip the prediction of the model. When we make a change to the query's feature, we *align* it to be equal to that of the reference point. The reference point is a "commons," i.e., a prototypical point of the opposite class as the query. In this section, we will focus on the basic definition of SEV, the selection criteria for the references, as well as three reference selection methods.

3.1 Recap of Sparse Explanation Values

We define SEV following Sun et al. [2024]. For a specific binary classification dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, with each $\boldsymbol{x}_i \in \mathbb{R}^p$, and the outcome of interest is $y_i \in \{0,1\}$. (This can be extended to multi-class classification by providing counterfactuals for every other class than the query's class.) We predict the outcome using a classifier $f : \mathcal{X} \to \{0,1\}$.

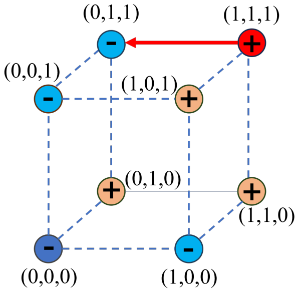


Figure 1: SEV Hypercube

Without loss of generality, in this paper, we are only interested in queries predicted as positive (class 1). We focus on providing a sparse explanation from the query to a *reference* that serves as a population commons, denoted $\boldsymbol{r}$. Human studies [Delaney et al., 2023] have shown that contrasting an instance with prototypical instances from another class provides more intuitive explanations than comparing it with instances from the same class. Thus, we define our references in the opposite class (negative class in this paper). To calculate SEV, we will align (i.e., equate) features from query $\boldsymbol{x}_i$ and reference $\tilde{\boldsymbol{x}}$ one at a time, checking at each time whether the prediction flipped. Thinking of these alignment steps as binary moves, it is convenient to represent the 2^p possible different alignment combinations as vertices on the boolean hypercube. The hypercube is defined below:

Definition 3.1 (SEV hypercube)**.** A SEV hypercube $\mathcal{L}_{f,i,\boldsymbol{r}}$ for a model f, an instance $\boldsymbol{x}_i$ with label $f(\boldsymbol{x}_i) = 1$, and a reference $\boldsymbol{r}$, is a graph with 2^p vertices. Here p is the number of features in $\boldsymbol{x}_i$ and $\boldsymbol{b}_v \in \{0,1\}^p$ is a Boolean vector that represents each vertex. Vertices u and v are adjacent when their Boolean vectors differ in one bit, $\|\boldsymbol{b}_u - \boldsymbol{b}_v\|_0 = 1$. 0's in $\boldsymbol{b}_v$ indicate the corresponding features are aligned, i.e., set to the feature values of the reference $\boldsymbol{r}$, while 1's indicate the true feature value of instance i. Thus, the actual feature values represented by the vertex v is $\boldsymbol{x}_i^{\boldsymbol{r},v}, := \boldsymbol{b}_v \odot \boldsymbol{x}_i + (\boldsymbol{1} - \boldsymbol{b}_v) \odot \boldsymbol{r}$, where $\odot$ is the Hadamard product. The score of vertex v is $f(\boldsymbol{x}_i^{\boldsymbol{r},v})$, also denoted as $\mathcal{L}_{f,i,\boldsymbol{r}}(\boldsymbol{b}_v)$.

The SEV hypercube definition can also be extended from a hypercube to a Boolean lattice as they have the same geometric structure. There are two variants of the Sparse Explanation Value: one gradually aligns the query to the reference (SEV$^-$), and the other gradually aligns the reference to the query (SEV$^+$). In this paper, we focus on SEV$^-$:

Table 2: Calculation process for SEV$^-$ = 1

	Type	Housing	Loan	Education	Y(Risk)
(1,1,1)	query	Rent	>10k	High School	High
(0,1,1)	SEV$^-$ Explanation	**Owning**	>10k	High School	**Low**
(0,0,0)	reference	Owning	<5k	Master	Low

Definition 3.2 (SEV$^-$)**.** For a positively-predicted query $\boldsymbol{x}_i$ (i.e., $f(\boldsymbol{x}_i) = 1$), the Sparse Explanation Value Minus (SEV$^-$) is the minimum number of features in the query that must be aligned to reference $\boldsymbol{r}$ to elicit a negative prediction from f. It is the length of the shortest path along the hypercube to obtain a negative prediction,

$$\text{SEV}^-(f, \boldsymbol{x}_i, \boldsymbol{r}) := \min_{\boldsymbol{b} \in \{0,1\}^p} \quad \|\boldsymbol{1} - \boldsymbol{b}\|_0 \quad \text{s.t.} \quad \mathcal{L}_{f,i,\boldsymbol{r}}(\boldsymbol{b}) = 0.$$

Figure 1 and Table 2 shows an example of SEV$^-$=1 in a credit risk evaluation setting. Since $p = 3$, we construct a SEV hypercube with $2^3 = 8$ vertices. The red vertex $(1, 1, 1)$ corresponds to the query. The dark blue vertex at $(0, 0, 0)$ represents the negatively-predicted reference value. The orange vertices are predicted to be positive, and the light blue vertices are predicted to be negative. To compute SEV$^-$, we start at $(1, 1, 1)$ and find the shortest path to a negatively-predicted vertex. On this hypercube, $(0, 1, 1)$ is closest. Translating this to feature space, this means that if the query's housing situation changes from renting to the reference value "owning," it would be predicted as negative. This means that **SEV$^-$ is equal to 1** in this case. The feature vector corresponding to this closest vertex $(0, 1, 1)$, is called the **SEV$^-$ explanation** for the query, denoted by $\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}}$ for reference $\boldsymbol{r}$.

3.2 Motivation of Our Work: Sensitivity to Reference Points

Since SEV$^-$ is determined by the path on a SEV hypercube and each hypercube is determined by the reference point, the SEV$^-$ is therefore sensitive to the selection of reference points. Adjusting the reference point trades off between *sparsity* (according to SEV$^-$) and *closeness* (measured by ℓ_2, ℓ_∞ or ℓ_0 distance between the query and its assigned reference point). Note that this trade-off exists because SEV$^-$ tends to be small when the reference is far from the query. More detailed explanations, visualizations, and experiments are shown in Appendix B.

Selecting References. The reference must represent the commons, meaning the negative population, and the generated explanations should represents the negative populations as well. Moreover, the negative population may have subpopulations; e.g., Diabetes patients may have higher blood glucose levels, while hypertension patients have higher blood pressure. To have meaningful coverage of the negative population, in this work, we consider *multiple* references, placed *within the various subpopulations*. This allows each point in the positive population to be closer to a reference. Let $\mathcal{R}$ denote possible placements of references. For query $\boldsymbol{x}_i$, an individual-specific reference $\boldsymbol{r}_i \in \mathcal{R}$ for $\boldsymbol{x}_i$ is chosen based on three criteria: it should be nearby (i.e., close), and should provide a sparse and reasonable explanation. That is, we are looking to minimize the following three objectives over placement of the reference $\boldsymbol{r}_i$:

$$\|\boldsymbol{x}_i - \boldsymbol{r}_i\|, \boldsymbol{r}_i \in \mathcal{R} \quad \text{(Closeness)} \tag{1}$$

$$\text{SEV}^-(f, \boldsymbol{x}_i, \boldsymbol{r}_i), \boldsymbol{r}_i \in \mathcal{R} \quad \text{(Sparsity)} \tag{2}$$

$$-P(\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i} | X^-) \quad \text{(Negated Credibility)}, \tag{3}$$

with the constraint that the references obey auditability, meaning that domain experts are able to check the references manually, or construct them manually. The function SEV$^-(f, \boldsymbol{x}_i, \boldsymbol{r}_i)$ in (2) represents the SEV$^-$ computed with the given function f, query $\boldsymbol{x}_i$, and the individual-specific reference $\boldsymbol{r}_i$ for generating the hypercube, $\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i}$ is the sparse explanation for the sample $\boldsymbol{x}_i$, and $P(\cdot|X^-)$ in the definition of credibility represents the probability density distribution of the negative population and $P(\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i}|X^-)$ is the density of the negative distribution at $\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i}$. If $P(\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i}|X^-)$ is large, $\boldsymbol{x}_i^{\text{expl},\boldsymbol{r}_i}$ is in a high-density region.

4 Meaningful and Credible SEV

We now describe cluster-SEV, which improves closeness at the expense of SEV, and its variant, tree-based SEV, which improves all three objectives and computational efficiency. We also present methods to improve the credibility and sparsity of the explanations.

4.1 Cluster-based SEV: Improving Closeness

This approach creates multiple references for the negative population. A clustering algorithm is used to group negative samples, and the resulting cluster centroids are assigned as references. A query is assigned to its closest cluster center:

$$\tilde{\boldsymbol{r}}_i \in \arg\min_{\boldsymbol{r}\in\mathcal{C}} \|\boldsymbol{x}_i - \boldsymbol{r}\|_2$$

where $\mathcal{C}$ is the collection of centroids obtained by clustering the negative samples. We refer to the SEV$^-$ produced by the grouped samples as cluster-based SEV, denoted SEV$^{©}$. Figure 2 illustrates the calculation of SEV$^{©}$ for two

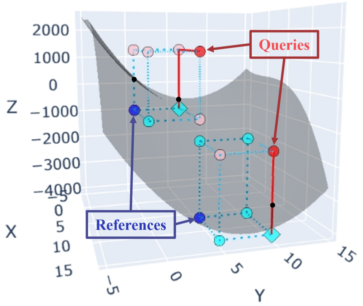


Figure 2: Cluster-based SEV

examples located in two different centroids. A red dot represents a query, while a blue dot represents a reference. For each instance, it selects the closest centroid and considers the SEV hypercube, where each cyan point represents a negatively predicted vertex and each pink point represents a positively predicted vertex. We deduce by following the red lines that the SEV© for the two queries are 2 and 1, respectively. The cluster centroids should serve as a cover for the negative class. To ensure that the cluster centroids have negative predictions, we use the soft clustering method of Bezdek et al. [1984] to constrain the predictions of the cluster centers. Details are in Appendix C.

4.2 Tree-based SEV: SEV© Variant with Useful Properties and Computational Benefits

Tree-based SEV is a special case of cluster-based SEV, where we consider each negative leaf as a reference candidate, and and find the sparsest explanation (path along the tree) to the nearest reference. Here, SEV$^-$ and ℓ_0 distance (i.e., edit distance) are equivalent. That is, we find the minimum number of features to change in order to achieve a negative prediction.

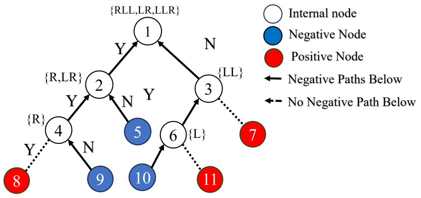


Figure 3: SEV^T Preprocessing

We denote SEV^T as the SEV$^-$ calculated based on this process. Here, we assume that trees have no trivial splits where all child leaves make the same prediction. If so, we would collapse those leaves before calculating the SEV^T. The first theorem below refers to decision paths that have negatively predicted child leaves. This is where taking one different choice at an internal split leads to a negative leaf.

Theorem 4.1. *With a single decision classifier DT and a positively-predicted query $\boldsymbol{x}_i$, define N_i as the leaf that captures it. If N_i has a sibling leaf, or any internal node in its decision path has a negatively-predicted child leaf, then SEV^T is* ***equal to 1***.

The second theorem states that SEV$^-$ and minimum edit distance from the query to negative leaves are equivalent.

Theorem 4.2. *With a single decision tree classifier DT and a positively-predicted query $\boldsymbol{x}_i$, with the set of all negatively predicted leaves as reference points, both SEV$^-$ and the ℓ_0 distance (edit distance) between the query and the SEV$^-$ explanation are minimized.*

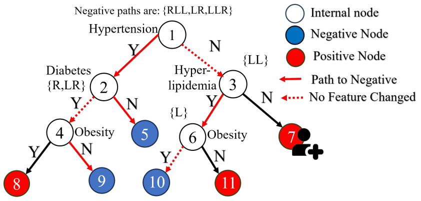


Figure 4: Efficient SEV^T calculation: Query (node ⑦) has SEV^T=1, which goes to node ⑩. The path to this node is recorded as LL at node ③, which is along the decision path to node ⑦.

The proofs of those two theorems are shown in Appendix L and M. The structure of tree models yields an extremely efficient way to calculate SEV$^-$. We perform an important preprocessing step before any SEV$^-$ calculations are done, which will make SEV$^-$ easier to calculate for all queries at runtime. At each internal node, we record all paths to negative leaves anywhere below it in the tree. This is described in Algorithm 2 in Appendix E. E.g., if the tree has binary splits, a path from an internal node to a leaf node might require us to go left, then right, then left. In that case, we store LRL on this internal node to record this path. Then, when a query arrives at runtime (in a positive leaf, since it has a positive prediction), we traverse directly up its decision path all the way to the root node. For all internal nodes in the decision path, we observe distances to each negative leaf, which were stored during preprocessing. We traverse each of these, and the minimum distance among these is the SEV$^-$. This is described in Algorithm 3 in Appendix E and illustrated in Figure 4. Note that we actually would traverse to each negative node because some internal decisions might not need to be changed along the path. In the example in Figure 4, we change the split at node ③, and use the value that the query already has for the split at node ⑥, landing in node ⑩, so SEV$^-$ is 1 not 2.

Table 3 walks through the calculation again, using the names of the features (hypertension, diabetes, etc.). On the first action line, the decision path to the query is ③→⑥→⑩. That means we check ① and ③ for negative paths, yielding path LL. We flip node ③ (change Hyperlipidemia

Table 3: Illustration of SEV^T calculation.

	Action	Hyper-tension	Diabetes	Hyper-lipidemia	Obesity	Have Stroke	# of changed condition (SEV)
Instance ①→③→⑦	**Check** **node ①&③**	No	Yes	No	Yes	Yes ⑦	
Flip at **node ③**	**Check LL** ③→⑥→⑩	No		Yes Flip at ③	Yes (Unchanged)	No ⑩	1
Flip at **node ①**	**Check LR** ②→⑤	Yes Flip at ①	No Flip at ②			No ⑤	2
	Check LLR ②→④→⑨	Yes Flip at ①	Yes (Unchanged)		No Flip at ④	No ⑨	2

to ‘yes’) and follow the LL path. We do not change Obesity to get to the negative node, so we record the SEV^T as 1 in that row. In our implementation, we simply stop when we reach an SEV^T=1 solution, but we will continue in order to illustrate how the calculation works. We go up to node ① and repeat the process for the LR and LLR paths. Those both have SEV^T=2.

4.3 Improving Credibility for All SEV Calculations

As we mentioned in Section 3.2, the credibility objective encourages explanations to be located in high-density region of the negative population. Previous SEV$^-$ definitions focus on sparsity and closeness objectives, but did not consider credibility. It is possible to increase credibility easily while constructing an explanation: if the explanation veers out of the high-density region, we continue walking along the SEV hypercube during SEV calculations. Specifically, we continue moving towards the reference until the vertex is in a high-density region. Since the reference is in a high-density region, walking towards it will eventually lead to a high-density point. The tree-based SEV explanations automatically satisfy high credibility:

Theorem 4.3. *With a single sparse decision tree classifier DT with support at least S in each negative leaf, the SEV^T explanation for query $\boldsymbol{x}_i$ always satisfies credibility at least $\frac{S}{N^-}$, where N^- is the total number of negative samples.*

This theorem can be easily proved because SEV$^-$ explanations generated by SEV^T are always the negative leaf nodes (which are the references), and the references are located in regions with support at least S by assumption.

4.4 Flexible Reference SEV: Improving Sparsity

From Section 3.2, we know that queries further from the decision boundary tend to have lower SEV$^-$. Based on this, we introduce Flexible Reference SEV (denoted SEV^F), which moves the reference value slightly in order to achieve a lower value of the model output $f(\tilde{\boldsymbol{r}})$, which, in turn, is likely to lead to lower SEV$^-$. Consider a given reference $\tilde{\boldsymbol{r}}$, and the decision function for classification $f(\cdot)$, the optimization for finding the optimal reference is: $\boldsymbol{r}^* \in \arg\min_{\boldsymbol{r}} f(\boldsymbol{r}) \quad \text{s.t}\|\boldsymbol{r}-\tilde{\boldsymbol{r}}\|_\infty \leq \epsilon_F$ where the $\arg\min$ is over reference candidates that are near the original reference value $\tilde{\boldsymbol{r}}$. The flexibility threshold ϵ_F represents the flexibility allowed for moving the reference within a ball. We limit flexibility so the explanation stays meaningful. Since it is impractical to explore all potential combinations of feature-value candidates, we address this problem by marginalizing. Specifically, we optimize the reference over each feature independently. The detailed algorithm for calculating Flexible Reference SEV, denoted SEV^F, is shown in Algorithm 1 in Appendix D. In Section 6.2, we show that moving the reference slightly can sometimes reduce the SEV, improving sparsity.

5 Optimizing Models for SEV$^-$

Above, we showed how to calculate SEV$^-$ for a fixed model. In this section, we describe how to train classifiers that optimize the average SEV$^-$ without loss in predictive performance. We propose two methods: minimizing an easy-to-optimize surrogate objective (Section 5.1) and searching for models with the smallest SEV from a “Rashomon set” of equally-good models (Section 5.2). In what follows, we assume that SEV$^-$ was calculated prior to optimization, that reference points were assigned to each query, and that these assignments do not change throughout the calculation.

5.1 Gradient-based SEV Optimization

Since we want to minimize expected test SEV$^-$, the most obvious approach would be to choose our model f to minimize average training SEV$^-$. However, since SEV calculations are not differentiable and they are combinatorial in the number of features and data points, this would be intractable. Following Sun et al. [2024], we instead design the optimization objective to penalize each sample where SEV$^-$ is more than 1. Thus, we propose the loss term:

$$\ell_{\text{SEV\_All\_Opt}-}(f) := \frac{1}{n^+}\sum_{i=1}^{n^+} \max\left(\min_{j=1,\dots,p} f((\mathbf{1}-\boldsymbol{e}_j)\odot\boldsymbol{x}_i+\boldsymbol{e}_j\odot\tilde{\boldsymbol{r}}_i),\ 0.5\right),$$

where $\boldsymbol{e}_j$ is the vector with a 1 in the j^{th} coordinate and 0’s elsewhere, n^+ is the number of queries, and the reference point $\tilde{\boldsymbol{r}}_i$ is specific to query $\boldsymbol{x}_i$ and chosen beforehand. Intuitively, $f((\mathbf{1}-\boldsymbol{e}_j)\odot\boldsymbol{x}_i+\boldsymbol{e}_j\odot\tilde{\boldsymbol{r}}_i)$ is the function value of query $\boldsymbol{x}_i$ where its feature j has been replaced with the reference’s feature j. $\min_{j=1,\dots,p} f((\mathbf{1}-\boldsymbol{e}_j)\odot\boldsymbol{x}_i+\boldsymbol{e}_j\odot\tilde{\boldsymbol{r}}_i)$ chooses the variable to replace that most reduces the function value. If the SEV$^-$ is 1, then when this replacement is made, the point now is on the negative side of the decision boundary and f is less than 0.5, in which case the $\max$ chooses 0.5. If SEV$^-$ is more than 1, then after replacement, f will still predict positive and be more than 0.5, in which case, its value will contribute to the loss. This loss is differentiable with respect to model parameters except at the "corners" and not difficult to optimize.

To put these into an algorithm, we optimize a linear combination of different loss terms,

$$\min_{f \in \mathcal{F}} \ell_{\text{BCE}}(f) + C_1 \ell_{\text{SEV\_All\_Opt}-}(f) \tag{4}$$

where ℓ_{BCE} is the Binary Cross Entropy Loss to control the accuracy of the training model and $\mathcal{F}$ is a class of classification models that estimate the probability of belonging to the positive class. $\ell_{\text{SEV\_All\_Opt}-}$ is the loss term that we have just introduced above. C_1 can be chosen using cross-validation. We define **All-Opt**$^-$ as the method that optimizes (4). Our experiments show that this method is not only effective in shrinking the average SEV$^-$ but often attains the minimum possible SEV$^-$ value of 1 for most or all queries.

5.2 Search-based SEV Optimization

As defined in Section 4.2, our goal is to find a model with the lowest average SEV$^-$ among classification models with the best performance.

The Rashomon set [Semenova et al., 2022, Fisher et al., 2019] is defined as the set of all models from a given class with performance approximately that of the best-performing model. The first method that stores the entire Rashomon set of any nontrivial function class is called TreeFARMS [Xin et al., 2022], which stores all good sparse decision trees in a data structure. TreeFARMS allows us to optimize multiple objectives over the space of sparse trees easily by enumeration of the Rashomon set to find all accurate models, and a loop through the Rashomon set to optimize secondary objectives. We use TreeFARMS and search through the Rashomon set for a model with the lowest average SEV$^-$:

$$\min_{f \in \mathcal{R}_{\text{set}}} \frac{1}{n^+} \sum_{i=1}^{n^+} \text{SEV}^T(f, \boldsymbol{x}_i),$$

where the Rashomon set is $\mathcal{R}_{\text{set}}$, and where we use SEV^T as the SEV$^-$ for each sparse tree in the Rashomon set. Recall that Algorithms 2 and 3 show how to calculate SEV^T. We call this search-based optimization as **TOpt**.

6 Experiments

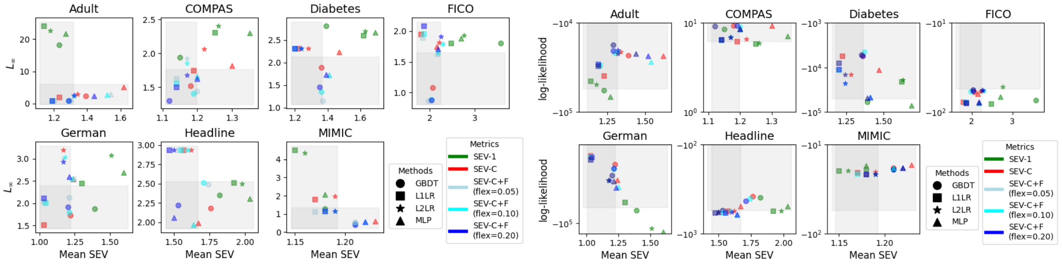
Training Datasets To evaluate whether our proposed methods would achieve sparser, more credible and closer explanations, we present experiments on seven datasets: (i) UCI Adult Income dataset for predicting income levels [Dua and Graff, 2017], (ii) FICO Home Equity Line of Credit Dataset for assessing credit risk, used for the Explainable Machine Learning Challenge [FICO, 2018], (iii) UCI German Credit dataset for determining creditworthiness [Dua and Graff, 2017], (iv) MIMIC-III dataset for predicting patient outcomes in intensive care units [Johnson et al., 2016a,b], (v) COMPAS dataset [Jeff Larson and Angwin, 2016, Wang et al., 2022a] for predicting recidivism, (vi) Diabetes dataset [Strack et al., 2014] for predicting whether patients will be re-admitted within two years, and (vii) Headline dataset for predicting whether the headline is likely to be shared by readers [Chen et al., 2023a]. Additional details on data and preprocessing are in Appendix A.

Training Models For SEV$^©$, we trained four baseline binary classifiers: (i, ii) logistic regression classifiers with ℓ_1 (L1LR) and ℓ_2 (L2LR) penalties, (iii) a gradient boosting decision tree classifier (GBDT), and (iv) a 2-layer multi-layer perceptron (MLP), and tested its performance with SEV^F added, and the credibility rules added. In addition, we trained All-Opt$^-$ variants of these models in which the SEV penalties described in the previous sections are implemented. For SEV^T methods, we compared tree-based models from CART, C4.5, and GOSDT [Lin et al., 2020] with the TOpt method proposed in Section 5.2. Details on training the methods is in Appendix F.

Evaluation Metrics To evaluate whether good references are selected for the queries, we evaluate sparsity and closeness (i.e., similarity of query to reference). For **sparsity**, we use the average number of feature changes (which is the same as ℓ_0 norm) between the query and the explanation; for **closeness**, we use the median ℓ_∞ norm between the generated explanation and the original query as the metric for SEV$^©$. For tree-based models, we use only SEV^T as the metric since SEV^T and ℓ_0 norm are equivalent; for **credibility**, we trained a Gaussian mixture model on the negative samples of each dataset, and used the mean log-likelihood of the generated explanations as the metric.

6.1 Cluster-based SEV shows improvement in credibility and closeness

Let us show that SEV$^{©}$ provides improved explanations. Here, we calculated the metric for different SEV$^{©}$ variants, SEV$^{©}$ and SEV$^{©+F}$(SEV$^{©}$ with flexible reference), and compared to the original SEV^{1}, where SEV^{1} is defined as the SEV^{-} calculation with single reference generated by the mean value of each numerical feature and mode value of each categorical feature of the negative population, as done in the original SEV paper [Sun et al., 2024] under various datasets and models.



(a) Sparsity (SEV^{-}) and Closeness (L$_\infty$)

(b) Sparsity (SEV^{-}) and Credibility (log-likelihood)

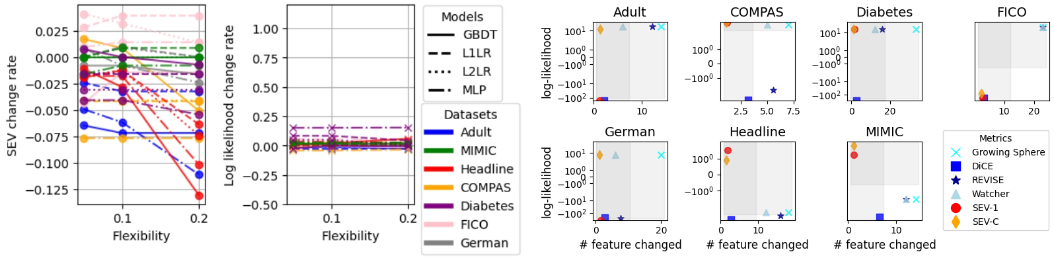
Figure 5: Explanation performance under different models and metrics. We desire lower SEV^{-} for sparsity, lower ℓ_∞ for closeness and higher log likelihood for credibility (shaded regions)

Figure 5a shows the relationship between spasity and variants, the scatter plot between mean SEV^{-} and mean ℓ_∞ for each explanation generated by different variants. We find that **SEV$^{©}$ improves closeness**, which was expected since the references were designed to be closer to the queries. Interestingly, SEV$^{©}$ sometimes has lower decision sparsity than SEV^{1}. SEV$^{©}$ was designed to trade off SEV^{-} for closeness, so it is surprising that it sometimes performs strictly better on both metrics, particularly for the COMPAS, Diabetes, and German Credit datasets.

Interestingly, we also find that even though we do not optimize credibility for our model, Figure 5b shows that SEV$^{©}$ improves credibility, particularly for the Adult, German, and Diabetes datasets by plotting the relationship between mean SEV^{-} and mean log-likelihood of the generated explanations. It is reasonable since the references are the cluster centroids for the negative samples, so the explanations are more likely to be located in the same high-density area. More detailed values for those methods and metrics are shown in Appendix H.

6.2 Flexible Reference SEV can improve sparsity without losing credibility

In Section 4.4, we proposed the flexible reference method for sparsifying SEV^{-} explanations, which moves the reference slightly away from the decision boundary. The blue points in Figure 5a and 5b have already shown that with small modification of the reference, the credibility of the explanations is not affected. Figure 6a shows how SEV^{-} and credibility change as we increase flexibility; SEV^{-} sometimes substantially decreases while credibility is maintained.



(a) SEV^{-}/Credibility change rate for varying flexibility

(b) Median Log likelihood and # of features changed

Figure 6: (a) Sparsity and Credibility as a function of the change of flexibility level (0 to 5%/10%/20%) under different models and datasets (b) The median log-likelihood and number of features within different counterfactual explanations. Points at the upper left corner are desired.

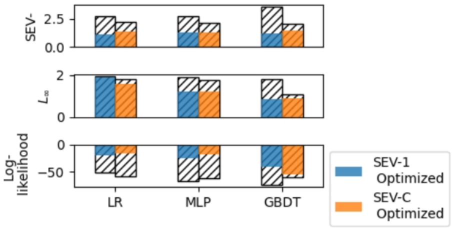
6.3 SEV^{-} provides the sparsest explanation compared to other counterfactual explanations

Recall that SEV^{-} flips features of the query to values of the population commons. This can be viewed as a type of counterfactual explanation, though typically, counterfactual explanations aim to find the minimal distance from one class to another. In this experiment, we compare the sparsity of SEV$^-$ calculations to that of baseline methods from the literature on counterfactual explanations, namely Watcher [Wachter et al., 2017], REVISE [Joshi et al., 2019], Growing Sphere [Laugel et al., 2017], and DiCE [Mothilal et al., 2020].

6.4 All-Opt$^-$ and TOpt optimize SEV$^-$, preserving model performance, explanation closeness and credibility

Even without optimization, our SEV$^-$ variants improve decision sparsity and/or closeness. If we are willing to retrain the prediction model as discussed in Section 5, we can improve these metrics further, creating accurate models with higher decision sparsity. Figure 7a shows that gradient-based SEV optimization can reduce the SEV without harming the closeness metric (ℓ_∞) and the credibility metrics (log-likelihood). The slashed bars represents the SEV$^-$ and ℓ_∞ metrics before optimization using different models, while the colored bars are the results after optimizing with All-Opt$^-$. We have also compared our results with ExpO [Plumb et al., 2020], which is a optimization method that maximizes the mean neighborhood fidelity of the queries, but we have found that explanations are not sparse, and it requires long training times; the detailed results are shown in Appendix K.

Figure 6b shows sparsity and credibility performance of all counterfactual explanation methods on different datasets under ℓ_2 logistic regression (other information, including ℓ_∞ norms for counterfactual explanation methods, is in Appendix G). All SEV variants are in warm colors, while competitors are in cool colors. SEV$^-$ methods have the sparest explanations, followed by DiCE. (A comparison of SEV$^-$ to DiCE is provided by Sun et al. [2024].) We point out that this comparison was made on methods that were not designed to optimize explanation sparsity. Importantly, sparsity is essential for human understanding [Rudin et al., 2022]. Moreover, it has been shown that SEV (especially SEV$^©$) would have more credible explanations than competitors, while explanations remain sparse.



(a) All-Opt$^-$ Performance

	Train Acc	Test Acc	Mean SEV^T
CART	0.71 ± 0.01	0.71 ± 0.01	1.10 ± 0.03
C4.5	0.71 ± 0.01	0.71 ± 0.01	1.13 ± 0.05
GOSDT	0.70 ± 0.01	0.70 ± 0.01	1.08 ± 0.02
TOpt	0.70 ± 0.01	0.70 ± 0.01	1.00 ± 0.02

(b) SEV^T performance on different tree-based models

Figure 7: (a) SEV$^-$ and ℓ_∞ before and after All-Opt$^-$ on the FICO Dataset. Slashed bars are before, solid color is after. (b) All tree-based models with similar accuracy have low SEV^T.

For the Tree-based SEV, we have applied the efficient computation procedure to different kinds of tree-based models, and compared them with the search-based optimization method (TOpt) for trees in Section 5. The search-based algorithm works perfectly in finding a good model without performance loss. It achieves a perfect average SEV score of 1.00.

Conclusion

Decision sparsity can be more useful than global model sparsity for individuals, as individuals care less about, and often do not even have access to, the global model. We presented approaches to achieving high decision sparsity, closeness and credibility, while being faithful to the model. One limitation of our method is that causal relationships may exist among features, invalidating certain transitions across the SEV hypercube. This can be addressed by searching across vertices that do not satisfy the causal relationship, though it requires knowledge of the causal graph. Another limitation is that to make the explanation more credible, the threshold to stop searching the SEV hypercube is not easy to determine. Future studies could focus on on these topics. Overall, our work has the potential to enhance a wide range of applications, including but not limited to loan approvals and employment hiring processes. Improved SEV translates directly into explanations that simply make more sense to those subjected to the decisions of models.

References

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 1996.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16(none):1 – 85, 2022. doi: 10.1214/21-SS133. URL `https://doi.org/10.1214/21-SS133`.

Yiyang Sun, Zhi Chen, Vittorio Orlandi, Tong Wang, and Cynthia Rudin. Sparse and faithful explanations without sparse models. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2024.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31:841, 2017.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017.

Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Pedreschi Dino. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5), 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, 2016a.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, aug 2010. ISSN 1532-4435.

A. Apicella, F. Isgrò, R. Prevete, and G. Tamburrini. Contrastive explanations to classification systems using sparse dictionaries. In *Lecture Notes in Computer Science*, pages 207–218. Springer International Publishing, 2019. doi: 10.1007/978-3-030-30642-7_19.

A. Apicella, F. Isgrò, R. Prevete, and G. Tamburrini. Middle-level features for the explanation of classification systems by sparse dictionary methods. *International Journal of Neural Systems*, 30 (08):2050040, July 2020. doi: 10.1142/s0129065720500409.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, 2016.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

Marcos Treviso and André FT Martins. The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118, 2020.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables, 2019.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi S Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. *arXiv preprint arXiv:1910.13294*, 2019.

Mo Yu, Yang Zhang, Shiyu Chang, and Tommi Jaakkola. Understanding interlocking dynamics of cooperative rationalization. *Advances in Neural Information Processing Systems*, 34:12822–12835, 2021.

Michael T Lash, Qihang Lin, Nick Street, Jennifer G Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2017.

Shubham Sharma, Alan H Gee, Jette Henderson, and Joydeep Ghosh. Faster-ce: Fast, sparse, transparent, and robust counterfactual explanations. *arXiv preprint arXiv:2210.06578*, 2022.

Marco Virgolin and Saverio Fracaros. On the robustness of sparse counterfactual explanations to adverse perturbations. *Artificial Intelligence*, 316:103840, 2023.

Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019. doi: 10.1109/MIS.2019.2957223.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 344–350, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375850. URL `https://doi.org/10.1145/3375627.3375850`.

Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.

Valentyn Boreiko, Maximilian Augustin, Francesco Croce, Philipp Berens, and Matthias Hein. Sparse visual counterfactual explanations in image space. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 133–148. Springer, 2022.

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of the web conference 2020*, pages 3126–3132, 2020.

Eoin Delaney, Arjun Pakrashi, Derek Greene, and Mark T Keane. Counterfactual explanations for misclassified images: How human and machine explanations differ. *Artificial Intelligence*, 324: 103995, 2023.

Gregory Plumb, Maruan Al-Shedivat, Ángel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020.

James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1827–1858, 2022.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole rashomon set of sparse decision trees. *Advances in Neural Information Processing Systems*, 35:14071–14084, 2022.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

FICO. Explainable machine learning challenge, 2018. URL https://community.fico.com/s/explainable-machine-learning-challenge. Accessed: 2018-11-02.

A. Johnson, T. Pollard, and R. Mark. MIMIC-III clinical database. https://physionet.org/content/mimiciii/, 2016a.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1), May 2016b. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

Lauren Kirchner Jeff Larson, Surya Mattu and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm, 2016. Accessed: 2023-02-01.

Caroline Wang, Bin Han, Bhrij Patel, and Cynthia Rudin. In Pursuit of Interpretable, Fair and Accurate Machine Learning for Criminal Recidivism Prediction. *Journal of Quantitative Criminology*, pages 1–63, 2022a. ISSN 0748-4518. doi: 10.1007/s10940-022-09545-w.

Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70, 000 clinical database patient records. *BioMed Research International*, 2014:1–11, 2014. doi: 10.1155/2014/781670. URL https://doi.org/10.1155/2014/781670.

Xi Chen, Gordon Pennycook, and David Rand. What makes news sharable on social media? *Journal of Quantitative Description: Digital Media*, 3, 2023a.

Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, pages 6150–6160. PMLR, 2020.

Zhi Chen, Chudi Zhong, Margo Seltzer, and Cynthia Rudin. Understanding and exploring the whole set of good sparse generalized additive models. *arXiv preprint arXiv:2303.16047*, 2023b.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021. URL http://jmlr.org/papers/v22/20-1061.html.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Zijie J Wang, Chudi Zhong, Rui Xin, Takuya Takagi, Zhi Chen, Duen Horng Chau, Cynthia Rudin, and Margo Seltzer. Timbertrek: Exploring and curating sparse decision trees with interactive visualization. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 60–64. IEEE, 2022b.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016b.

Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. *Advances in neural information processing systems*, 31, 2018.