What made you do this?
Understanding black-box decisions with sufficient input subsets

Brandon Carter* **Jonas Mueller*** **Siddhartha Jain** **David Gifford**
MIT Computer Science and Artificial Intelligence Laboratory

Abstract

Local explanation frameworks aim to rationalize particular decisions made by a black-box prediction model. Existing techniques are often restricted to a specific type of predictor or based on input saliency, which may be undesirably sensitive to factors unrelated to the model's decision making process. We instead propose *sufficient input subsets* that identify minimal subsets of features whose observed values alone suffice for the same decision to be reached, even if all other input feature values are missing. General principles that globally govern a model's decision-making can also be revealed by searching for clusters of such input patterns across many data points. Our approach is conceptually straightforward, entirely model-agnostic, simply implemented using instance-wise backward selection, and able to produce more concise rationales than existing techniques. We demonstrate the utility of our interpretation method on various neural network models trained on text, image, and genomic data.

1 Introduction

The rise of neural networks and nonparametric methods in machine learning (ML) has driven significant improvements in prediction capabilities, while simultaneously earning the field a reputation of producing complex black-box models. Vital applications, which could benefit most from improved prediction, are often deemed too sensitive for opaque learning systems. Consider the widespread use of ML for screening people, including models that deny defendants' bail (Kleinberg et al., 2018) or reject loan applicants (Sirignano et al., 2018). It is imperative that such decisions can be interpretably rationalized. Interpretability is also crucial in scientific applications, where it is hoped that general principles may be extracted from accurate predictive models (Doshi-Velez and Kim, 2017; Lipton, 2016).

One simple explanation for *why* a particular black-box decision is reached may be obtained via a sparse subset of the input features whose values form the basis for the model's decision – a *rationale*. For text (or image) data, a rationale might consist of a subset of positions in the document (or image) together with the words (or pixel-values) occurring at these positions (see Figures 1 and 8). To ensure interpretations remain fully faithful to an arbitrary model, our rationales do not attempt to summarize the (potentially complex) operations carried out within the model, and instead merely point to the relevant information it uses to arrive at a decision (Lei et al., 2016). For high-dimensional inputs, sparsity of the rationale is imperative for greater interpretability.

Here, we propose a local explanation framework to produce rationales for a learned model that has been trained to map inputs $\mathbf{x} \in \mathcal{X}$ via some arbitrary learned function $f : \mathcal{X} \to \mathbb{R}$. Unlike many other interpretability techniques, our approach is not restricted to vector-valued data and does not require gradients of f. Rather, each input example is solely presumed to have a set of indexable features $\mathbf{x} = [x_1, \dots, x_p]$, where each $x_i \in \mathbb{R}^d$ for $i \in [p] = \{1, \dots, p\}$. We allow for features that are unordered (set-valued input) and whose number p may vary from input to input. A rationale corresponds to a sparse subset of these indices $S \subseteq [p]$ together with the specific values of the features in this subset.

To understand why a certain decision was made for a given input example $\mathbf{x}$, we propose a particular rationale called a *sufficient input subset* (SIS). Each SIS consists of a minimal input pattern present in $\mathbf{x}$ that alone suffices for f to produce the same decision, even

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

*Equal contribution. Code for this paper is available at: `http://github.com/b-carter/SufficientInputSubsets` if provided no other information about the rest of $\mathbf{x}$. Presuming the decision is based on $f(\mathbf{x})$ exceeding some prespecified threshold $\tau \in \mathbb{R}$, we specifically seek a minimal-cardinality subset S of the input features such that $f(\mathbf{x}_S) \geqslant \tau$. Throughout, we use $\mathbf{x}_S \in \mathcal{X}$ to denote a modified input example in which all information about the values of features outside subset S has been masked with features in S remaining at their original values. Thus, each SIS characterizes a particular standalone input pattern that drives the model toward this decision, providing sufficient justification for this choice from the model's perspective, even without any information on the values of the other features in $\mathbf{x}$.

In classification settings, f might represent the predicted probability of class C where we decide to assign the input to class C if $f(\mathbf{x}) \geqslant \tau$, chosen based on precision/recall considerations. Each SIS in such an application corresponds to a small input pattern that on its own is highly indicative of class C, according to our model. Note that by suitably defining f and τ with respect to the predictor outputs, any particular decision for input $\mathbf{x}$ can be precisely identified with the occurrence of $f(\mathbf{x}) \geqslant \tau$, where higher values of f are associated with greater confidence in this decision.

For a given input $\mathbf{x}$ where $f(\mathbf{x}) \geqslant \tau$, this work presents a simple method to find a complete collection of sufficient input subsets, each satisfying $f(\mathbf{x}_S) \geqslant \tau$, such that there exists no additional SIS outside of this collection. Each SIS may be understood as a disjoint piece of evidence that would lead the model to the same decision, and why this decision was reached for $\mathbf{x}$ can be unequivocally attributed to the SIS-collection. Furthermore, global insight on the general principles underlying the model's decision-making process may be gleaned by clustering the types of SIS extracted across different data points (see Figure 7 and 9). Such insights allow us to compare models based not only on their accuracy, but also on human-determined relevance of the concepts they target. Our method's simplicity facilitates its utilization by non-experts who may know very little about the models they wish to interrogate.

2 Related Work

Certain neural network variants such as attention mechanisms (Sha and Wang, 2017) and the generator-encoder of Lei et al. (2016) have been proposed as powerful yet human-interpretable learners. Other interpretability efforts have tailored decompositions to certain convolutional/recurrent networks (Murdoch et al., 2018; Olah et al., 2017, 2018), but these approaches are model-specific and only suited for ML experts. Many applications necessitate a model outside of these families, either to ensure supreme accuracy, or if training is done separately with access restricted to a black-box API (Caruana et al., 2015; Tramer et al., 2016).

An alternative model-agnostic approach to interpretability produces local explanations of f for a particular input $\mathbf{x}$. Local explanation often relies on attribution methods that quantify how much each feature influences the output of f at $\mathbf{x}$. Examples include LIME, which locally approximates f (Ribeiro et al., 2016), saliency maps based on gradients of f (Baehrens et al., 2010; Simonyan et al., 2014), Layer-wise Relevance Propagation (Bach et al., 2015), as well as the discrete DeepLIFT approach (Shrikumar et al., 2017) and its continuous variant – Integrated Gradients (IG) (Sundararajan et al., 2017), developed to ensure attributions reflect the cumulative difference in f at $\mathbf{x}$ vs. a reference input. A separate class of input-signal-based explanation techniques such as DeConvNet (Zeiler and Fergus, 2014), Guided Backprop (Springenberg et al., 2015), and PatternNet (Kindermans et al., 2018) employ gradients of f in order to identify input patterns that cause f to output large values. However, many gradient-based saliency methods have been deemed unreliable, depending not only on the learned function f, but also on its specific architectural implementation and how inputs are scaled (Kindermans et al., 2017, 2018). More like our approach, recent techniques from Dabkowski and Gal (2017); Kim et al. (2018); Chen et al. (2018) also aim to identify input patterns that best explain certain decisions, but additionally require either a predefined set of such patterns or an auxiliary neural network trained to identify them.

In comparison with the aforementioned methods, our SIS approach is: conceptually simple, entirely faithful to any type of model, and requires neither gradients of f nor auxiliary training of the underlying model f or a surrogate explanation model. Also related to our subset-selection methodology are the ideas of Li et al. (2017) and Fong and Vedaldi (2017), which for a particular input seek a small feature subset whose omission causes a substantial drop in f such that a different decision would be reached. However, this objective can produce adversarial artifacts that are hard to interpret. In contrast, we focus on identifying small subsets of input features whose values suffice to ensure f outputs significantly positive predictions, even in the absence of any other information about the rest of the input. While the techniques of Li et al. (2017) and Fong and Vedaldi (2017) produce rationales that remain highly dependent on the rest of the input outside of the selected feature subset, each rationale identified by our SIS approach is independently considered by f as an entirely sufficient justification for a particular decision in the absence of other information.

3 Methods

Our approach to rationalizing why a particular black-box decision is reached only applies to input examples $\mathbf{x} \in \mathcal{X}$ that meet the decision criterion $f(\mathbf{x}) \geqslant \tau$. For such an input $\mathbf{x}$, we aim to identify a SIS-collection of disjoint feature subsets $S_1, \ldots, S_K \subseteq [p]$ that satisfy the following criteria:

(1) $f(\mathbf{x}_{S_k}) \geqslant \tau$ for each $k = 1, \ldots, K$

(2) There exists no feature subset $S' \subset S_k$ for some $k = 1, \ldots, K$ such that $f(\mathbf{x}_{S'}) \geqslant \tau$

(3) $f(\mathbf{x}_R) < \tau$ for $R = [p] \setminus \bigcup_{k=1}^{K} S_k$ (the remaining features outside of the SIS-collection)

Criterion (1) ensures that for any SIS S_k, the values of the features in this subset alone suffice to justify the decision in the absence of any information regarding the values of the other features. To ensure information that is not vital to reach the decision is not included within the SIS, criterion (2) encourages each SIS to contain a minimal number of features, which facilitates interpretability. Finally, we require that our SIS-collection satisfies a notion of completeness via criterion (3), which states that the same decision is no longer reached for the input after the entire SIS-collection has been masked. This implies the remaining feature values of the input no longer contain sufficient evidence for the same decision. Figures 2 and 8 show SIS-collections found in text/image inputs.

Recall that $\mathbf{x}_S \in \mathcal{X}$ denotes a modified input in which the information about the values of features outside subset S is considered to be missing. We construct $\mathbf{x}_S$ as new input whose values on features in S are identical to those in the original $\mathbf{x}$, and whose remaining features $x_i \in [p] \backslash S$ are each replaced by a special mask $z_i \in \mathbb{R}^{d_i}$ used to represent a missing observation. While certain models are specially adapted to handle inputs with missing observations (Smola et al., 2005), this is generally not the case. To ensure our approach is applicable to all models, we draw inspiration from data imputation techniques which are a common way to represent missing data (Rubin, 1976).

Two popular strategies include hot-deck imputation, in which unobserved values are sampled from their marginal feature distribution, and mean imputation, in which each z_i simply fixed to the average value of feature i in the data. Note that for a linear model, these two strategies are expected to produce an identical change in prediction $f(\mathbf{x}) - f(\mathbf{x}_S)$. We find in practice that the change in predictions resulting from either masking strategy is roughly equivalent even for nonlinear models such as neural networks (Figure S12). In this work, we favor the mean-imputation approach over sampling-based imputation, which would be computationally-expensive and nondeterministic (undesirable for facilitating interpretability). One may also view $\mathbf{z}$ as the *baseline* input value used by feature attribution methods (Sundararajan et al., 2017; Shrikumar et al., 2017), a value which should not lead to particularly noteworthy decisions. Since our interests primarily lie in rationalizing atypical decisions, the average input arising from mean imputation serves as a suitable baseline. Zeros have also been used to mask image/categorical data (Li et al., 2017), but empirically, this mask appears undesirably more informative than the mean (predictions more affected by zero-masking).

For an arbitrarily complex function f over inputs with many features p, the combinatorial search to identify sets which satisfy objectives (1)-(3) is computationally infeasible. To find a SIS-collection in practice, we employ a straightforward backward selection strategy, which is here applied separately on an example-by-example basis (unlike standard statistical tools which perform backward selection globally to find a fixed set of features for all inputs). The **SIScollection** algorithm details our straightforward procedure to identify disjoint SIS subsets that satisfy (1)-(3) approximately (as detailed in §3.1) for an input $\mathbf{x} \in \mathcal{X}$ where $f(\mathbf{x}) \geqslant \tau$.

Our overall strategy is to find a SIS subset S_k (via **BackSelect** and **FindSIS**), mask it out, and then repeat these two steps restricting each search for the next SIS solely to features disjoint from the currently found SIS-collection $S_1, \ldots, S_k$, until the decision of interest is no longer supported by the remaining feature values. In the **BackSelect** procedure, $S \subset [p]$ denotes the set of remaining unmasked features that are to be considered during backward selection. For the current subset S, step 3 in **BackSelect** identifies which remaining feature $i \in S$ produces the *minimal* reduction in $f(\mathbf{x}_S) - f(\mathbf{x}_{S\backslash\{i\}})$ (meaning it least reduces the output of f if additionally masked), a question trivially answered by running each of the remaining possibilities through the model. This strategy aims to gradually mask out the least important features in order to reveal the core input pattern that is perceived by the model as sufficient evidence for its decision. Finally, we build our SIS up from the last ℓ features omitted during the backward selection, selecting a ℓ value just large enough to meet our sufficiency criterion (1). Because this approach always queries a prediction over the joint set of remaining features S, it is better suited to account for interactions between these features and ensure their sufficiency (i.e. that $f(\mathbf{x}_S) \geqslant \tau$) compared to a forward selection in the opposite direction which builds the SIS upwards one feature at a time by greedily maximizing marginal gains. Throughout its execution, **BackSelect** attempts to maintain the sufficiency of $\mathbf{x}_S$ as the set S shrinks.

```
**SIScollection**($f$, $\mathbf{x}$, $\tau$)
$S = [p]$
**for** $k = 1, 2, \ldots$ **do**
    $R$ = **BackSelect**$(f, \mathbf{x}, S)$
    $S_k$ = **FindSIS**$(f, \mathbf{x}, \tau, R)$
    $S \leftarrow S \backslash S_k$
    **if** $f(\mathbf{x}_S) < \tau$: **return** $S_1, \ldots, S_{k-1}$
```

```
**BackSelect**($f$, $\mathbf{x}$, $S$)
$R$ = empty stack
**while** $S \neq \varnothing$ **do**
    $i^* = \text{argmax}_{i \in S} f(\mathbf{x}_{S \backslash \{i\}})$
    Update $S \leftarrow S \backslash \{i^*\}$
    Push $i^*$ onto top of $R$
**return** $R$
```

```
**FindSIS**($f$, $\mathbf{x}$, $\tau$, $R$)
$S = \varnothing$
**while** $f(\mathbf{x}_S) < \tau$ **do**
    Pop $i$ from top of $R$
    Update $S \leftarrow S \cup \{i\}$
**if** $f(\mathbf{x}_S) \geqslant \tau$: **return** $S$
**else**: **return** *None*
```

3.1 Properties of the SIS-collection

Given p input features, our algorithm requires $\mathcal{O}(p^2 k)$ evaluations of f to identify k SIS, but we can achieve $\mathcal{O}(pk)$ by parallelizing each argmax in **BackSelect** (e.g. batching on GPU). Throughout, let $S_1, \ldots, S_K$ denote the output of **SIScollection** when applied to a given input $\mathbf{x}$ for which $f(\mathbf{x}) \geqslant \tau$. Disjointness of these sets is crucial to ensure computational tractability and that the number of SIS per example does not grow huge and hard to interpret. Proposition 1 below proves that each SIS produced by our procedure will satisfy an approximate notion of minimality. Because we desire minimality of the SIS as specified by (2), it is not appropriate to terminate the backward elimination in **BackSelect** as soon as the sufficiency condition $f(\mathbf{x}_S) \geqslant \tau$ is violated, due to the possible presence of local minima in f along the path of subsets encountered during backward selection (as shown in Figure S5).

Proposition 2 additionally guarantees that masking out the entirety of the feature values in the SIS-collection will ensure the model makes a different decision. Given $f(\mathbf{x}) \geqslant \tau$, it is thus necessarily the case that the observed values responsible for this decision lie within the SIS-collection $S_1, \ldots, S_K$. We point out that for an easily reached decision, where $f(\mathbf{z}) \geqslant \tau$ (i.e. this decision is reached even for the average input), our approach will not output any SIS. Because this same decision would likely be anyway reached for a vast number of inputs in the training data (as a sort of default decision), it is conceptually difficult to grasp what particular aspect of the given $\mathbf{x}$ is responsible.

Proposition 1. *There exists no feature i in any set $S_1, \ldots, S_K$ that can be additionally masked while retaining sufficiency of the resulting subset (i.e. $f(\mathbf{x}_{S_k \backslash \{i\}}) < \tau$ for any $k = 1, \ldots, K, i \in S_k$). Also, among all subsets S considered during the backward selection phase used to produce S_k, this set has the smallest cardinality of those which satisfy $f(\mathbf{x}_S) \geqslant \tau$.*

Proposition 2. *For $\mathbf{x}_{[p] \backslash S^*}$, modified by masking all features in the entire SIS-collection $S^* = \bigcup_{k=1}^{K} S_k$, we must have: $f(\mathbf{x}_{[p] \backslash S^*}) < \tau$ when $S^* \neq [p]$.*

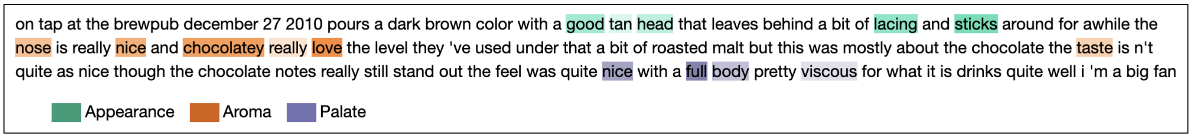
Unfortunately, nice assumptions like convexity/submodularity are inappropriate for estimated functions in ML. We present various simple forms of practical decision functions for which our algorithms are guaranteed to produce desirable explanations. Example 1 considers interpreting functions of a generalized linear form, Examples 2 & 3 describe functions whose operations resemble generalized logical *OR* & *AND* gates, and Example 4 considers functions that seek out a particular input pattern. Note that features ignored by f are always masked in our backward selection and thus never appear in the resulting SIS-collection.

Example 1. *Suppose the input data are vectors and $f(\mathbf{x}) = g(\beta^T \mathbf{x} + \beta_0)$, where g is monotonically increasing. We also presume $\tau > g(\beta_0)$ and the data were centered such that each feature has mean zero (for ease of notation). In this case, $S_1, \ldots, S_K$ must satisfy criteria (1)-(3). S_1 will consist of the features whose indices correspond to the largest ℓ entries of $\{\beta_1 x_1, \ldots, \beta_p x_p\}$ for some suitable ℓ that depends on the value of τ. It is also guaranteed that $f(\mathbf{x}_{S_1}) \geqslant f(\mathbf{x}_S)$ for any subset $S \subseteq [p]$ of the same cardinality $|S| = \ell$. For each individual feature i where $g(\beta_i x_i + \beta_0) \geqslant \tau$, there will be exist a corresponding SIS S_k consisting only of $\{i\}$. No SIS will include features whose coefficient $\beta_i = 0$, or those whose difference between the observed and average value z_i ($= 0$ here) is of an opposite sign than the corresponding model coefficient (i.e. $\beta_i(x_i - z_i) \leqslant 0$).*

Example 2. *Let $f(\mathbf{x}) = \max\{g_1(\mathbf{x}_{S'_1}), \ldots, g_L(\mathbf{x}_{S'_L})\}$ for some disjoint $S'_1, \ldots, S'_L \subset [p]$ and functions $g_1, \ldots, g_L$, such that for the given $\mathbf{x}$ and threshold τ: $g_1(\mathbf{x}_{S'_1}) > \cdots > g_L(\mathbf{x}_{S'_L}) \geqslant \tau$ and $g_k(\mathbf{x}_{S'_k \backslash \{i\}}) < \tau$ for each $1 \leqslant k \leqslant L, i \in S'_k$. Such f might be functions that model strong interactions between the features in each S_k or look for highly specific value patterns to occur these subsets. In this case,* **SIScollection** *will return L sets such that $S_1 = S'_1, S_2 = S'_2, \ldots, S_L = S'_L$.*

Example 3. *If $f(\mathbf{x}) = \min\{g_1(\mathbf{x}_{S'_1}), \ldots, g_L(\mathbf{x}_{S'_L})\}$ and the same conditions from Example 2 are met, then* **SIScollection** *will return a single set $S_1 = \bigcup_{k=1}^{L} S'_k$.*

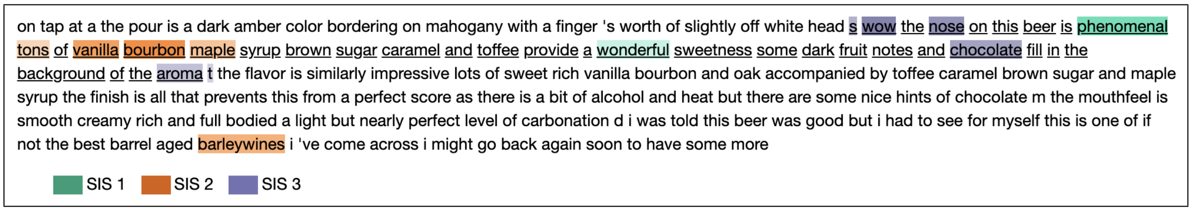
Example 4. *Suppose $\mathbf{x} \in \mathbb{R}^p$ with $f(\mathbf{x}) = h(||\mathbf{x}_S - \mathbf{c}_S||)$ where h is monotonically decreasing and $\mathbf{c}_S$ specifies a fixed pattern of input values for features in a certain subset S. For input $\mathbf{x}$ and threshold choice $\tau = f(\mathbf{x})$,* **SIScollection** *will return a single set $S_1 = \{i \in S : |x_i - c_i| < |z_i - c_i|\}$.*



on tap at the brewpub december 27 2010 pours a dark brown color with a good tan head that leaves behind a bit of lacing and sticks around for awhile the nose is really nice and chocolatey really love the level they 've used under that a bit of roasted malt but this was mostly about the chocolate the taste is n't quite as nice though the chocolate notes really still stand out the feel was quite nice with a full body pretty viscous for what it is drinks quite well i 'm a big fan

🟩 Appearance 🟧 Aroma 🟪 Palate

Figure 1: Beer review with one sufficient input subset identified for the prediction of each aspect.



on tap at a the pour is a dark amber color bordering on mahogany with a finger 's worth of slightly off white head s wow the nose on this beer is phenomenal tons of vanilla bourbon maple syrup brown sugar caramel and toffee provide a wonderful sweetness some dark fruit notes and chocolate fill in the background of the aroma t the flavor is similarly impressive lots of sweet rich vanilla bourbon and oak accompanied by toffee caramel brown sugar and maple syrup the finish is all that prevents this from a perfect score as there is a bit of alcohol and heat but there are some nice hints of chocolate m the mouthfeel is smooth creamy rich and full bodied a light but nearly perfect level of carbonation d i was told this beer was good but i had to see for myself this is one of if not the best barrel aged barleywines i 've come across i might go back again soon to have some more

🟩 SIS 1 🟧 SIS 2 🟪 SIS 3

Figure 2: Beer review with three disjoint SIS S_1, S_2, S_3 identified for a positive aroma prediction. Underlined are sentences that human labelers manually annotated as capturing the aroma sentiment.

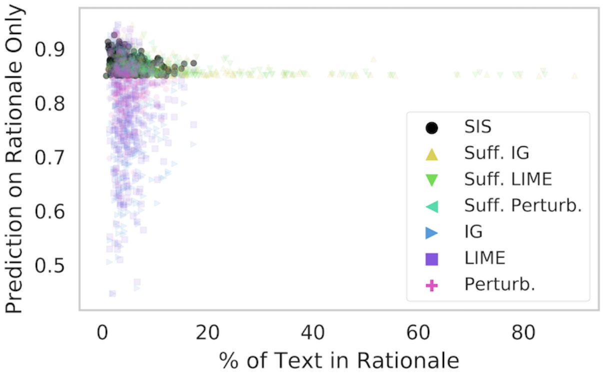


Figure 3: Prediction on rationales only vs. rationale length for various methods in reviews with positive aroma prediction ($\tau = 0.85$).

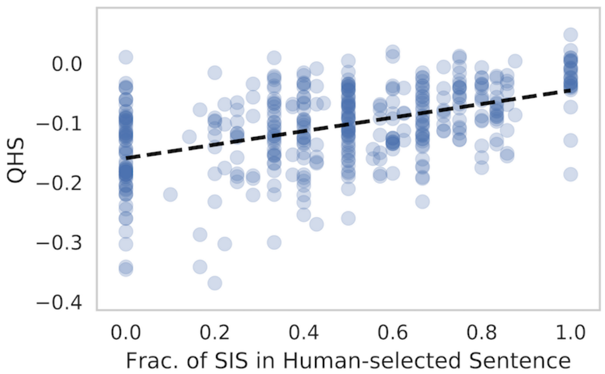


Figure 4: QHS vs. similarity between SIS & annotation in the reviews with positive aroma sentiment (Pearson $\rho = 0.491$, p-value $= 1.5\text{e}{-25}$).

4 Results

We apply our methods to analyze neural networks for text, DNA, and image data. **SIScollection** is compared with alternative subset-selection methods for producing rationales (see descriptions in Supplement §S1). Note that our **BackSelect** procedure determines an ordering of elements, R, subsequently used to construct the SIS. Depictions of each SIS are shaded based on the feature order in R (darker = later), which can indicate relative feature importance within the SIS.

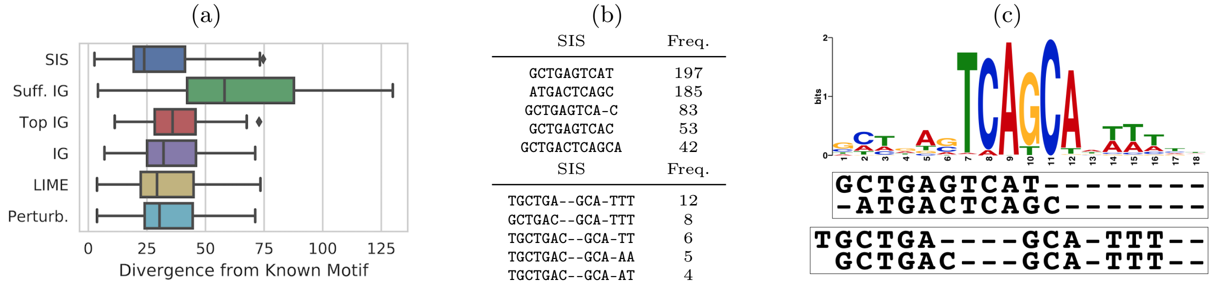
In the “Suff. IG,” “Suff. LIME,” and “Suff. Perturb.” (*sufficiency constrained*) methods, we instead compute the ordering of elements R according to the feature attribution values output by integrated gradients (Sundararajan et al., 2017), LIME (Ribeiro et al., 2016), or a perturbative approach that measures the change in prediction when individually masking each feature (see §S1). The rationale subset S produced under each method is subsequently assembled using **FindSIS** exactly as in our approach and thus is guaranteed to satisfy $f(\mathbf{x}_S) \geqslant \tau$. In the “IG,” “LIME,” and “Perturb.” (*length constrained*) methods, we use the same previously described ordering R, but always select the same number of features in the rationale as in the SIS produced by our method (per example). We also compare against the additional “Top IG” method, in which top features from R are added into the rationale until sum of integrated gradients attributions suggests that the rationale has met our sufficiency criterion (see §S1).

4.1 Sentiment Analysis of Reviews

We first consider a dataset of beer reviews from McAuley et al. (2012). Taking the text of a review as input, different LSTM networks (Hochreiter and Schmidhuber, 1997) are trained to predict user-provided numerical ratings of aspects like aroma, appearance, and palate (details in §S4). Figure 1 shows a sample beer review where we highlight the SIS identified for the LSTM that predicts each aspect. Each SIS only captures sentiment toward the relevant aspect. Figure 2 depicts the SIS-collection identified from a review the LSTM decided to flag for positive aroma.

```
CACTGTCATTCTCTTGGTCAGCCCTGGACATCCCTGGAAAGGATGACTCAGCTGTCCGTTTTAAACAGGGTAGTTCAGAAGAATACATTCCTGGTTATTCA
TTTTTTTCTCCCTTCGATTTCCACTATGATTTGTATTTCCTTTGTTCTGCTGACTTTGCAATTTCGGTTGTTTTTTCTAAATTTCTTAGGGTGAAAACTGA
```

Figure 5: Two DNA sequences that receive positive TF binding predictions for the MAFF factor (SIS is shaded).



SIS	Freq.
GCTGAGTCAT	197
ATGACTCAGC	185
GCTGAGTCA-C	83
GCTGAGTCAC	53
GCTGACTCAGCA	42
SIS	**Freq.**
TGCTGA--GCA-TTT	12
GCTGAC--GCA-TTT	8
TGCTGAC--GCA-TT	6
TGCTGAC--GCA-AA	5
TGCTGAC--GCA-AT	4

Figure 6: **(a)** KL divergence between JASPAR motifs (known ground truth) and rationales found via various methods. Shown are results for 422 TF datasets (each one summarized by median divergence). **(b)** In the SIS found in data from one TF, DBSCAN identified two clusters (most frequently-occurring SIS in each shown). **(c)** Known JASPAR motif (top) and alignment with cluster modes (bottom).

Figure 3 shows that when the alternative methods described in §4 are length constrained, the rationales they produce often badly fail to meet our sufficiency criterion. Thus, even though the same number of feature values are preserved in the rationale and these alternative methods select the features to which they have assigned the largest attribution values, their rationales lead to significantly reduced f outputs compared to our SIS subsets. If the sufficiency constraint is instead enforced for these alternative methods, the rationales they identify become significantly larger than those produced by **SIScollection**, and also contain many more unimportant features (Table S2, Figure S13).

Benchmarking interpretability methods is difficult because a learned f may behave counterintuitively such that seemingly unreasonable model explanations are in fact faithful descriptions of a model's decision-making process. For some reviews, a human annotator has manually selected which sentences carry the relevant sentiment for the aspect of interest, so we treat these annotations as an alternative rationale for the LSTM prediction. For a review $\mathbf{x}$ whose true and predicted aroma exceed our decision threshold, we define the *quality of human-selected sentences for model explanation* QHS $= f(\mathbf{x}_S) - f(\mathbf{x})$ where S is the human-selected-subset of words in the review (see examples in Figure S18). High variability of QHS in the annotated reviews (Figure 4) indicates the human rationales often do not contain sufficient information to preserve the LSTM's decision. Figure 4 shows the LSTM makes many decisions based on different subsets of the text than the parts that humans find appropriate for this task. Reassuringly, our SIS more often lie within the selected annotation for reviews with high QHS scores.

4.2 Transcription Factor Binding

We next analyze convolutional neural networks (CNN) used to classify whether a given transcription factor (TF) will bind to a specific DNA sequence (Zeng et al., 2016). From 422 different datasets of DNA sequences bound-or-not by different TFs (and 422 different CNN models), we extract SIS-collections from sequences with high (top 10%) predicted binding affinity for the TF profiled in each dataset (details in §S2). Figure 5 depicts two input examples and the corresponding identified SIS. Again, rationales produced via our SIS approach are shorter and better at preserving large f-values than rationales from other methods (Figures S3 and S4).

To predict binding so accurately, the CNN must faithfully reflect the biological mechanisms that relate the DNA sequence to the probability of TF occupancy. We evaluate the rationales found by our methods against known TF binding motifs from JASPAR (Mathelier et al., 2015), adopting KL divergence between the known motif and each proposed rationale as a quality measure (see §S2.3). Figure 6a shows the divergence of rationales produced by **SIScollection** is significantly lower than that of rationales identified using other methods (Wilcoxon $p \leqslant$ 1e$-$5 in all cases). SIS is thus more effective at uncovering the underlying biological principles than the alternative methods we applied.

4.3 MNIST Digit Classification

Finally, we study a 10-way CNN classifier trained on the MNIST handwritten digits data (LeCun et al., 1998). Here, we only consider predicted probabilities for one class of interest at a time and always set $\tau = 0.7$ as the probability threshold for deciding that an image

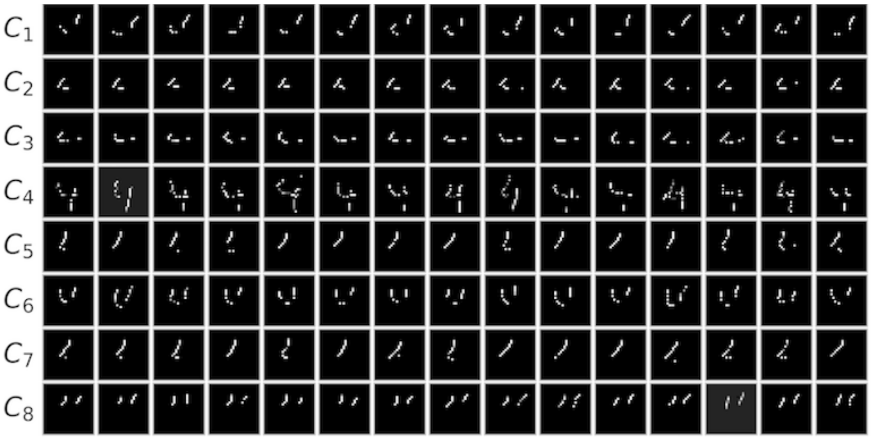


Figure 7: Eight clusters of SIS identified from examples of digit 4. Each row contains fifteen random SIS from a single cluster.

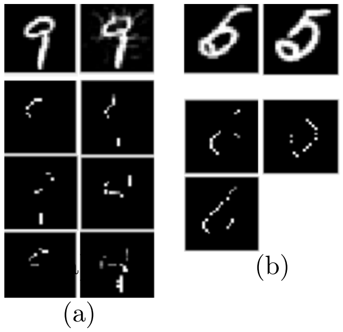


Figure 8: **(a)** SIS for correctly classified 9 (1st column) and when adversarially perturbed toward class 4 (2nd column). **(b)** SIS for digits 5 that are misclassified as 6 (1st column) and as 0 (2nd column).

belongs to the class. We extract the SIS-collection from all corresponding test set examples (details in §S3). Example images and corresponding SIS-collections are shown in Figures 8 and S8. Figure 8a illustrates how the SIS-collection drastically changes for an example of a correctly-classified 9 that has been adversarially manipulated (Carlini and Wagner, 2017) to become confidently classified as the digit 4. Furthermore, these SIS-collections immediately enable us to understand why certain misclassifications occur (Figure 8b).

4.4 Clustering SIS for General Insights

Identifying the different input patterns that justify a decision can help us better grasp the general operating principles of a model. To this end, we cluster all of the SIS produced by **SIScollection** applied across a large number of data examples that received the same decision. Clustering is done via DBSCAN, a widely applicable algorithm that merely requires specifying pairwise distances between points (Ester et al., 1996).

We first apply this procedure to the SIS found across all test-set DNA sequences which our CNN model predicted would be bound by some TF. Here, the pairwise distance between two sufficient input subsets is taken to be the Levenshtein (edit) distance. Figure 6 shows the clusters for a particular TF where two SIS clusters were found. Despite no contiguity being enforced in our algorithm, each cluster is comprised of short sequences that clearly capture different aspects of the underlying DNA motif known to bind this TF.

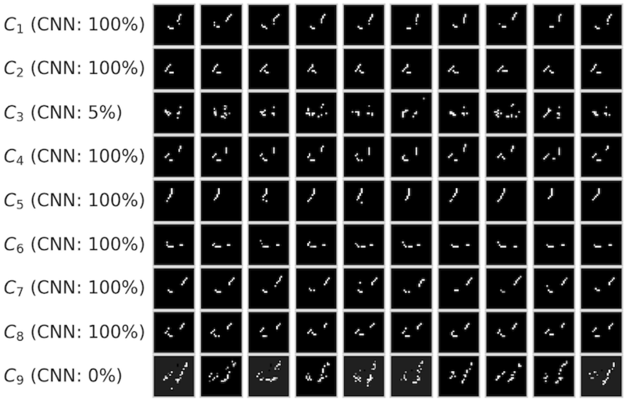


Figure 9: Jointly clustering the MNIST digit 4 SIS from CNN and MLP. We list the percentage of SIS in each cluster stemming from the CNN (rest from MLP).

We also apply DBSCAN clustering to the SIS found across all MNIST test-examples confidently identified by the CNN as a particular class. Pairwise distances are here defined as the *energy distance* (Rizzo and Székely, 2016) over pixel locations between two SIS subsets (see §S3.3). Figure 7 depicts the SIS clusters identified for digit 4 (others in Figure S9). These reveal distinct feature patterns learned by the CNN to distinguish 4 from other digits, which are clearly present in the vast majority of test set images confidently classified as a 4. For example, cluster C_8 depicts parallel slanted lines, a pattern that never occurs in other digits.

Subsequently, we cluster the SIS found across held-out beer reviews (Test-Fold in Table S1) that received positive aroma predictions from our LSTM network. The distance between two SIS is taken as the Jaccard distance between their bag of words representations. Three clusters depicted in Table 1 (rest in Tables S3, S4) reveal isolated phrases that the LSTM associates with positive aromas in the absence of other context.

The general insights revealed by our SIS-clustering can also be used to compare the operating-behavior of different models. For the beer reviews, we also train a CNN to compare with our existing LSTM (see §S4.6). For MNIST, we train a multilayer perceptron (MLP)

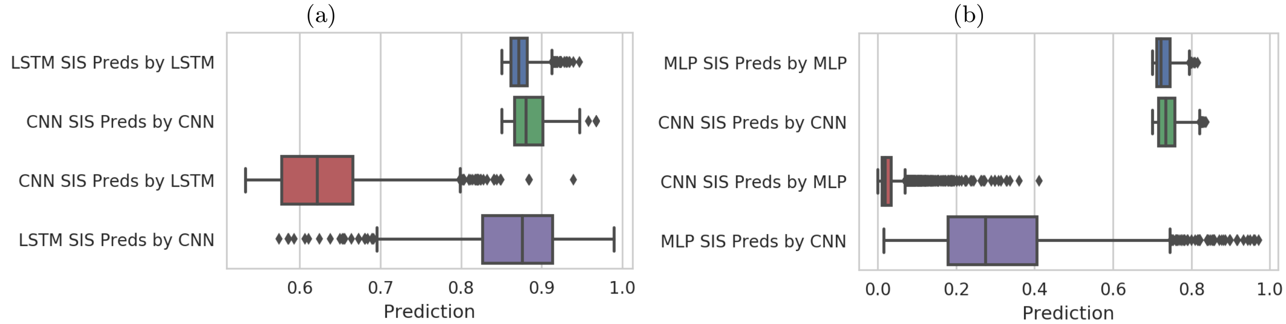


Figure 10: Predictions by one model on the SIS extracted from the other model in: **(a)** beer reviews with positive LSTM/CNN aroma predictions, and **(b)** MNIST digits confidently classified as 4 by CNN/MLP.

Table 1: 3 clusters of SIS extracted from beer reviews with positive CNN aroma predictions. Each row shows 4 most frequent unique SIS in a cluster (each SIS shown as ordered word list with text-positions omitted). Each unique SIS can be present many times in one cluster.

Clu.	**SIS #1**	**SIS #2**	**SIS #3**	**SIS #4**
C_1	smell amazing wonderful	nice wonderful nose	wonderful amazing	amazing amazing
C_2	grapefruit mango pineapple	pineapple grapefruit pineapple grapefruit	hops grapefruit pineapple floyds	mango pineapple incredible
C_3	creme brulee brulee	creme brulee decadent	incredible creme brulee	creme brulee exceptional

Table 2: Joint clustering of the SIS from beer reviews predicted to have positive aroma by LSTM or CNN. Dashes are used in clusters with under 4 unique SIS. Percentages quantify SIS per cluster from the LSTM.

Clu.	**LSTM**	**SIS #1**	**SIS #2**	**SIS #3**	**SIS #4**
C_1	0%	delicious	-	-	-
C_2	0%	very nice	-	-	-
C_3	20%	rich chocolate	very rich	chocolate complex	smells rich
C_4	33%	oak chocolate	chocolate raisins raisins oak bourbon	chocolate oak	raisins chocolate
C_5	70%	complex aroma	aroma complex peaches complex	aroma complex interesting cherries	aroma complex

and compare to our existing CNN (see §S3.5). Both networks exhibit similar performance in each task, so it is not immediately clear which model would be preferable in practice. Figure 10 shows the SIS extracted under one model are typically insufficient to receive the same decision from the other model, indicating these models base their positive predictions on different evidence.

Figure 9 depicts results from a joint clustering of all SIS extracted from held-out MNIST images confidently classified as a 4 by either the MLP or CNN. Evidently, our MNIST-CNN bases its confidence primarily on spatially-contiguous strokes comprising only a small portion of each digit. MLP-decisions are in contrast based on pixels located throughout the digit, demonstrating this model relies more on the global shape of the handwriting. Thus, the CNN is more susceptible to mistaking other (non-digit) handwritten characters for 4s if they happen to share some of the same strokes. Table 2 contains results of jointly clustering the SIS extracted from beer reviews with positive aroma predictions under our LSTM or text-CNN. This CNN tends to learn localized (unigram/bigram) word patterns, while the LSTM identifies more complex multi-word interactions that truly seem more relevant to the target aroma value. Many CNN-SIS are simply phrases with universally-positive sentiment, indicating this model is less capable at distinguishing between positive sentiment toward aroma vs. other aspects such as taste/look.

5 Discussion

This work introduced the idea of interpreting black-box decisions on the basis of sufficient input subsets – minimal input patterns that alone provide sufficient evidence to justify a particular decision. Our methodology is easy to understand for non-experts, applicable to all ML models without any additional training steps, and remains fully faithful to the underlying model without making approximations. While we focus on local explanations of a single decision, clustering the SIS-patterns extracted from many data points reveals insights about a model's general decision-making process. Given multiple models of comparable accuracy, SIS-clustering can uncover critical operating differences, such as which model is more susceptible to spurious training data correlations or will generalize worse to counterfactual inputs that lie outside the data distribution.

Acknowledgements

We thank Haoyang Zeng and Ge Liu for help with the TF data/models. This work was supported by NIH Grants R01CA218094, R01HG008363, and R01HG008754.

References

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One*, 10(7):e0130140.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2018). Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*.

Dabkowski, P. and Gal, Y. (2017). Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608*.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. (2017). The (un) reliability of saliency methods. In *NIPS Workshop: Interpreting, Explaining and Visualizing Deep Learning - Now what?*

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. (2018). Learning how to explain neural networks: PatternNet and PatternAttribution. In *International Conference on Learning Representations*.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. (2018). Human decisions and machine predictions. *The Quarterly Journal of Economics*, 133(1):237–293.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Empirical Methods in Natural Language Processing*.

Li, J., Monroe, W., and Jurafsky, D. (2017). Understanding neural networks through representation erasure. *arXiv:1612.08220*.

Lipton, Z. C. (2016). The mythos of model interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*.

Mathelier, A., Fornes, O., Arenillas, D. J., Chen, C.-y., Denay, G., Lee, J., Shi, W., Shyr, C., Tan, G., Worsley-Hunt, R., et al. (2015). Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 44(D1):D110–D115.

McAuley, J., Leskovec, J., and Jurafsky, D. (2012). Learning attitudes and attributes from multi-aspect reviews. In *IEEE International Conference on Data Mining*, pages 1020–1025.

Murdoch, W. J., Liu, P. J., and Yu, B. (2018). Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations*.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Rizzo, M. L. and Székely, G. J. (2016). Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Sha, Y. and Wang, M. D. (2017). Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*.

Sirignano, J. A., Sadhwani, A., and Giesecke, K. (2018). Deep learning for mortgage risk. *arXiv:1607.02470*.

Smola, A. J., Vishwanathan, S., and Hofmann, T. (2005). Kernel methods for missing variables. In *Artificial Intelligence and Statistics*.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*.

Tramer, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In *USENIX Security Symposium*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*.

Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. (2016). Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, 32(12):i121.