Sparse Robust Regression for Explaining Classifiers

Anton Björklund[1(✉)], Andreas Henelius[1], Emilia Oikarinen[1], Kimmo Kallonen[2], and Kai Puolamäki[1]

[1] Department of Computer Science, University of Helsinki, Helsinki, Finland
{anton.bjorklund,andreas.henelius,emilia.oikarinen,
kai.puolamaki}@helsinki.fi
[2] Helsinki Institute of Physics, University of Helsinki, Helsinki, Finland
kimmo.kallonen@helsinki.fi

Abstract. Real-world datasets are often characterised by outliers, points far from the majority of the points, which might negatively influence modelling of the data. In data analysis it is hence important to use methods that are robust to outliers. In this paper we develop a robust regression method for finding the largest subset in the data that can be approximated using a sparse linear model to a given precision. We show that the problem is NP-hard and hard to approximate. We present an efficient algorithm, termed SLISE, to find solutions to the problem. Our method extends current state-of-the-art robust regression methods, especially in terms of scalability on large datasets. Furthermore, we show that our method can be used to yield interpretable explanations for individual decisions by opaque, black box, classifiers. Our approach solves shortcomings in other recent explanation methods by not requiring sampling of new data points and by being usable without modifications across various data domains. We demonstrate our method using both synthetic and real-world regression and classification problems.

1 Introduction and Related Work

In analyses of real-world data we often encounter outliers, i.e., points which are far from the majority of the other data points. Such points are problematic as they may negatively influence modelling of the data. This is observed in, e.g., ordinary least-squares regression where already a single data point may lead to arbitrarily large errors [11]. It is hence important to use *robust methods* that effectively ignore the effect of outliers. A number of approaches have been proposed for robust regression, see, e.g., [27] for a review. Our proposed method is most closely related to *Least Trimmed Squares* (LTS) [2,26,28] that finds a subset of size k minimising the sum of the squared residuals in this subset, in contrast to methods that de-emphasise [33] or penalise [20,30,34] outliers.

In this paper we present a sparse robust regression method that outperforms many of the existing state-of-the-art robust regression methods in terms of scalability on large datasets, termed SLISE (Sparse LInear Subset Explanations).

P. Kralj Novak et al. (Eds.): DS 2019, LNAI 11828, pp. 351–366, 2019.
https://doi.org/10.1007/978-3-030-33778-0_27

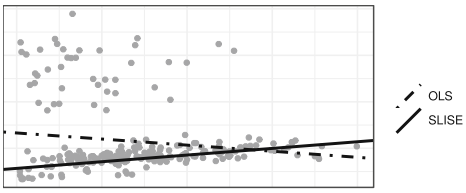


Fig. 1. Robust regression.

Table 1. Classifier probabilities for *high income*.

	Education	
Age	*Low*	*High*
Young	0.07	0.31
Old	0.22	0.61

Specifically, we consider *finding the largest subset of data items that can be represented by a linear model to a given accuracy*. Hence, there is an important difference between our method and LTS: with LTS the size of the subset is fixed and specified a priori. Furthermore, the linear models obtained from SLISE are sparse, meaning that the model coefficients are easier to interpret, especially for datasets with many attributes.

Example 1: Robust Regression. Figure 1 shows a dataset containing outliers in the top left corner. Here ordinary least-squares regression (OLS) finds the wrong model due to the influence of these outliers. In contrast, SLISE *finds the largest subset of points that can be approximated by a (sparse) linear model*, yielding high robustness by ignoring the outliers.

Interestingly, it turns out that our robust regression method can also be used to *explain individual decisions by opaque (black box) machine learning models*: e.g., why does a classifier predict that an image contains the digit 2? The need for interpretability stems from the fact that high accuracy is not always sufficient; we must understand *how* the model works. This is important in safety-critical real-world applications, e.g., in medicine [6], but also in science, such as in physics when classifying particle jets [18]. In terms of explanations we consider *post-hoc interpretation of opaque models*, i.e., understanding predictions from already existing models, in contrast to creating models directly aiming for interpretability (e.g., super-sparse linear integer models [32] or decision sets [19]). In general, model explanations can be divided into *global* explanations (for the entire model), e.g., [1,10,16,17], and *local* explanations (for a single classification instances), e.g., [5,13,21,25]. Here we are interested in the latter. For a survey of explanations see, e.g., [15].

To explain an instance, we need to find a (simple and interpretable) model that *matches the black box model locally* in the *neighbourhood* of the instance whose classification we want to explain. Defining this neighbourhood is important but non-trivial (for discussion, see, e.g., [14,24]). The two central questions are: (i) how do we find the local model and (ii) how do we define the neighbourhood? Our approach solves these two problems at the same time by finding the largest subset of data items such that the residuals of a linear model passing through the instance we want to explain are minimised.

Example 2: Explanations. Consider a simple toy dataset of persons with the attributes age $\in \{0,1\}$ and education $\in \{0,1\}$, where 0 denotes low age and

education and 1 high age and education, respectively. Assume that the dataset consists mostly of people with high education, if we for example are studying factors affecting salaries within the faculty of a university department. Now, we are given a classifier that outputs the probability of high income (vs. low income), given these two attributes. Our task is to find the most important attribute used by the classifier when estimating the income level of an old professor in the dataset. Looking only at the class probabilities, shown in Table 1, it appears that education is the most significant attribute, and this is indeed what, e.g., the state-of-the art local explanation method LIME [25] finds. We, however, argue that this explanation is misleading: our toy data set contains very few instances of persons with low education, and therefore knowing the education level does not really give any information about the class. We argue that *in this dataset* age is a better determinant of high income, and this is found by SLISE.

The above example shows the importance of the interaction between the model and the data. The model in Table 1 is actually a simple logistic regression[1]. Hence, even if the model is simple, a complex structure in the data can make interpretation non-trivial. LIME found the simple logistic regression model, whereas we found the behaviour of the model in the dataset. This distinction is significant because it suggests that you cannot always cleanly separate the model from the data. An example of this is conservation laws in physical systems. Accurate data will never violate such laws, which is something the model can rely on. Without adhering to the data during the explanation you may therefore find explanations that violate the laws of physics. SLISE satisfies such constraints automatically by observing how the classifier performs in the dataset, instead of randomly sampling (possibly non-physical) points around the item of interest (as in, e.g., [5,13,21,25]). Another advantage is that we do not need to define a neighbourhood of a data item, which is especially important in cases where modelling the distance is difficult, such as with images.

Contributions. We develop a novel robust regression method with applications to local explanations of opaque machine learning models. We consider the problem of *finding the largest subset that can be approximated by a sparse linear model* which is **NP**-hard and hard to approximate (Theorem 1) and present an approximative algorithm for solving it (Algorithm 1). We demonstrate empirically using synthetic and real-world datasets that SLISE outperforms state-of-the-art robust regression methods and yields sensible explanations for classifiers.

Organisation. In Sect. 2 we formalise our problem for both robust regression and local explanations, and show its complexity. We then discuss practical numeric optimisation in Sect. 3. The algorithm is presented in Sect. 4, followed by the empirical evaluation in Sect. 5. We end with the conclusions in Sect. 6.

2 Problem Definition

Our goal is to develop a linear regression method with applications to both (i) robust *global linear regression model* and (ii) providing a *local linear regression*

[1] Probability of high income is given by $p = \sigma(-2.53 + 1.73 \cdot \text{education} + 1.26 \cdot \text{age})$.

model of the decision surface of an opaque model in the vicinity of a particular data item. In the second case the simple linear model thus provides an explanation for the (typically more) complex decision surface of the opaque model.

Let (X, Y), where $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^n$, be a dataset consisting of n pairs $\{(x_i, y_i)\}_{i=1}^n$ where we denote the ith d-dimensional item (row) in X by x_i (the *predictor*) and similarly the ith element in Y by y_i (the *response*). Furthermore let ε be the largest tolerable error and λ be a regularisation coefficient. We now state the main problem in this paper:

Problem 1. Given $X \in \mathbb{R}^{n \times d}$, $Y \in \mathbb{R}^n$, and non-negative $\varepsilon, \lambda \in \mathbb{R}$, find the regression coefficients $\alpha \in \mathbb{R}^d$ minimising the *loss function*

$$\texttt{Loss}(\varepsilon, \lambda, X, Y, \alpha) = \sum\nolimits_{i=1}^{n} H(\varepsilon^2 - r_i^2)\left(r_i^2/n - \varepsilon^2\right) + \lambda \|\alpha\|_1, \tag{1}$$

where the residual errors are given by $r_i = y_i - \alpha^\intercal x_i$, $H(\cdot)$ is the Heaviside step function satisfying $H(u) = 1$ if $u \geq 0$ and $H(u) = 0$ otherwise, and $\|\alpha\|_1 = \sum_{i=1}^{d} |\alpha_i|$ denotes the L1-norm. If necessary, X can be augmented with a column of all ones to accommodate the *intercept* term of the model.

Alternatively, the Lagrangian term $\lambda\|\alpha\|_1$ in Eq. (1) can be replaced by a constraint $\|\alpha_i\|_1 \leq t$ for some t. Note that Problem 1 is a combinatorial problem in disguise, where we try to find a maximal subset S, as can be seen by rewriting Eq. (1) as (using the shorthand $[n] = \{1, \ldots, n\}$)

$$\texttt{Loss}(\varepsilon, \lambda, X, Y, \alpha) = \sum_{i \in S} \left(r_i^2/n - \varepsilon^2\right) + \lambda \|\alpha\|_1 \text{ where } S = \{i \in [n] \mid r_i^2 \leq \varepsilon^2\}. \tag{2}$$

The loss function of Eq. (1) (and Eq. (2)) thus consists of three parts; the maximisation of subset size $\sum_{i \in S} \varepsilon^2 = |S|\varepsilon^2$, the minimisation of the residuals $\sum_{i \in S} r_i^2/n \leq \varepsilon^2$, and the LASSO-regularisation $\lambda\|\alpha\|_1$. The main goal is to maximise the subset and this is reflected in the loss function, since any decrease of the subset size has an equal or greater impact on the loss than all the residuals combined. At the limit of $\varepsilon \to \infty$, it follows that $S = [n]$ and Problem 1 is equivalent to LASSO [31]. We now state the following theorem concerning the complexity of Problem 1.

Theorem 1. *Problem 1 is* **NP***-hard and hard to approximate.*

Proof. We prove the theorem by a reduction to the MAXIMUM SATISFYING LINEAR SUBSYSTEM problem [4, Problem MP10], which is known to be **NP**-hard. In MAXIMUM SATISFYING LINEAR SUBSYSTEM we are given the system $X\alpha = y$, where $X \in \mathbb{Z}^{n \times m}$ and $y \in \mathbb{Z}^n$ and we want to find $\alpha \in \mathbb{Q}^m$ such that as many equations as possible are satisfied. This is equivalent to Problem 1 with $\varepsilon = 0$ and $\lambda = 0$. Also, the problem is not approximable within n^γ for some $\gamma > 0$ [3]. □

Local Explanations. To provide a local explanation for a data item (x_k, y_k) where $k \in [n]$, we use an additional constraint requiring that the regression plane passes through this item, i.e., we add the constraint $r_k = 0$ to Problem 1. This

constraint is easily met by centring the data on the item (x_k, y_k) to be explained: $y_i \to y_i - y_k$ and $x_i \to x_i - x_k$ for all $i \in [n]$, in which case $r_k = 0$ and any potential intercept is zero. Hence, it suffices to consider Problem 1 both when finding the best global regression model and when providing a local explanation for a data item.

In practice, we employ the following procedure to generate local explanations for classifiers. If a classifier outputs class probabilities $P \in \mathbb{R}^n$ we transform them to linear values using the logit transformation $y_i = \log(p_i/(1-p_i))$, yielding a vector $Y \in \mathbb{R}^n$. This new vector $Y - y_k$ is what we use for finding the explanation.

Now, the local linear model, α from Problem 1, and the subset, S from Eq. (2), constitute the explanation for the data item of interest. Note that the linear model is comparable to the linear model obtained using standard logistic regression, i.e., we approximate the black box classifier by a logistic regression in the vicinity of the point of interest.

3 Numeric Approximation

We cannot effectively solve the optimisation problem in Problem 1 in the general case. Instead, we relax the problem by replacing the Heaviside function with a sigmoid function σ and a continuous and differentiable rectifier function $\phi(u) \approx \min(0, u)$. This allows us to compute the gradient and find α by minimising

$$\beta\text{-}\texttt{Loss}(\varepsilon, \lambda, X, Y, \alpha) = \sum\nolimits_{i=1}^{n} \sigma(\beta(\varepsilon^2 - r_i^2))\phi\left(r_i^2/n - \varepsilon^2\right) + \lambda\|\alpha\|_1, \qquad (3)$$

where the parameter β determines the steepness of the sigmoid and the rectifier function ϕ is parametrised by a small constant $\omega > 0$ such that $\phi(u) = u$ for $u < -\omega$, $\phi(u) = -(u^2/\omega + \omega)/2$ for $-\omega \leq u \leq 0$, and $\phi(u) = -\omega/2$ for $0 < u$. It is easy to see that Eq. (3) is a smoothed variant of Eq. (1) and that the two become equal when $\beta \to \infty$ and $\omega \to 0^+$.

We perform this minimisation using *graduated optimisation*, where the idea is to iteratively solve a difficult optimisation problem by progressively increasing the complexity [23]. A natural parametrisation for the complexity of our problem is via the β parameter. We start from $\beta = 0$ which corresponds to a convex optimisation problem equivalent to LASSO, and gradually increase the value of β towards ∞ which corresponds to the Heaviside solution of Eq. (1). At each step, we use the previous optimal value of α as a starting point for minimisation of Eq. (3). It is important that the optima of the consecutive solutions with increasing values of β are close enough, which is why we derive an approximation ratio between the solutions with different values of β. We observe that our problem can be rewritten as a maximisation of $-\beta\text{-}\texttt{Loss}(\varepsilon, \lambda, X, Y, \alpha)$. The choice of β does not affect the L1-norm and we omit it for simplicity ($\lambda = 0$).

Theorem 2. *Given $\varepsilon, \beta_1, \beta_2 > 0$, such that $\beta_1 \leq \beta_2$, and the functions $f_j(r) = -\sigma(\beta_j(\varepsilon^2 - r^2))\phi(r^2/n - \varepsilon^2)$, and $G_j(\alpha) = \sum_{i=1}^{n} f_j(r_i)$ where $r_i = y_i - \alpha^\intercal x_i$ and $j \in \{1, 2\}$. For $\alpha_1 = \arg\max_\alpha G_1(\alpha)$ and $\alpha_2 = \arg\max_\alpha G_2(\alpha)$ the inequality $G_2(\alpha_2) \leq K G_2(\alpha_1)$ always holds, where $K = G_1(\alpha_1)/\left(G_2(\alpha_1) \min_r f_1(r)/f_2(r)\right)$ is the approximation ratio.*

Parameters: (1) Dataset $X \in \mathbb{R}^{n \times d}$, $Y \in \mathbb{R}^n$, (2) error tolerance ε, (3) regularisation coefficient λ, (4) sigmoid steepness $\beta_{\max}$, (5) target approximation ratio $r_{\max}$

1 **Function** `SLISE`(X, Y, ε, λ, $\beta_{\max}$, $r_{\max}$)
2 $\quad\alpha \leftarrow$ `OrdinaryLeastSquares`(X, Y) and $\beta \leftarrow 0$
3 $\quad$**while** $\beta < \beta_{\max}$ **do**
4 $\quad\quad\alpha \leftarrow$ `OWL-QN`(β-`Loss`, ε, λ, X, Y, α)
5 $\quad\quad\beta \leftarrow \beta'$ such that `AppoximationRatio`(X, Y, ε, β, β', α) $= r_{\max}$
6 $\quad\alpha \leftarrow$ `OWL-QN`(β-`Loss`, ε, λ, X, Y, α)
$\quad$**Result**: α

Algorithm 1: The SLISE algorithm.

Proof. Let us first argue the non-negativity of f_1 and f_2. The inequalities $\sigma(z) > 0$ and $\phi(z) < 0$ hold for all $z \in \mathbb{R}$, thus $f_j(r) > 0$. Now, by definition, $G_1(\alpha_2) \leq G_1(\alpha_1)$. We denote $r_i^* = y_i - \alpha_2^\intercal x_i$ and $k = \min_r f_1(r)/f_2(r)$, which allows us the rewrite and approximate:

$$G_1(\alpha_2) = \sum\nolimits_{i=1}^{n} f_1(r_i^*) = \sum\nolimits_{i=1}^{n} f_2(r_i^*) f_1(r_i^*)/f_2(r_i^*) \geq k G_2(\alpha_2).$$

Then $G_2(\alpha_2) \leq G_1(\alpha_2)/k \leq G_1(\alpha_1)/k \leq G_2(\alpha_1) G_1(\alpha_1)/(k G_2(\alpha_1))$, and the inequality from the theorem holds. □

We use Theorem 2 to choose the sequence of β values ($\beta_1 = 0, \beta_1, \ldots, \beta_l = \beta_{\max}$) so that at each step the approximation ratio as defined by K stays within a bound specified by the parameter $r_{\max}$ in Algorithm 1.

4 The SLISE Algorithm

In this section we describe an approximate numeric algorithm Algorithm 1 (SLISE) for solving Problem 1. As a starting point for the regression coefficients α we use the solution obtained from an ordinary least squares regression (OLS) on the full dataset (Algorithm 1, line 2). We now perform graduated optimisation (lines 3–5) in which we gradually increase the value of β from 0 to $\beta_{\max}$. At each iteration, we find the model α using the current value of β, such that β-`Loss` in Eq. (3) is minimised (line 4). To perform this optimisation we use OWL-QN [29], which is a quasi-Newton optimisation method with built-in L1-regularisation. We then increase β gradually (line 5) such that the approximation ratio K in Theorem 2 equals $r_{\max}$.

The time complexity of SLISE is affected by the three main parts of the algorithm; the loss function, OWL-QN, and graduated optimisation. The evaluation of the loss function has a complexity of $\mathcal{O}(nd)$, due to the multiplication between the linear model α and the data-matrix X. OWL-QN has a complexity of $\mathcal{O}(dp_o)$, where p_p is the number of iterations. Graduated optimisation is also an iterative method $\mathcal{O}(dp_g)$, but it only adds the approximation ratio calculation $\mathcal{O}(nd)$ (which is not dominant). Combining these complexities yields a complexity of $\mathcal{O}(nd^2p)$ for SLISE, where $p = p_o + p_g$ is the total number of iterations.

Table 2. The datasets. The synthetic dataset can be generated to the desired size.

	EMNIST	IMDB	PHYSICS	SYNTHETIC
Items	40 000	25 000	260 000	n
Dimensions	784	1000	5	d
Type	Image	Text	Tabular	-
Classifier	CNN	LR, SVM	NN	-

5 Experiments

SLISE has applications in both robust regression and for explaining black box models, and the experiments are hence divided into two parts. In the first part (Sect. 5.1) we consider SLISE as a *robust regression* method and demonstrate that (i) SLISE scales better on high-dimensional datasets than competing methods, (ii) SLISE is very robust to noise, and (iii) the solution found using SLISE is optimal. In the second part (Sect. 5.2) we use SLISE to *explain predictions from opaque models*. The experiments were run using R (v. 3.5.1) on a high-performance cluster [12] (4 cores from an Intel Xeon E5-2680 2.4 GHz with 16 Gb RAM). SLISE and the code to run the experiments is released as open source and is available from http://www.github.com/edahelsinki/slise.

Datasets. We use real (EMNIST [9], IMDB [22], PHYSICS [8]) and synthetic datasets in our experiments (properties given in Table 2). Synthetic datasets are generated as follows. The data matrix $X \in \mathbb{R}^{n \times d}$ is created by sampling from a normal distribution with zero mean and unit variance. The response vector $Y \in \mathbb{R}^n$ is created by $y_i \leftarrow a^\intercal x_i$ (plus some normal noise with zero mean and 0.05 variance), where $a \in \mathbb{R}^d$ is one of nine linear models drawn from a uniform distribution between -1 and 1. Each model creates 10% of the Y-values, except one that creates 20% of the Y-values. This larger chunk should enable robust regression methods to find the corresponding model.

Pre-processing. It is important both for robust regression and for local explanations to ensure that the magnitude of the coefficients in α are comparable, since sparsity is enforced by L1-penalisation of the elements in α. Hence, we normalize the PHYSICS datasets dimension-wise by subtracting the mean and dividing by the standard deviation. For EMNIST the data items are 28×28 images and we scale the pixel values to the range $[-1, 1]$. Some of the pixels have the same value for all images (i.e., the corners) so these pixels were removed and the images flattened to vectors of length 672. And for the text data in IMDB we form a bag-of-words model using the 1000 most common words after case normalisation, removal of stop words and punctuation, and stemming. The obtained word frequencies are divided by the most frequent word in each review to adjust for different review lengths, yielding real-valued vectors of length 1000. The Y-values for all datasets are scaled to approximately be within $[-0.5, 0.5]$ based on the 5^{th} and 95^{th} quantiles.

Table 3. Properties of regression methods. RR stands for robust regression.

Algorithm	Robust	Sparse	R-Package	Description
SLISE	Yes	Yes		RR with variable-size subsets
FAST-LTS [28]	Yes	No	robustbase	RR with fixed-size (50%) subsets
SPARSE-LTS [2]	Yes	Yes	robustHD	Sparse LTS solutions
MM-ESTIMATOR [34]	Yes	No	MASS	Maximum likelihood-based RR
MM-LASSO [30]	Yes	Yes	pense	Sparse MM-ESTIMATOR solutions
LAD-LASSO [33]	Maybe	Yes	MTE	Combines LAD (Least Absolute Deviation) and a LASSO penalty
LASSO [31]	Yes	No	glmnet	OLS with a L1-norm

Classifiers. We use four high-performing classifiers; a convolutional neural network (CNN), a normal neural network (NN), a logistic regression (LR), and a support vector machine (SVM), see Table 2. The classifiers are used to obtain class probabilities p_i of the given data instances. As described in Sect. 2 we transform p_i:s into linear values using the logit transformation $y_i = \log(p_i/(1-p_i))$.

Default Parameters. The two most important parameters for SLISE are the error tolerance ε and the sparsity λ. These, however, depend on the use-case and dataset and *must* be manually adjusted. The default is to use $\lambda = 0$ (no sparsity) and $\varepsilon = 0.1$ (10 % error tolerance due to the scaling mentioned above). The parameter $\beta_{\max}$ must only be large enough to make the sigmoid function essentially equivalent to a Heaviside function. As a default we use $\beta_{\max} = 30/\varepsilon^2$. The division by ε^2 is used to counteract the effects the choice of ε has on the shape of the sigmoid. The maximum approximation ratio $r_{\max}$ is used to control the step size for the graduated optimisation. We used $r_{\max} = 1.2$, which for our datasets provided good speed without sacrificing accuracy.

5.1 Robust Regression Experiments

We compare SLISE to five state-of-the-art robust regression methods (Table 3, LASSO is included as a baseline). All algorithms have been used with default settings. Not all methods support sparsity, and when they do, finding an equivalent regularisation parameter λ is difficult. Hence, unless otherwise noted, all sparse methods are used with almost no sparsity ($\lambda = 10^{-6}$).

Scalability. We first investigate the scalability of the methods. Most of the methods have similar theoretical complexities of $\mathcal{O}(nd^2)$ or $\mathcal{O}(nd^2p)$, but for the iterative methods the number of iterations p might vary. We empirically determine the running time on synthetically generated datasets with (i) $n \in \{500,\ 1\ 000,\ 5\ 000,\ 10\ 000,\ 50\ 000,\ 100\ 000\}$ items and $d = 100$ dimensions, and (ii) $d \in \{10,\ 50,\ 100,\ 500,\ 1\ 000\}$ dimensions and $n = 10\ 000$ items. The methods that support sparsity have been used with different levels of sparsity ($\lambda \in \{0, 0.01, 0.1, 0.5\}$) and the mean running times are presented. We use a cutoff-time of 10 min.

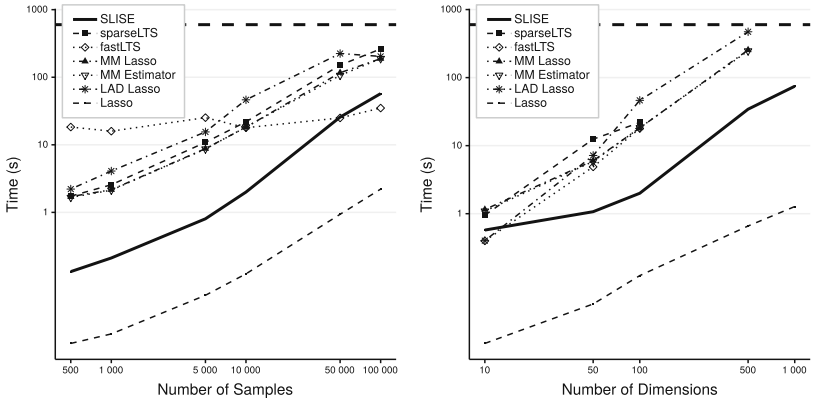


Fig. 2. Running times in seconds. Left: Varying the number of samples with fixed $d = 100$. Right: Varying the number of dimensions with fixed $n = 10\,000$. The cutoff time of 600 s is shown using a dashed horizontal line at $t = 600$.

The results are shown in Fig. 2. We observe that SLISE scales very well in comparison to the other robust regression methods. In Fig. 2 (left) SLISE outperforms all methods except FAST-LTS, which uses subsampling to keep the running time fixed for varying sizes of n. In Fig. 2 (right) we see that SLISE consistently outperforms the other robust regression methods for all $d > 10$ and it is the only robust regression method that allows us to obtain results even for a massive $10\,000 \times 1\,000$ dataset in less than 100 s (the other robust regression algorithms did not yield results within the cutoff time).

Robustness. Next we compare the methods' robustness to noise. We start with a dataset D in which a fraction δ of data items are corrupted by replacing the response variable with random noise (uniformly distributed between $\min(Y)$ and $\max(Y)$), yielding a corrupted dataset D_δ. The regression functions are learned from D_δ, after which the total sum of the residuals are determined in the clean data D. If a method is robust to noise the residuals in the clean data will be small, since the noise from the training data is ignored by the model. The results, using the PHYSICS data, are shown in Fig. 3 (left). Due to the varying subset size SLISE is able to reach higher noise fractions before breaking down than LTS. Note that at high noise fractions all methods are expected to break down.

Optimality. Finally, we demonstrate that the solution found using SLISE optimises the loss of Eq. (1). The SLISE algorithm is designed to find the largest subset such that the residuals are upper-bounded by ε. To investigate if the model found using SLISE is optimal, we determine a regression model (i.e., obtain a coefficient vector α) using each algorithm. We then calculate the value of the loss-function in Eq. (1) for each model with varying values of ε. The results, using SYNTHETIC data with $n = 1\,000$ and $d = 30$, are shown in Fig. 3 (right). All loss-values have been normalised with respect to the LASSO model at the corresponding value of

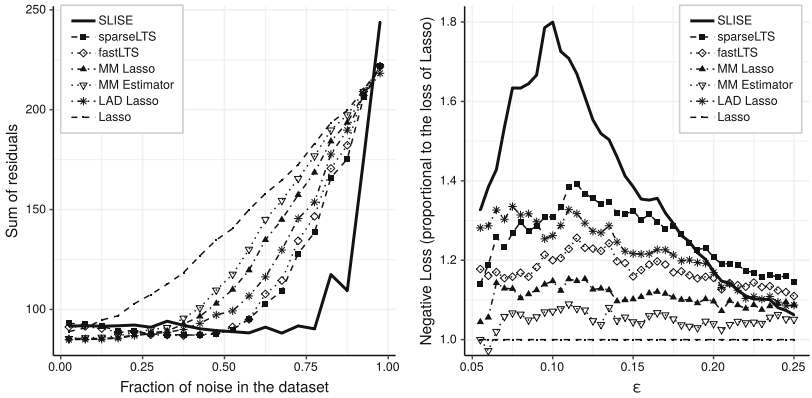


Fig. 3. Left: Robustness of SLISE to noise. The x-axis shows the fraction of noise and the y-axis the sum of the residuals. Small residuals indicate a robust method. Right: Optimality of SLISE. Negative loss-values are shown, normalised with respect to the corresponding loss for LASSO. Higher values are better.

ε and the curve for LASSO hence appears constant. SLISE consistently has the smallest loss in the region around $\varepsilon = 0.1$, as expected.

5.2 Local Explanation Experiments

Text Classification. We first compare SLISE to LIME [25], which also provides explanations in terms of sparse linear models. We use the IMDB dataset and explain a logistic regression model. LIME was used with default parameters and the number of features was set to 8. SLISE was also used with default parameters, except using $\lambda = 0.75$ to yield a sparsity comparable to LIME. The results are shown Fig. 4. The LIME-explanation surprisingly shows that the word `street` is important. `Street` indeed has a positive coefficient in the global model, but the word is quite rare, only occurring in 2.6% of all reviews. SLISE, in contrast, takes this into account and focuses on the words `great`, `fun`, and `enjoy`. The results for both algorithms are practically unchanged when all reviews with the word `street` are removed from the test dataset, i.e., LIME emphasises this word *even though it is not a meaningful discriminator for this dataset.*

Figure 5 shows a second text example with an ambiguous phrase (`not bad`). The classification is incorrect (negative), since the SVM cannot take the interaction between the words `not` and `bad` into account. The explanation from SLISE reveals this by giving negative weights to the words `wasn't` and `bad`.

Image Classification. We now demonstrate how SLISE can be used to explain the classification of a digit from EMNIST, the `2` shown in Fig. 6a. We use SLISE with default parameters, except using a sparsity of $\lambda = 2$, and a dataset with 50% images of the digit `2` and 50% images of other digits (`0`, `1`, `3-9`).

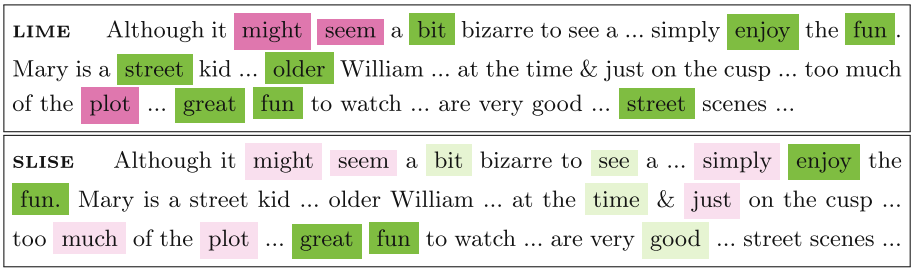


Fig. 4. Comparing LIME (top) and SLISE (bottom) with a logistic regression on the IMDB dataset. Parts without any weight from either model are left out for brevity.

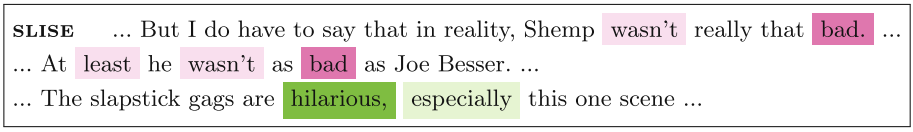
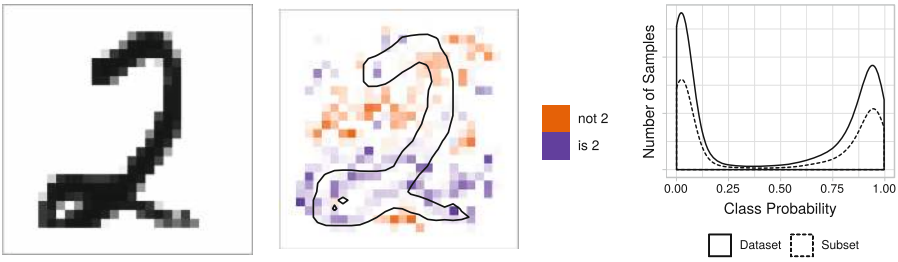


Fig. 5. SLISE explaining how the SVM does not model `not bad` as a positive phrase.

Approximation as Explanation. The linear model α approximates the opaque function (here a CNN) in the region around the item being explained. The model weights allow us to deduce features that are important for the classification. Figure 6b shows a *saliency map* in terms of the weight vector α. Each pixel corresponds to a coefficient in the α-vector and the colour of the pixel indicates its importance in distinguishing a digit 2 from other digits. Purple denotes a pixel supporting positive classification of a 2, and orange a pixel not supporting a 2. More saturated colours correspond to more important weights. We see that the long horizontal line at the bottom is important in identifying 2s, as this feature is missing in other digits. Also, the empty space in the middle-left separates 2s from other digits (i.e., if there is data here the digit is unlikely a 2).

Figure 6c shows the class probability distributions for the test dataset and the found subset S. To deduce which features in α that distinguish one class (e.g., 2s from the other digits) we must ensure that the found subset S contains items from both classes (as here in Fig. 6c), otherwise, the projection is to a linear subspace where the class probability is unchanged. During our empirical evaluation of the EMNIST dataset this did not happen.

Subset as Explanation. Unlike many other explanation methods the subset found by SLISE consists of real samples. This makes the subset interesting to examine. Figure 7a shows six digits from the subset and how the linear model interacts with them. We see why the 1 is less likely to be a 2 than the 8 (0.043 vs 0.188). Another interesting question is for which digits the approximation is not valid,



(a) (b) (c)

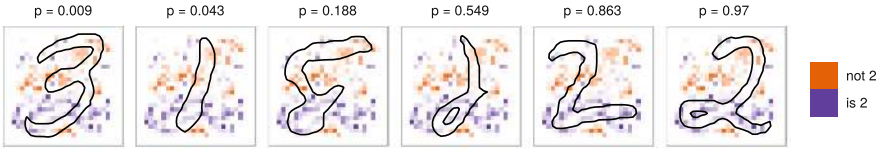
Fig. 6. (a) The digit being explained. (b) Salience map showing the regression weights of the linear model found using SLISE. The instance being explained is overlaid in the image. Purple colour indicates a weight supporting positive classification of a 2, and orange colour indicates a weight not in support of classifying the item as a 2. (c) Class probability distributions for the full dataset and for the found subset S.

in other words which digits are outside the subset. Figure 7b shows a scatterplot of the dataset used to find an explanation for the 2 (shown on a black background). The data items in the subset S lie within the corridor marked with dashed green lines. The top right contains digits to which both SLISE and the classifier assign high likelihoods of being 2s. The bottom left contains digits unlike 2s. The data items in the top left and bottom right contain items for which the local SLISE model is not valid and they are not part of the subset. We see that Z-like 2s and L-like 6s are particularly ill-suited for this approximation.

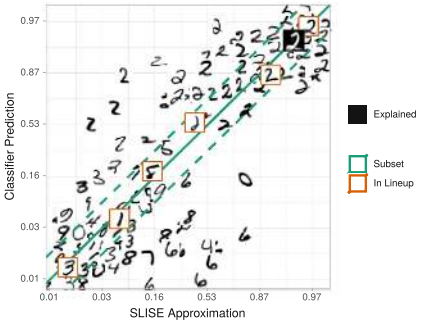
Modifying the Subset Size. The subset size controls the locality of explanations. Large subsets lead to more general explanations, while small subsets may cause overfitting on features specific to the subset. Figure 7d shows a progression of explanations for a 2 (similar to Fig. 6b) in order of decreasing subset size (from $\varepsilon = 0.64$ to $\varepsilon = 0.02$). We observe that these explanations emphasise slightly different regions due to the change in locality (and hence in the model). Note that $\varepsilon \to \infty$ is equivalent to logistic regression through the item being explained.

Modifying the Dataset. The dataset used to find the explanation can be modified in order to answer specific questions. E.g., restricting the dataset to only 2s and 3s allows investigation of what separates a 2 from a 3. This is shown in Fig. 7c. We see that 3s are distinguished by their middle horizontal stroke and the 2s by the bottom horizontal strokes ("split" due to the bottom curve of 3s).

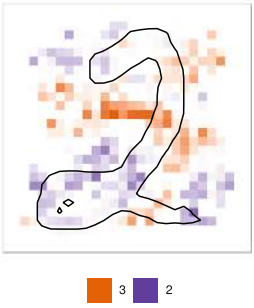
Classification of Particle Jets. Some datasets adhere to a strict generating model, this is the case for, e.g., the PHYSICS dataset, which contains particle jets extracted from simulated proton-proton collision events [8]. Here the laws of



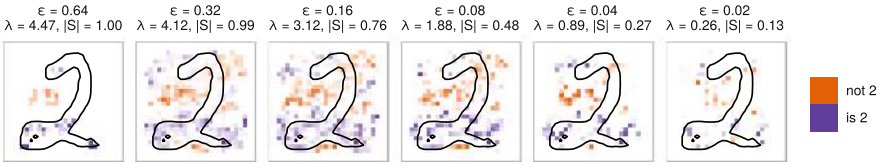
(a) Lineup of digits in the subset



(b) Scatterplot of a sample of 100 digits



(c) Separating 2 from 3



(d) Linear models with different parameters

Fig. 7. Exploring how SLISE's model interacts with other digits than the one being explained (a and b), how varying the parameters affects the explanation (d), and how modifying the dataset can answer specific questions (c).

physics must not be violated, and SLISE automatically adheres to this constraint by only using real data to construct the explanation. In Table 4 we use SLISE to explain a classification made by a neural network. The classification task in question is to decide whether the initiating particle of the jet was a *quark* or a *gluon*. The total energy of the jet is on average distributed differently among its constituents depending on the jet's origin [7]. Here, the SLISE explanation shows the importance of the energy distribution variable `QG_ptD`.

Table 4. SLISE explanation for why an example in the PHYSICS dataset is a quark jet.

	Pt	Girth	QG_ptD	QG_axis2	QG_mult
Jet	1196	0.020	0.935	0.002	16
$\boldsymbol{\alpha}$	0.01	−0.05	0.18	−0.02	0

6 Conclusions

This paper introduced the SLISE algorithm, which can be used both for robust regression and to explain classifier predictions. SLISE extends existing robust regression methods, especially in terms of scalability, important in modern data analysis. In contrast to other methods, SLISE finds a subset of variable size, adjustable in terms of the error tolerance ε. SLISE also yields sparse solutions.

SLISE yields meaningful and interpretable explanations for classifier decisions and can be used without modification for various types of data and without the need to evaluate the classifier outside the data set. This simplicity is important as it provides consistent operation across data domains. It is important to take the data distribution into account, and if the data has a strict generating model it is also crucial not to perturb the data. The local explanations provided by SLISE take the interaction between the model and the distribution of the data into account, which means that even simple global models might have non-trivial local explanations. Future work includes investigating various initialisation schemes for SLISE (currently an OLS solution is used).

Acknowledgements. Supported by the Academy of Finland (decisions 326280 and 326339). We acknowledge the computational resources provided by Finnish Grid and Cloud Infrastructure [12].

References

1. Adler, P., et al.: Auditing black-box models for indirect influence. In: ICDM, pp. 1–10 (2016)
2. Alfons, A., Croux, C., Gelper, S.: Sparse least trimmed squares regression for analyzing high-dimensional large data sets. Ann. Appl. Stat. **7**(1), 226–248 (2013)
3. Amaldi, E., Kann, V.: The complexity and approximability of finding maximum feasible subsystems of linear relations. Theor. Comput. Sci. **147**(1), 181–210 (1995)
4. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties, 2nd edn. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-58412-1
5. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.: How to explain individual classification decisions. JMLR **11**, 1803–1831 (2010)
6. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In: SIGKDD, pp. 1721–1730 (2015)

7. CMS Collaboration: Performance of quark/gluon discrimination in 8 TeV pp data. CMS-PAS-JME-13-002 (2013)
8. CMS Collaboration: Dataset QCD_Pt15to3000_TuneZ2star_Flat_8TeV_pythia6 in AODSIM format for 2012 collision data. CERN Open Data Portal (2017)
9. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters. arXiv:1702.05373 (2017)
10. Datta, A., Sen, S., Zick, Y.: Algorithmic transparency via quantitative input influence: theory and experiments with learning systems. In: IEEE S&P, pp. 598–617 (2016)
11. Donoho, D.L., Huber, P.J.: The notion of breakdown point. In: A festschrift for Erich L. Lehmann, pp. 157–184 (1983)
12. Finnish Grid and Cloud Infrastructure, urn:nbn:fi:research-infras-2016072533
13. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. arXiv:1704.03296 (2017)
14. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. arXiv:1805.10820 (2018)
15. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. CSUR **51**(5), 93:1–93:42 (2018). https://doi.org/10.1145/3236009
16. Henelius, A., Puolamäki, K., Boström, H., Asker, L., Papapetrou, P.: A peek into the black box: exploring classifiers by randomization. DAMI **28**(5–6), 1503–1529 (2014)
17. Henelius, A., Puolamäki, K., Ukkonen, A.: Interpreting classifiers through attribute interactions in datasets. In: WHI, pp. 8–13 (2017)
18. Komiske, P.T., Metodiev, E.M., Schwartz, M.D.: Deep learning in color: towards automated quark/gluon jet discrimination. JHEP **01**, 110 (2017)
19. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: a joint framework for description and prediction. In: SIGKDD, pp. 1675–1684 (2016)
20. Loh, P.L.: Scale calibration for high-dimensional robust regression. arXiv preprint arXiv:1811.02096 (2018)
21. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: NIPS, pp. 4765–4774 (2017)
22. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: ACL HLT, pp. 142–150 (2011)
23. Mobahi, H., Fisher, J.W.: On the link between gaussian homotopy continuation and convex envelopes. In: Tai, X.-C., Bae, E., Chan, T.F., Lysaker, M. (eds.) EMMCVPR 2015. LNCS, vol. 8932, pp. 43–56. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14612-6_4
24. Molnar, C.: Interpretable Machine Learning (2019). https://christophm.github.io/interpretable-ml-book
25. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: SIGKDD, pp. 1135–1144 (2016)
26. Rousseeuw, P.J.: Least median of squares regression. J. Am. Stat. Assoc. **79**(388), 871–880 (1984)
27. Rousseeuw, P.J., Hubert, M.: Robust statistics for outlier detection. WIRES Data Min. Knowl. Discov. **1**(1), 73–79 (2011)
28. Rousseeuw, P.J., Van Driessen, K.: An algorithm for positive-breakdown regression based on concentration steps. In: Gaul, W., Opitz, O., Schader, M. (eds.) Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 335–346. Springer, Heidelberg (2000)

29. Schmidt, M., Berg, E., Friedlander, M., Murphy, K.: Optimizing costly functions with simple constraints: a limited-memory projected quasi-newton algorithm. In: AISTATS, pp. 456–463 (2009)
30. Smucler, E., Yohai, V.J.: Robust and sparse estimators for linear regression models. Comput. Stat. Data Anal. **111**, 116–130 (2017)
31. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Series. B Stat. Methodol. **58**(1), 267–288 (1996)
32. Ustun, B., Traca, S., Rudin, C.: Supersparse linear integer models for interpretable classification. arXiv:1306.6677v6 (2014)
33. Wang, H., Li, G., Jiang, G.: Robust regression shrinkage and consistent variable selection through the LAD-Lasso. J. Bus. Econ. Stat. **25**(3), 347–355 (2007)
34. Yohai, V.J.: High breakdown-point and high efficiency robust estimates for regression. Ann. Stat. **15**(2), 642–656 (1987). https://doi.org/10.1214/aos/1176350366