Feature Selection using Stochastic Gates

Yutaro Yamada [*1] **Ofir Lindenbaum** [*2] **Sahand Negahban** [1] **Yuval Kluger** [2 3]

Abstract

Feature selection problems have been extensively studied in the setting of linear estimation (e.g. LASSO), but less emphasis has been placed on feature selection for non-linear functions. In this study, we propose a method for feature selection in neural network estimation problems. The new procedure is based on probabilistic relaxation of the ℓ_0 norm of features, or the count of the number of selected features. Our ℓ_0-based regularization relies on a continuous relaxation of the Bernoulli distribution; such relaxation allows our model to learn the parameters of the approximate Bernoulli distributions via gradient descent. The proposed framework simultaneously learns either a nonlinear regression or classification function while selecting a small subset of features. We provide an information-theoretic justification for incorporating Bernoulli distribution into feature selection. Furthermore, we evaluate our method using synthetic and real-life data to demonstrate that our approach outperforms other commonly used methods in both predictive performance and feature selection.

1. Introduction

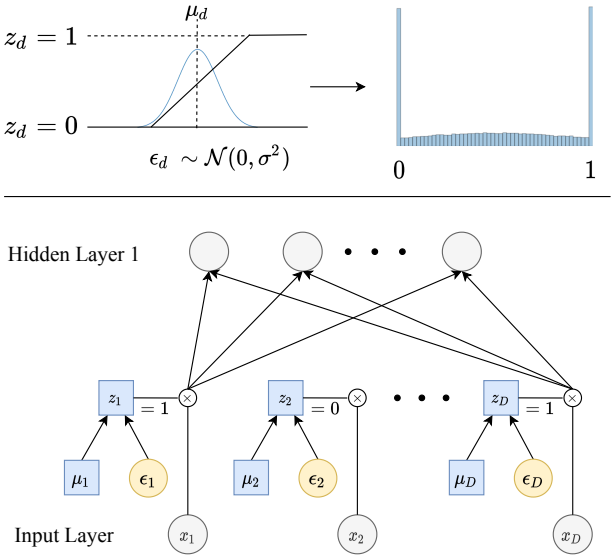
Feature selection is a fundamental task in machine learning and statistics. Selecting a subset of relevant features may result in several potential benefits: reducing experimental costs (Min et al., 2014), enhancing interpretability (Ribeiro et al., 2016), speeding up computation, reducing memory and even improving model generalization on unseen data (Chandrashekar & Sahin, 2014). For example, in biomedical studies, machine learning can provide effective diagnostics or prognostics models. However, the number of features (e.g., genes or proteins) often exceeds the number of samples. In this setting, feature selection can lead to improved risk assessment and provide meaningful biological insights. While neural networks are good candidates for learning diagnostics models, identifying relevant features while building compact predictive models remains an open challenge.

Feature selection methods are classified into three major categories: filter methods, wrapper methods, and embedded methods. **Filter methods** attempt to remove irrelevant features prior to learning a model. These methods filter features using a per-feature relevance score that is created based on statistical measures (Battiti, 1994; Peng et al., 2005; Estévez et al., 2009; Song et al., 2007; 2012; Chen et al., 2017). Wrapper methods (Kohavi & John, 1997b; Stein et al., 2005; Zhu et al., 2007; Reunanen, 2003; Allen, 2013) use the outcome of a model to determine the relevance of each feature. **Wrapper methods** require recomputing the model for each subset of features and, thus, become computationally expensive, especially in the context of deep neural networks (Verikas & Bacauskiene, 2002; Kabir et al., 2010; Roy et al., 2015). **Embedded methods** aim to remove this burden by learning the model while simultaneously selecting the subset of relevant features. The Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996) is a well-known embedded method, whose objective is to minimize the loss while enforcing an ℓ_1 constraint on the weights of the features. LASSO is scalable and widely used (Hans, 2009; Li et al., 2011; 2006), but it is restricted to the domain of linear functions and suffers from shrinkage of the model parameters. It seems natural to extend the LASSO using neural networks; however, gradient descent on an ℓ_1 regularized objective neither performs well in practice nor sparsifies the input layer (Li et al., 2016; Scardapane et al., 2017; Feng & Simon, 2017).

To overcome these limitations, we develop a fully *embedded feature selection* method for nonlinear models. Our method improves upon the LASSO formulation by: a) capturing nonlinear interactions between features via neural network modeling and b) employing an ℓ_0-like regularization using gates with weights parametrized by a smooth variant of a Bernoulli distribution. These two improvements are jointly formulated as a fully differentiable neural network that provides a solution to the important long-standing problem of feature selection for nonlinear functions.

[*]Equal contribution [1]Department of Statistics and Data Science, Yale University, Connecticut, USA [2]Program in Applied Mathematics, Yale University, Connecticut, USA [3]School of Medicine, Yale University, Connecticut, USA. Correspondence to: Yuval Kluger <yuval.kluger@yale.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.



Algorithm 1 STG: Feature selection using stochastic gates

Input: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, target variables $\boldsymbol{y} \in \mathbb{R}^N$, regularization parameter λ, number of epochs M, learning rate γ.
Output: Trained model $f_{\boldsymbol{\theta}}$ and parameter $\boldsymbol{\mu} \in \mathbb{R}^D$.

1: Initialize the model parameter $\boldsymbol{\theta}$. Set $\boldsymbol{\mu} = 0.5$.
2: **for** $k = 1, ..., K$ **do**
3: **for** $d = 1, ..., D$ **do**
4: Sample $\epsilon_d^{(k)} \sim N(0, \sigma^2)$
5: Compute $z_d^{(k)} = \max(0, \min(1, \mu_d + \epsilon_d^{(k)}))$
6: **end for**
7: **end for**
8: Compute the loss $\hat{L} = \frac{1}{NK} \sum_{n,k} L(f_\theta(\boldsymbol{x}_n \odot \boldsymbol{z}^{(k)}), y_n)$
9: Compute the regularization $R = \lambda \sum_d \Phi(\frac{\mu_d}{\sigma})$
 $\boldsymbol{\theta} := \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} \hat{L}$ and $\boldsymbol{\mu} := \boldsymbol{\mu} - \gamma \nabla_{\boldsymbol{\mu}} (\hat{L} + R)$
10: Repeat M epochs

Figure 1. Top left: Each stochastic gate z_d is drawn from the STG approximation of the Bernoulli distribution (shown as the blue histogram on the right). Specifically, z_d is obtained by applying the hard-sigmoid function to a mean-shifted Gaussian random variable (step 5 in algorithm 1). Bottom left: The z_d stochastic gate is attached to the x_d input feature, where the trainable parameter μ_d controls the probability of the gate being active. Right: Pseudocode of our algorithm for feature selection. See the supplementary material for a discussion of σ and λ's selection.

Specifically, our contributions are as follows:

- We identify the limitations of the logistic-distribution-based Bernoulli relaxation (Maddison et al., 2016; Jang et al., 2017; Louizos et al., 2017) in feature selection and present a Gaussian-based alternative termed stochastic gate (STG), which is better in terms of model performance and consistency of feature selection.
- We develop an embedded nonlinear feature selection method by introducing the stochastic gates to the input layer (the feature space) of a neural network.
- We justify our probabilistic approach by analyzing the constrained Mutual Information maximization objective of feature selection.

We demonstrate the advantages of our method for classification, regression, and survival analysis tasks using numerous examples.

Notation: Vectors are denoted by **bold** lowercase letters $\boldsymbol{x}$ and random vectors as bold uppercase letters $\boldsymbol{X}$. Scalars are denoted by lower case letters y, while random variables are uppercase Y. A set is represented by a script font $\mathcal{S}$. For example the n^{th} vector-valued observation is denoted as $\boldsymbol{x}_n$ whereas X_d represents the d^{th} feature of the vector-valued random variable $\boldsymbol{X}$. Let $[n] = 1, 2, \ldots, n$. For a set $\mathcal{S} \subset [\mathcal{D}]$ let the vector $\boldsymbol{s} \in \{0, 1\}^D$ be the characteristic function for the set. That is $s_i = 1$ if $i \in \mathcal{S}$ and 0 otherwise. For two vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ we denote $\boldsymbol{x} \odot \boldsymbol{z}$ to be the element-wise product between $\boldsymbol{x}$ and $\boldsymbol{z}$. Thus, if we let $\boldsymbol{s} \in \{0, 1\}^D$ be the characteristic vector of $\mathcal{S}$, then we may define $\boldsymbol{x}_\mathcal{S} = \boldsymbol{x} \odot \boldsymbol{s}$. The ℓ_1 norm of $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|_1 = \sum_{i=1}^D |x_i|$. Finally, the ℓ_0 norm of $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|_0$ and counts the total number of non-zero entries in the vector $\boldsymbol{x}$.

2. Problem Setup and Background

Let $\mathcal{X} \subset \mathbb{R}^D$ be the input domain with corresponding response domain $\mathcal{Y}$. Given realizations from some unknown data distribution $P_{X,Y}$, the goal of embedded feature selection methods is to simultaneously select a subset of indices $\mathcal{S} \subset \{1, ...D\}$ and construct a model $f_{\boldsymbol{\theta}} \in \mathcal{F}$ that predicts Y based on the selected features $\boldsymbol{X}_\mathcal{S}$.

Given a loss L, the selection of features $\mathcal{S} \subset [\mathcal{D}]$, and choice of parameters $\boldsymbol{\theta}$ can be evaluated in terms of the following risk:

$$R(\boldsymbol{\theta}, \boldsymbol{s}) = \mathbb{E}_{X,Y} L(f_\theta(\boldsymbol{X} \odot \boldsymbol{s}), Y), \tag{1}$$

where we recall that $\boldsymbol{s} = \{0, 1\}^D$ is a vector of indicator variables for the set $\mathcal{S}$, and $\odot$ denotes the point-wise product. Embedded feature selection methods search for parameters $\boldsymbol{\theta}$ and $\boldsymbol{s}$ that minimize $R(\boldsymbol{\theta}, \boldsymbol{s})$ such that $\|\boldsymbol{s}\|_0$ is small compared to D.

2.1. Feature Selection for Linear Models

We first review the feature selection problem in the linear setting for a least squares loss. Given observations $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, a natural objective derived from (1) is the

constrained empirical risk minimization

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\theta}^T\boldsymbol{x}_n - y_n)^2 \quad \text{s.t. } \|\boldsymbol{\theta}\|_0 \leq k. \tag{2}$$

Since the above problem is intractable, several authors replace the ℓ_0 constraint with a surrogate function, $\Omega(\boldsymbol{\theta}) : \mathbb{R}^D \rightarrow \mathbb{R}_+$, designed to penalize the number of selected features in $\boldsymbol{\theta}$. A popular choice for Ω is the ℓ_1 norm, which yields a convex problem and more precisely the LASSO optimization (Tibshirani, 1996). Computationally efficient algorithms for solving the LASSO problem have been proposed (Tibshirani, 1996; Nesterov, 2013; Qian et al., 2019). While the original LASSO focuses on the constrained optimization problem, the regularized least squares formulation, which is often used in practice, yields the following minimization objective:

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\theta}^T\boldsymbol{x}_n - y_n)^2 + \lambda\|\boldsymbol{\theta}\|_1. \tag{3}$$

The hyperparameter λ trades off the amount of regularization versus the fit of the objective[1]. The ℓ_1-regularized method is effective for feature selection and prediction; however, it achieves this through shrinkage of the coefficients and is restricted to linear models. To avoid shrinkage, non-convex choices for Ω have been proposed (Fan & Li, 2001). As demonstrated in several studies (Huang et al., 2007; Laporte et al., 2013; Zhu et al., 2017), non-convex regularizers perform well both theoretically and empirically in prediction and feature selection.

Our goal is to develop a regularization technique that both avoids shrinkage and performs feature selection while learning a nonlinear function. To allow nonlinearities, Kernel methods have been considered (Yamada et al., 2014), but scale quadratically in the number of observations. An alternative approach is to model $f_{\boldsymbol{\theta}}$ using a neural network with ℓ_1 regularization on the input weights (Li et al., 2016; Scardapane et al., 2017; Feng & Simon, 2017). However, in practice, introducing an ℓ_1 penalty into gradient descent does not sparsify the weights and requires post-training thresholding. Below, we present our method that applies a differentiable approximation of an ℓ_0 penalty on the first layer of a neural network.

3. Proposed Method

To implement an ℓ_0 regularization to either linear or nonlinear models, we introduce a probabilistic and computationally efficient neural network approach. It is well known that an exact ℓ_0 regularization is computationally expensive and intractable for high dimensions. Moreover, the ℓ_0 norm cannot be incorporated into a gradient descent based optimization. To overcome these limitations, a probabilistic formulation provides a compelling alternative. Specifically, we introduce Bernoulli gates applied to each of the d input nodes of a neural network. A random vector $\tilde{\boldsymbol{S}}$ represents these Bernoulli gates, whose entries are independent and satisfy $\mathbb{P}(\tilde{S}_d = 1) = \pi_d$ for $d \in [D]$, respectively. If we denote the empirical expectation over the observations as $\hat{\mathbb{E}}_{X,Y}$, then, the empirical regularized risk (Eq. 1) becomes

$$\hat{R}(\boldsymbol{\theta}, \boldsymbol{\pi}) = \hat{\mathbb{E}}_{X,Y}\mathbb{E}_{\tilde{S}}\left[L(f_\theta(\boldsymbol{X} \odot \tilde{\boldsymbol{S}}), Y) + \lambda||\tilde{\boldsymbol{S}}||_0\right], \tag{4}$$

where $\mathbb{E}_{\tilde{S}}||\tilde{\boldsymbol{S}}||_0$ boils down to the sum of Bernoulli parameters $\sum_{d=1}^{D}\pi_d$. Note that, if we constrain $\pi_d \in \{0, 1\}$, this formulation is equivalent to the constrained version of equation (1), with a regularized penalty on cardinality rather than an explicit constraint. Moreover, this probabilistic formulation converts the combinatorial search to a search over the space of Bernoulli distribution parameters (also motivated in Section 4). Thus, the problem of feature selection translates to finding $\boldsymbol{\theta}^*$ and $\boldsymbol{\pi}^*$ that minimize the empirical risk based on the formulation in Eq. 4.

Minimization of the empirical risk via gradient descent seems like a natural way to simultaneously determine the model parameters $\boldsymbol{\theta}^*$ and Bernoulli-based feature selection parameters $\boldsymbol{\pi}^*$. However, optimization of a loss function, which includes discrete random variables, suffers from high variance (see supplementary for more details and (Mnih & Rezende, 2016)). To overcome this limitation, several authors have proposed using a continuous approximation of discrete random variables, such as the Concrete (Jang et al., 2017; Maddison et al., 2016) or Hard-Concrete (HC) (Louizos et al., 2017).

We observed that the HC still suffers from high variance and, thus, is not suited for the task of feature selection. Therefore, we develop an empirically superior continuous distribution that is fully differentiable and implemented only to activate or deactivate the gates linking each feature (node) to the rest of the network. Our method provides an embedded feature selection algorithm with superior results in terms of both accuracy and capturing informative features compared with the state-of-the-art.

3.1. Bernoulli Continuous Relaxation for Feature Selection

Feature selection requires stability in the selected set of features. The use of logistic distributions such as the Concrete (Jang et al., 2017; Maddison et al., 2016) and HC (Louizos et al., 2017) induces high variance in the approximated Bernoulli variables due to the heavy-tailedness, which often leads to inconsistency in the set of selected features. To

[1] λ has a one-to-one correspondence to k in the convex setting via Lagrangian duality.

address such limitations, we propose a Gaussian-based continuous relaxation for the Bernoulli variables $\tilde{S}_d$ for $d \in [D]$. We refer to each relaxed Bernoulli variable as a stochastic gate (STG) defined by $z_d = \max(0, \min(1, \mu_d + \epsilon_d))$, where ϵ_d is drawn from $\mathcal{N}(0, \sigma^2)$ and σ is fixed throughout training. This approximation can be viewed as a clipped, mean-shifted, Gaussian random variable as shown in the left part of Fig. 1. Furthermore, the gradient of the objective with respect to μ_d can be computed via the chain rule, which is commonly known as the reparameterization trick (Miller et al., 2017; Figurnov et al., 2018).

We can now write our objective as a minimization of the empirical risk $\hat{R}(\boldsymbol{\theta}, \boldsymbol{\mu})$:

$$\min_{\boldsymbol{\theta}, \boldsymbol{\mu}} \hat{\mathbb{E}}_{X,Y} \mathbb{E}_Z \left[L(f_\theta(\boldsymbol{X} \odot \boldsymbol{Z}), Y) + \lambda ||\boldsymbol{Z}||_0 \right], \quad (5)$$

where $\boldsymbol{Z}$ is a random vector with D independent variables z_d for $d \in [D]$. Under the continuous relaxation, the expected regularization term in the objective $\hat{R}(\boldsymbol{\theta}, \boldsymbol{\mu})$ (Eq. 5) is simply the sum of the probabilities that the gates $\{z_d\}_{d=1}^D$ are active or $\sum_{d \in [D]} \mathbb{P}(z_d > 0)$. This sum is equal to $\sum_{d=1}^D \Phi\left(\frac{\mu_d}{\sigma}\right)$, where Φ is the standard Gaussian CDF. To optimize the empirical surrogate of the objective (Eq. 5), we first differentiate it with respect to $\boldsymbol{\mu}$. This computation is done using a Monte Carlo sampling gradient estimator which gives

$$\frac{1}{K} \sum_{k=1}^{K} \left[L'(\boldsymbol{z}^{(k)}) \frac{\partial z_d^{(k)}}{\partial \mu_d} \right] + \lambda \frac{\partial}{\partial \mu_d} \Phi\left(\frac{\mu_d}{\sigma}\right),$$

where K is the number of Monte Carlo samples. Thus, we can update the parameters μ_d for $d \in [D]$ via gradient descent.

Altogether, Eq. 5 is optimized using SGD over the model parameters $\boldsymbol{\theta}$ and the parameters $\boldsymbol{\mu}$, where the latter substitute the parameters $\boldsymbol{\pi}$ in Eq. 4. See Algorithm 1 for a pseudocode of this procedure.

To remove the stochasticity from the learned gates after training, we set $\hat{z}_d = \max(0, \min(1, \mu_d))$, which informs what features are selected. In our experiments, for all synthetic datasets, we observe that the coordinates of $\hat{z}$ converge to 0 or 1. However, when the signal is weak (e.g. the class samples are not separated) training the gates until convergence may cause overfitting of the model parameters. In these cases, setting a cutoff value (e.g. 0.5) and performing early stopping is beneficial. In the supplementary material, we discuss our choice of σ.

4. Connection to Mutual Information

In this section, we use a Mutual Information (MI) perspective to show an equivalence between a constrained ℓ_0-based optimization for feature selection and an optimization over Bernoulli distribution parameters.

4.1. Mutual Information based objective

From an information theoretic standpoint, the goal of feature selection is to find the subset of features $\mathcal{S}$ that has the highest Mutual Information (MI) with the target variable Y. MI between two random variables can be defined as $I(\boldsymbol{X}; Y) = H(Y) - H(Y|\boldsymbol{X})$, where $H(Y)$ and $H(Y|\boldsymbol{X})$ are the entropy of $p_Y(Y)$ and the conditional entropy of $p_{Y|\mathbf{X}}(Y|\boldsymbol{X})$, respectively (Cover & Thomas, 2006). We can then formulate the task as selecting $\mathcal{S}$ such that the mutual information between $\boldsymbol{X}_\mathcal{S}$ and Y is maximized:

$$\max_{\mathcal{S}} I(\boldsymbol{X}_\mathcal{S}; Y) \quad \text{s.t. } |\mathcal{S}| = k, \quad (6)$$

where k is the hypothesized number of relevant features.

4.2. Introducing randomness

Under mild assumptions, we show that one can replace the deterministic search over the set $\mathcal{S}$ (or corresponding indicator vector $\boldsymbol{s}$) with a search over the parameters of the distributions that model $\boldsymbol{s}$. Our proposition is based on the following two assumptions:

Assumption 1: There exists a subset of indices $\mathcal{S}^*$ with a cardinality equal to k such that for any $i \in \mathcal{S}^*$ we have $I(X_i; Y|\boldsymbol{X}_{\setminus\{i\}}) > 0$.

Assumption 2: $I(\boldsymbol{X}_{\mathcal{S}^{*c}}; Y|\boldsymbol{X}_{\mathcal{S}^*}) = 0$.

Discussion of assumptions: Assumption 1 states that including an element from $\mathcal{S}^*$ improves prediction accuracy. This assumption is equivalent to stating that feature i is strongly relevant (Kohavi & John, 1997a; Brown et al., 2012). Assumption 2 simply states that $\mathcal{S}^*$ is a superset of the Markov Blanket of the variable Y (Brown et al., 2012). The assumptions are quite benign. For instance, they are satisfied if $\boldsymbol{X}$ is drawn from a Gaussian with a non-degenerate covariance matrix and $Y = f(\boldsymbol{X}_{\mathcal{S}^*}) + w$, where w is noise independent of $\boldsymbol{X}$ and f is not degenerate. With these assumptions in hand, we may present our result.

Proposition 1. *Suppose that the above assumptions hold for the model. Then, solving the optimization* (6) *is equivalent to solving the optimization*

$$\max_{\mathbf{0} \le \boldsymbol{\pi} \le \mathbf{1}} I(\boldsymbol{X} \odot \tilde{\boldsymbol{S}}; \boldsymbol{Y}) \quad \textit{s.t.} \quad \sum_i \mathbb{E}[\tilde{S}_i] \le k, \quad (7)$$

where the coordinates $\tilde{S}_i$ are drawn independently at random from a Bernoulli distribution with parameter π_i.

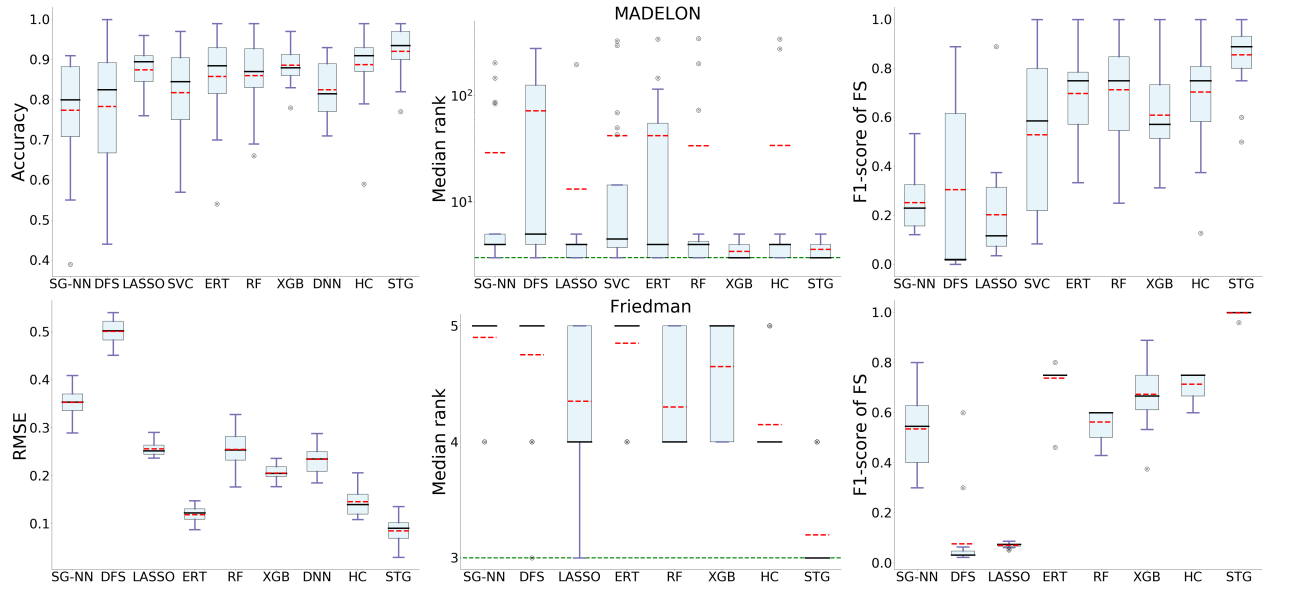


Figure 2. Evaluation of the proposed method using synthetic data. Top row: classification using the MADELON dataset with 5 informative and 495 nuisance features. Bottom row: regression using a modified version of the Friedman dataset, which also consists of 5 informative and 495 nuisance features. Left column: Accuracy/root mean squared error (RMSE). Middle column: median rank of informative features. Right column: F1-score that measures success in retrieving the informative features. We also evaluated the accuracy and MSE of a neural network with no feature selection (DNN). Black bars represent the medians, and dashed red lines are the means. In the middle column, dashed green lines are the optimal median ranks.

Due to length constraints, we leave the proof of this proposition and how it bridges the MI maximization (6) and risk minimization (2) to the supplementary material.

5. Related Work

The three most related works to this study are (Louizos et al., 2017), (Chen et al., 2018) and (Chang et al., 2017). In (Louizos et al., 2017), they introduce the Hard-Concrete (HC) distribution as a continuous surrogate for Bernoulli distributions in the context of model compression. The authors demonstrate that applying the HC to all of the weights leads to fast convergence and improved generalization. They did not evaluate the HC for the task of feature selection where stability of the selection is an important property.

In (Chen et al., 2018), the Concrete distribution is used to develop a framework for interpreting pre-trained models. Their method is focused on finding a subset of features given a particular sample and, therefore, is not appropriate for general feature selection. In (Chang et al., 2017), the Concrete distribution is used for feature ranking. The method is not fully embedded and requires model retraining to achieve feature selection.

Bernoulli relaxation techniques that are based on logistic distributions (e.g. Concrete/Gumbel-Softmax and HC) are not suitable for feature selection. Specifically, the use of the Concrete/Gumbel-Softmax distribution ranks features but retains all of them (no feature selection). In contrast to our Gaussian-based relaxation of Bernoulli distributions (STG), the logistic-based HC yields high-variance gradient estimates. For model sparsification, this high variance is not problematic because the sparsity pattern within the network does not matter as long as the method achieves enough sparsity as a whole. For feature selection based on the HC approach, however, the subsets of selected features at different runs vary substantially. Thus, the stability of the HC-based feature selection is poor; see Section 8. Furthermore, higher gradient variance will also result in a slower SGD convergence, which has been demonstrated empirically in Section 6 and the supplementary material.

6. Experiments

Here, we evaluate our proposed embedded feature selection method. We implemented [2] it using both the STG and HC (Louizos et al., 2017) distributions and tested on several artificial and real datasets. We compare our method with several classification and regression algorithms including embedded methods such as LASSO (Tibshirani, 1996), linear support vector classification (SVC) (Chang & Lin, 2008), deep feature selection (DFS) (Li et al., 2016) and group-sparse regularization for deep neural networks (SG-NN) (Scardapane et al., 2017). Our method is also compared with leading tree-based wrapper methods - extremely randomized trees (ERT) (Rastogi & Shim, 2000), random forests (RF) (Díaz-Uriarte & De Andres, 2006) and XGBoost (XGB) (Chen & Guestrin, 2016). See the supplementary material for details

[2] https://github.com/runopti/stg

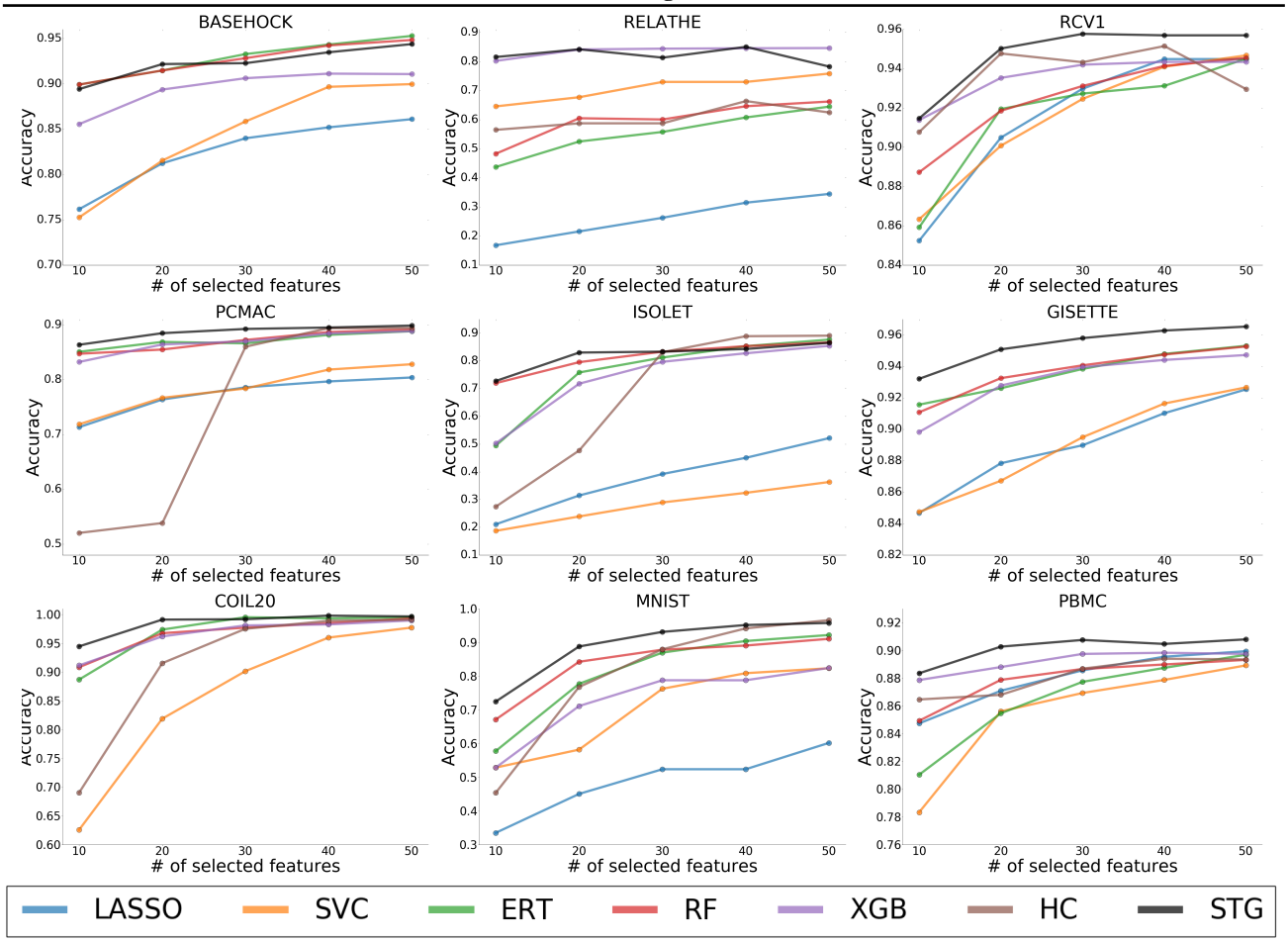


Figure 3. Classification accuracy vs. number of selected features. Descriptions of the 9 datasets appear in Table 1.

on the hyper-parameters of all methods.

6.1. Synthetic data in the $D > N$ regime

Now we present empirical results in the challenging regime where the number of features exceeds the number of samples ($D > N$). We use synthetic datasets with informative and irrelevant nuisance variables. We begin with the MADELON dataset, a hard classification problem suggested in the NIPS 2003 feature selection challenge (Guyon et al., 2005). This dataset consists of 5 informative features and 495 nuisance features. See the supplementary material for additional details on the MADELON dataset.

Next, we present a regression scenario based on a modification of the Friedman regression dataset (Friedman, 1991). In this dataset, all 500 variables are uniformly distributed in $[0, 1]$, and the response is defined by the following function:

$$Y = 10\sin(X_1X_2)^2 + 20X_3^2 + 10\ \text{sign}(X_4X_5 - 0.2) + \xi,$$

where ξ is drawn from $\mathcal{N}(0, 1)$. Then, Y is centered and divided by its maximal value.

For the above synthetic classification and regression datasets, we generate 600 samples of which we use 450 for training, 50 for validation and 100 for a test set. The hyper-parameter (controlling the number of selected features) for each method is optimized based on the validation performance. The experiment is repeated 20 times, and the accuracy/root mean squared error (RMSE), median rank and F1-score that measures feature selection performance are presented in Fig. 2. To compute the median ranks, we utilize the scores that each method assigns to the features. We then rank all features based on these scores and compute the median of the ranks of the 5 informative features. Thus, the optimal median rank in these examples is 3. The F1-score measure for feature selection is defined as $F1 = 2(\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall})$, where precision and recall are computed by comparing the selected/removed features to the informative/nuisance features. For example, a model which retains all of the features has a recall of 1 with a precision of $5/500$.

The results presented in Fig. 2 demonstrate our embedded method's ability to learn a powerful predictive model in the regime of $D > N$, where the majority of variables are not informative. Even though our median rank performance

is comparable to tree-based methods, we outperform all baseline in F1-scores. This demonstrates that our embedded method is a strong candidate for the task of finding complex relations in high dimensional data.

We encourage the reader to look at the supplementary material, where we provide additional experiments including a more challenging variant of the MADELON data, the XOR data, the two-moons data and 3 artificial regression datasets based on (Gregorová et al., 2018).

6.2. Classification on real world data

We now turn our attention towards using several real-world labeled datasets to evaluate our method. Most of the datasets are collected from the ASU feature selection database available online [3]. The dimensions, sample size and domains of the data are versatile and are detailed in Table 1. On all of the datasets - except MNIST (LeCun et al., 1998), RCV-1 (Lewis et al., 2004) and the PBMC (Zheng et al., 2017) - we perform 5-fold cross validation and report the average accuracies vs. the number of features that the model uses for several baselines. For MNIST, RCV-1 and the PBMC - we used prefixed training and testing sets which are described in the supplementary material. The results are presented in Fig. 3.

Compared to the alternative linear embedded methods (i.e. LASSO and SVC), the nonlinearity of our method provides a clear advantage. While there are regimes in which the tree-based methods slightly outperform our method, they require retraining a model based on the selected features; however, our method is only trained once and learns the model and features simultaneously. This experiment also demonstrates that the STG is more suited for the task of feature selection than the HC. Note that in this experiment we did not include the DFS and SG-NN, as they do not sparsify the weights and, therefore, cannot be evaluated vs. the number of selected features.

Due to lack of space we leave the real word regression experiments to the supplementary material.

7. Cox Proportional Hazard Models for Survival Analysis

A standard model for survival analysis is the Cox Proportional Hazard Model. In (Katzman et al., 2018), the authors proposed DeepSurv that extends the Cox model to neural networks. We incorporate our method into DeepSurv to see how our procedure improves survival analysis based on gene expression profiles from the breast cancer dataset called METABRIC (Curtis et al., 2012) (along with additional commonly used clinical variables.) See the supplementary material for more details about the dataset and experimental setup.

We compare our method Cox-STG with four other methods: a Cox model with ℓ_1 regularization (Cox-LASSO), Random Survival Forest (RSF) (Ishwaran et al., 2008), Cox-HC, and the original DeepSurv. We evaluate the predictive ability of the learned models based on the concordance index (CI), a standard performance metric for model assessment in survival analysis; it measures the agreement between the rankings of the predicted and observed survival times. The performance of each model in terms of the CI and the number of selected features are reported in Table 2. The Cox-STG method outperforms the other baselines indicating that our approach identifies a small number of informative variables while maintaining high predictive performance.

8. Evaluating stochastic regularization schemes

In this section, we elaborate on two aspects of our proposed method that lead to performance gains: (i) benefits of our non-convex regularization and injected noise, and (ii) advantages of the Gaussian based STG over the logistic based HC distribution in terms of feature selection performance.

To demonstrate these performance gains, we perform a controlled experiment in a linear regression setting. We first generate the data matrix, $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, $D = 64$, with values randomly drawn from $\mathcal{N}(0, 1)$ and construct the response variable

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{w}, \tag{8}$$

where the values of the noise $\boldsymbol{w}_i, i = 1, ..., N$ are drawn independently from $\mathcal{N}(0, 0.5)$. As suggested by (Wainwright, 2009), we use a known sparsity $\|\boldsymbol{\beta}^*\|_0 = k$, set by $k = \lceil 0.4D^{0.75} \rceil = 10$. For each number of samples N in the range $[10, 250]$, we run 200 simulations and count the portion of correctly recovered informative features (i.e. the support of $\boldsymbol{\beta}^*$). For LASSO, the regularization parameter was set to its optimal value $\alpha_N = \sqrt{\frac{2\sigma^2 \log(D-k)\log(k)}{N}}$ (Wainwright, 2009). For STG and HC, we set $\lambda_N = C\alpha_N$, such that C is a constant selected using a cross validated grid search in the range $[0.1, 10]$. To evaluate the effect of non-convex regularization and noise injection, we compare the STG to a deterministic non-convex (DNC) counterpart of our method (see definition below) and LASSO, which is convex. To gain insights on (ii), we also compare the HC.

We define the deterministic non-convex (DNC) objective as

$$\min_{\boldsymbol{\theta},\boldsymbol{\mu}} \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\theta}^T\boldsymbol{x}_n \odot \tilde{\boldsymbol{z}} - y_n)^2 + \lambda \sum_{d=1}^{D} \Phi\left(\frac{\mu_d}{0.5}\right), \tag{9}$$

where $\boldsymbol{\Phi}$ is the standard Gaussian CDF. Combined with $\tilde{z}_d$, this non-convex regularized objective is deterministic and

[3] http://featureselection.asu.edu/datasets.php

Table 1. Description of the real-world data used for empirical evaluation.

	BASEHOCK	RELATHE	RCV1	PCMAC	ISOLET	GISETTE	COIL20	MNIST	PBMC
Features (D)	7862	4322	47236	3289	617	5000	1024	784	17126
Train size	1594	1141	2320	1554	1248	5600	1152	60000	2074
Test size	398	285	20882	388	312	1400	288	10000	18666
Classes	2	2	2	2	26	2	20	10	2
Data type	Text	Text	Text	Text	Audio	Image	Image	Image	scRNA-seq

Table 2. Performance comparison of survival analysis on METABRIC. We repeat the experiment 5 times with different training/testing splits and report the mean and standard deviation on the testing set.

	DEEPSURV	RSF	COX-LASSO	COX-HC	COX-STG
C-INDEX	0.612 (0.009)	0.626 (0.006)	0.580 (0.003)	0.615 (0.007)	**0.633** (0.005)
# FEATURES	221 (ALL)	221 (ALL)	44 (0)	14 (1.72)	2 (0)

differentiable, and its solution can be searched via gradient descent.

As demonstrated in Fig. 4, the non-convex formulation requires less samples for perfect recovery of informative features than the LASSO. The injected noise based on the HC and STG provides a further improvement. Finally, we observe that the STG is more stable and has a lower variance than the HC, as shown by the shaded colors.

Application of the deterministic formulation is associated with a phenomenon that causes the gradient of an input feature to vanish and *never* acquire a nonzero value if it is zeroed out in an early training phase. In contrast, when we apply STG, a feature that at a certain step has a zero value is not permanently locked because the gate associated with it may change its value from zero to one at a later phase during training. This is due to the injected noise that allows our proposed method to *reevaluate* the gradient of each gate. In Fig. 4, we demonstrate this "second chance" effect using $N = 60$ samples and presenting the gate's values (throughout training) for an active feature.

The advantage of the Gaussian-based STG distribution over the HC distribution stems from the heavier tail of HC, whose form is a logistic distribution. We demonstrate that the heavy-tail distribution is not suitable for feature selection due to its high variance. An ideal feature selection algorithm is expected to identify a consistent set of features across different runs (feature stability), but HC selects many different features in each run resulting in high variance or lack of stability of the selected features.

To further examine the effect of heavy tail distributions, we train two identical neural networks on MNIST but use two different distributions for the gates: Gaussian-STG and HC. Both regularization parameters are tuned to retain 6 features. In Fig. 5, we show that the selected features from the Gaussian-STG are much more consistent across 20 runs than HC. Furthermore, the variance in the number of selected features is 3.8 for HC and 1 for STG. The average accuracies of STG and HC on the test are comparable: 92.4% and 91.7%, respectively.

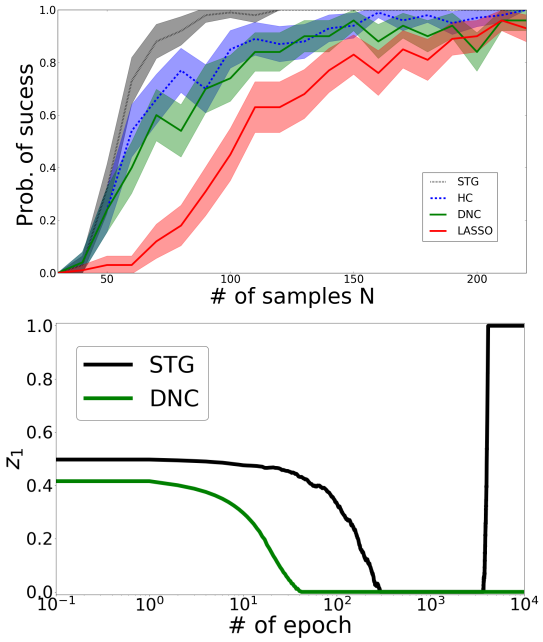


Figure 4. Feature selection in linear regression (see Section 8). The goal is to identify the subset of informative features. Top: Probability of recovering the informative features as a function of the number of samples. Comparison between STG, HC, LASSO and DNC. Bottom: The value of a gate z_1 throughout training. In STG, injected noise may lead to a "second chance" effect, which in this example occurs after 4000 epochs (black line). In the deterministic DNC setting (green line), a feature's elimination causes its gradient to vanish for the rest of the training.

9. Feature Selection with Correlations

Lastly, we evaluate our proposed method using data with correlated features. In real-world, high-dimensional datasets, many features are correlated. Such correlations introduce a challenge for feature selection. For instance, in the most extreme case if there are copies of the same feature, then it is not clear which to select. As another example consider if a large subset of features are a function of a small subset of features that we wish to identify. That large subset of

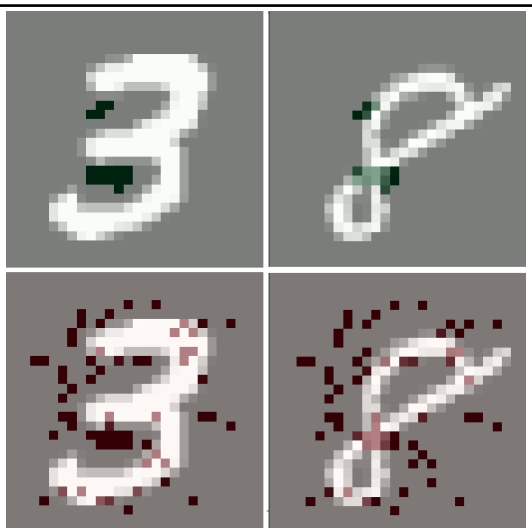


Figure 5. Comparing stability of feature selection by STG and HC. We train our method to classify $3s$ and $8s$ (from MNIST) with a regularization parameter tuned to retain ~ 6 features. We repeat the experiment using 20 random initializations. Dark pixels represent the union of selected features based on the STG (top) and HC (bottom) overlaid on top of two randomly sampled examples from each class. This demonstrates that an HC-based feature selection does not provide a stable selection of features across different runs.

seemingly useful features can confound a feature selection method. Below, we consider a number of examples in various correlated feature settings and demonstrate the strong performance of STG.

We first evaluate the proposed method in a linear setting. To introduce correlated features, we extend the linear regression experiment described in Section 8 using a correlated design matrix with a covariance matrix whose values are defined by $\Sigma_{i,j} = 0.3^{|i-j|}$. We run 100 simulations and present the probability of recovering the correct support of β^*. Fig. 6 shows that even if the features are correlated, STG successfully recovers the support with fewer samples than HC, DNC, and LASSO.

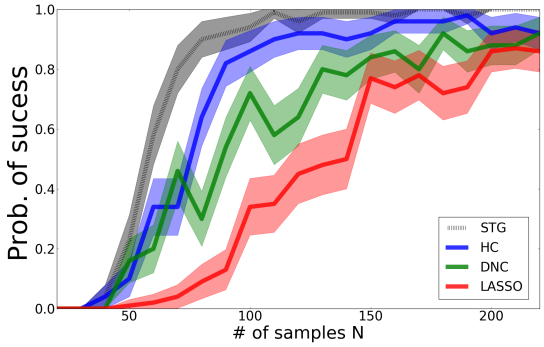


Figure 6. Feature selection in linear regression using a correlated design matrix. Probability of recovering the informative features as a function of the number of samples. Comparison between STG, HC, LASSO and DNC.

Next we evaluate our method in a non-linear setting using a variant of the MADELON dataset, which includes correlated features. Following (Guyon et al., 2005), the first 5 informative features of MADELON are used to create 15 additional coordinates based on a random linear combination of the first 5. A Gaussian noise $\mathcal{N}(0,1)$ is injected to each feature. Next, additional 480 nuisance coordinates drawn from $\mathcal{N}(0,1)$ are added. Finally, 1% of the labels are flipped. [4] We use $1,500$ points from this dataset and evaluate the ability of STG to detect the informative features.

Fig. 9 shows the precision of feature selection (black line) and the number of selected features (red line) as a function of the regularization parameter λ in the range $[0.01, 10]$. We observe that there is a wide range of λ values in which our method selects only relevant features (i.e. the precision is 1). Furthermore, there is a wide range of λ values in which 5 features are selected consistently.

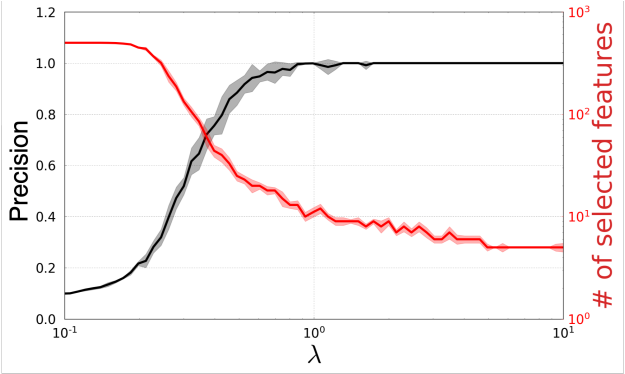


Figure 7. An empirical evaluation of the effect the regularization parameter λ has on the precision of feature selection (black line) and the number of selected features (red line). The precision and the number of selected features are presented on the left and right side of the y-axis, respectively. The means are displayed as solid lines while the standard deviations are marked as shaded regions around the means.

10. Conclusion

In this paper, we propose a novel embedded feature selection method based on stochastic gates. It has an advantage over previous ℓ_1 regularization based methods in its ability to achieve a high level of sparsity in nonlinear models such as neural networks, without hurting performance.

We justify our probabilistic feature selection framework from an information theoretic perspective. In experiments, we demonstrate that our method consistently outperforms existing embedded feature selection methods in both synthetic datasets and real datasets.

[4] generated using `dataset.make_classification` from `scikit-learn` (`http://scikit-learn.org/`)

Acknowledgements

The authors thank Nicolas Casey and the anonymous reviewers for their helpful feedback. This work was supported by the National Institutes of Health [R01GM131642, R01HG008383, P50CA121974 and R61DA047037], National Science Foundation DMS 1723128, and the Funai Overseas Scholarship to YY.

References

Allen, G. I. Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2):284–299, 2013.

Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of machine learning research*, 13(Jan):27–66, 2012.

Chandrashekar, G. and Sahin, F. A survey on feature selection methods. *Computers & Electrical Engineering*, 40 (1):16–28, 2014.

Chang, C.-H., Rampasek, L., and Goldenberg, A. Dropout feature ranking for deep learning models. 12 2017.

Chang, Y.-W. and Lin, C.-J. Feature ranking using linear svm. In *Causation and Prediction Challenge*, pp. 53–64, 2008.

Chen, J., Stern, M., Wainwright, M. J., and Jordan, M. I. Kernel feature selection via conditional covariance minimization. In *Advances in Neural Information Processing Systems*, pp. 6946–6955, 2017.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, 2016.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006. ISBN 0471241954.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346, 2012.

Díaz-Uriarte, R. and De Andres, S. A. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Feng, J. and Simon, N. Sparse-Input Neural Networks for High-dimensional Nonparametric Regression and Classification. *ArXiv e-prints*, November 2017.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pp. 441–452, 2018.

Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics*, pp. 1–67, 1991.

Gregorová, M., Ramapuram, J., Kalousis, A., and Marchand-Maillet, S. Large-scale nonlinear variable selection via kernel random features. *arXiv preprint arXiv:1804.07169*, 2018.

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. Result analysis of the nips 2003 feature selection challenge. In *Advances in neural information processing systems*, pp. 545–552, 2005.

Hans, C. Bayesian lasso regression. *Biometrika*, 96(4): 835–845, 2009.

Huang, J., Xie, H., et al. Asymptotic oracle properties of scad-penalized least squares estimators. In *Asymptotics: Particles, processes and inverse problems*, pp. 149–166. Institute of Mathematical Statistics, 2007.

Ishwaran, H., Kogalur, U., Blackstone, E., and Lauer, M. Random survival forests. *Annals of Applied Statistics*, 2 (3):841–860, 9 2008. ISSN 1932-6157. doi: 10.1214/08-AOAS169.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. 2017. URL https://arxiv.org/abs/1611.01144.

Kabir, M. M., Islam, M. M., and Murase, K. A new wrapper feature selection approach using neural network. *Neurocomputing*, 73(16-18):3273–3283, 2010.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18, 2018.

Kohavi, R. and John, G. H. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997a.

Kohavi, R. and John, G. H. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997b.

Laporte, L., Flamary, R., Canu, S., Déjean, S., and Mothe, J. Nonconvex regularizations for feature selection in ranking with sparse svm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1118–1130, 2013.

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

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Li, F., Yang, Y., and Xing, E. P. From lasso regression to feature vector machine. In *Advances in Neural Information Processing Systems*, pp. 779–786, 2006.

Li, W., Feng, J., and Jiang, T. Isolasso: a lasso regression approach to rna-seq based transcriptome assembly. In *International Conference on Research in Computational Molecular Biology*, pp. 168–188. Springer, 2011.

Li, Y., Chen, C.-Y., and Wasserman, W. W. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5): 322–336, 2016.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l0 regularization. *CoRR*, abs/1712.01312, 2017.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016. URL http://arxiv.org/abs/1611.00712.

Miller, A., Foti, N., D'Amour, A., and Adams, R. P. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*, pp. 3708–3718, 2017.

Min, F., Hu, Q., and Zhu, W. Feature selection with test cost constraint. *International Journal of Approximate Reasoning*, 55(1):167–179, 2014.

Mnih, A. and Rezende, D. J. Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*, 2016.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Peng, H., Long, F., and Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

Qian, J., Du, W., Tanigawa, Y., Aguirre, M., Tibshirani, R., Rivas, M. A., and Hastie, T. A fast and flexible algorithm for solving the lasso in large-scale and ultrahigh-dimensional problems. *BioRxiv*, pp. 630079, 2019.

Rastogi, R. and Shim, K. Public: A decision tree classifier that integrates building and pruning. *Data Mining and Knowledge Discovery*, 4(4):315–344, 2000.

Reunanen, J. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3(Mar):1371–1382, 2003.

Ribeiro, M. T., Singh, S., and Guestrin, C. "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*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL http://doi.acm.org/10.1145/2939672.2939778.

Roy, D., Murty, K. S. R., and Mohan, C. K. Feature selection using deep neural networks. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–6. IEEE, 2015.

Scardapane, S., Comminiello, D., Hussain, A., and Uncini, A. Group sparse regularization for deep neural networks. *Neurocomput.*, 241(C):81–89, June 2017. ISSN 0925-2312. doi: 10.1016/j.neucom.2017.02.029. URL https://doi.org/10.1016/j.neucom.2017.02.029.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bedo, J. Supervised feature selection via dependence estimation. In *Proceedings of the 24th international conference on Machine learning*, pp. 823–830. ACM, 2007.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May):1393–1434, 2012.

Stein, G., Chen, B., Wu, A. S., and Hua, K. A. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pp. 136–141. ACM, 2005.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Verikas, A. and Bacauskiene, M. Feature selection with neural networks. *Pattern Recognition Letters*, 23(11):1323–1335, 2002.

Wainwright, M. J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, May 2009. ISSN 0018-9448. doi: 10.1109/tit.2009.2016018. URL http://dx.doi.org/10.1109/tit.2009.2016018.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., and Sugiyama, M. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8:14049, 2017.

Zhu, P., Zhu, W., Wang, W., Zuo, W., and Hu, Q. Non-convex regularized self-representation for unsupervised feature selection. *Image and Vision Computing*, 60:22–29, 2017.

Zhu, Z., Ong, Y.-S., and Dash, M. Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):70–76, 2007.