Deep supervised feature selection using Stochastic Gates

Yutaro Yamada 1* Ofir Lindenbaum 2* Sahand Negahban 1
Yuval Kluger 2,3,4†

1Statistics Department; 2Applied Mathematics Program;
3Computational Biology and Bioinformatics;
4Department of Pathology;
Yale University, New Haven, CT, USA

†Corresponding author. E-mail: yuval.kluger@yale.edu
Address: 333 Cedar St, New Haven, CT 06510, USA
* These authors contributed equally.

January 31, 2019

Abstract

Feature selection problems have been extensively studied for linear estimation, for instance, 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 high-dimensional non-linear function estimation problems. The new procedure is based on minimizing the \( \ell_0 \) norm of the vector of indicator variables that represent if a feature is selected or not. Our approach relies on the continuous relaxation of Bernoulli distributions, which allows our model to learn the parameters of the approximate Bernoulli distributions via gradient descent. This general framework simultaneously minimizes a loss function while selecting relevant features. Furthermore, we provide an information-theoretic justification of incorporating Bernoulli distribution into our approach and demonstrate the potential of the approach on synthetic and real-life applications.

1 Introduction

Technological advances are leading to the generation of large complex data sets both in sample size and dimensionality. The collected data sets encapsulate both opportunities and challenges. For instance, in biology, we have access to tremendous amounts of biological markers and wish to model their interactions for prediction purposes. Unfortunately, that requires far more data than is generally available from clinical trials. A method to mitigate this challenge is to identify the key set of
features that influence prediction. Finding a subset of meaningful features might not only help the analytic task but also provide new scientific findings and improve the interpretability of models Ribeiro et al. (2016). Furthermore, reducing the number of features has computational advantages and has been shown to improve model generalization on unseen data Chandrashekar & Sahin (2014). In high-dimensional settings, there has been numerous works on non-parametric feature selection Friedman (1991); Tibshirani (1996); Lafferty & Wasserman (2008); Ravikumar et al. (2009); Meinshausen & Yu (2009); Raskutti et al. (2012).

Feature selection methods may be classified into three major categories: filter methods, wrapper methods, and embedded methods. Filter methods attempt to remove irrelevant features prior to classification. Typically, a relevance score is created for each feature based on some statistical measure. Filter methods have been demonstrated to be useful for various applications in Koller & Sahami (1996); Bekkerman et al. (2003); Davidson & Jalan (2010). More recent filter methods, such as Song et al. (2007, 2012); Chen et al. (2017) use kernels to represent nonlinear interactions between features. Wrapper methods use the classifiers outcome to determine the relevance of each feature. Among some of the wrapper approaches are Tree-based feature selection methods such as Kohavi & John (1997); Stein et al. (2005), sequential wrapper methods such as Zhu et al. (2007); Reunanen (2003) and an iterative kernel based wrapper method Allen (2013). A few neural network based wrapper methods include Verikas & Bacauskiene (2002); Kabir et al. (2010); Roy et al. (2015). The main disadvantage of all mentioned wrapper methods is that they require recomputing the classifier for each subset of features.

Embedded methods aim at relieving the computational burden. This is done by simultaneously learning the model and the subset of most relevant features. The Mutual Information (MI) based embedded methods include Battiti (1994); Peng et al. (2005); Estévez et al. (2009). Perhaps the most common embedded method is the Least Absolute Shrinkage and Selection Operator (LASSO) Tibshirani (1996). LASSO minimizes an objective function while enforcing an $\ell_1$ constraint on the weights of the features. Although LASSO has been extended in various works Hans (2009); Li et al. (2011, 2006), it remains a linear method, which limits the classification and regression capabilities. LASSO is extended to nonlinear functions in Yamada et al. (2014) by applying feature wise kernels. A few attempts have been made to generalize the objective used in LASSO to a neural network. In Li et al. (2016) a combination of $\ell_1$ and $\ell_2$ applied to the weights from the first layer are added as a regularization term to the training loss. This idea is extended in Scardapane et al. (2017) by imposing the regularization simultaneously on groups of weights outgoing from the same neuron. In practice these methods require an additional thresholding of the weights and their performance deteriorates for a large number of layers.

To overcome these limitations, we develop a practical fully embedded feature selection method for nonlinear functions. The proposed learning procedure aims at simultaneously minimizing the $\ell_0$ norm of a randomly selected subset of features along with a general loss function. Inspired by the recent efforts of developing a continuous differentiable approximation to discrete distributions Jang et al. (2017), Maddison et al. (2016), we present a simple relaxation of Bernoulli distributions, which enables us to introduce stochastic gates on the inputs, whose probability being active will be jointly learned with the model parameters via gradient descent.
Our formulation naturally extends from linear models to neural networks by introducing indicator variables in the input layer of a network. We also provide an information-theoretic interpretation on the Bernoulli relaxation of the best subset selection, which justifies the introduction of randomness for feature selection in our risk minimization. We apply our feature selection method to various artificial and real datasets to demonstrate its effectiveness. Finally, we demonstrate that for feature selection tasks, our method has an advantage over the previously proposed relaxation.

2 Background

Feature selection can be considered as finding the subset of features that lead to the largest possible generalization or equivalently to minimal risk. Every subset of features is modeled by a vector of indicator variables $z \in \{0, 1\}^D$, where the point-wise product with an input $x \in \mathbb{R}^D$ provides a subset of features.

Given a parameterized family of regression functions $f_\theta : \mathbb{R}^D \to \mathbb{R}$, we try to find a vector of indicator variables $z^* \in \{0, 1\}^D$ and a vector of parameters $\theta^* \in \Theta$ that minimize the expected risk:

$$R(\theta, z) = \mathbb{E}_{X,Y} L(f_\theta(z \odot x), y),$$

where $\odot$ denotes the point-wise product, $L$ is a loss function and $(X, Y)$ come from some data distribution $P$. The empirical risk minimizer is calculated using the observations $\{x_n, y_n\}_{n=1}^N$, where $x_n$ is a feature vector and $y_n$ is a response variable or class label. In some cases, we will also have the additional constraint $\Omega(z) \leq k$, where $\Omega : [0, 1]^D \to \mathbb{R}_+$ measures the sparsity of a given indicator variable $z$.

2.1 Feature Selection as an Optimization Problem

Most penalized linear models are expressed in terms of the following minimization:

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^N L(\theta^T x_n, y_n) + \lambda \Omega(\theta),$$

where $\Omega(\Theta) : \mathbb{R}^D \to \mathbb{R}_+$ is a penalizing term. The hyper-parameter $\lambda$ balances the empirical error with the penalizing term. Examples of losses are the $\ell_1$ hinge loss, the $\ell_2$ loss, and the logistic loss. Common penalizing terms include the $\ell_1$ norm and the $\ell_2$ norm.

The minimization in Eq. (2) inspired the least absolute shrinkage and selection operator (LASSO), which enables a computationally efficient feature selection procedure. The objective function of LASSO can be written as follows:

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^N ||\theta^T x_n - y_n||_2^2 + \lambda ||\theta||_1.$$  

The above minimization problem has been applicable in numerous scientific fields and in other domains.
3 Proposed Method

The rise of $\ell_1$ norm constraints in sparse estimation was partially due to its computational efficiency - the $\ell_1$ norm is the closest convex function to the $\ell_0$ norm Hastie et al. (2015). By adding the $\ell_1$ norm as a regularization term, LASSO’s minimization procedure is prone to select more relevant features. We take a probabilistic approach to approximate the $\ell_0$ norm, which can extend to non-linear models while remaining computationally efficient.

To view the above optimization from a probabilistic perspective, one can introduce a Bernoulli distribution $B(z|\pi)$, where $\pi_d$ is the parameter of indicator variable $z_d, d = 1, ..., D$.

Then, Eq. (1) becomes

$$R(\theta, \pi) = E_{X,Y} E_{B(z|\pi)} [L(f_\theta(z \odot x), y) + \lambda ||z||_0]$$

$$= E_{X,Y} E_{B(z|\pi)} [L(f_\theta(z \odot x), y)] + \lambda \sum_{d=1}^{D} \pi_d,$$

where $B(z|\pi)$ is the product of $D$ independent Bernoulli distributions, and $x, y \sim P(X, Y)$ are the input and response variables from the data distribution.

Our goal is to find $\theta^*$ and $\pi^*$ that minimize $R(\theta, \pi)$ via tractable methods such as gradient descent.

3.1 Issues in Gradient Estimation

The first term of the empirical risk $\hat{R}(\theta, \pi)$ is expressed as

$$\sum_{z:(0,1)^D} \sum_{n=1}^{N} [L(f_\theta(z \odot x_n), y_n) \prod_{d=1}^{D} \pi_d (1 - \pi_d)^{1 - z_d}].$$

In practice, we have to replace the outer sum, which enumerates $2^D$ possibilities of the indicator variables, with Monte Carlo samples from the product of Bernoulli distributions $B(z|\pi)$. However, a Monte Carlo estimate of $\frac{\partial}{\partial \pi_d} \hat{R}(\theta, \pi)$ suffers from high variance.

The exact gradient of the empirical risk with respect to $\pi_d$ is

$$\sum_{z:(0,1)^D, z_d=1} [L(z)p_{z_i \neq d}] - \sum_{z:(0,1)^D, z_d=0} [L(z)p_{z_i \neq d}],$$

where $p(z_{i \neq d}) = \prod_{i \neq d}^{D} \pi_i^{z_i}(1 - \pi_i)^{1 - z_i}$ and we absorb the model $f_\theta(\cdot)$ and the data into $L(\cdot)$. We can see that even the sign of the gradient estimate becomes inaccurate if we can only access a small number of Monte Carlo samples, which creates difficulties in optimization.

3.2 Continuous Relaxation

As detailed in Subsection 3.1, the optimization of a loss which consists of a discrete random variable suffers from high variance. To address this limitation, we introduce
a continuous relaxation of Bernoulli distributions, which allows the model to utilize
the gradient information of the loss evaluated at approximated Bernoulli samples,
and thus reduce the variance of the gradient estimation in practice. For each fea-
ture $d = 1, \ldots, D$, we first sample a random variable $\epsilon$ from the standard Gaussian
distribution $\mathcal{N}(0, 1)$, then we shift and scale it using parameters $\mu_d$ and $\sigma_d$. We pass
the rescaled variable to the hard-sigmoid function $h(x) = \max(0, \min(1, ax + 0.5))$, where $a$ is the slope parameter. The procedure is summarized as follows:

$$x_d = \mu_d + \sigma_d \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1),$$

$$\tilde{z}_d = \max(0, \min(1, ax_d + 0.5)).$$

The resulting random variable $\tilde{z}_d$ acts as a continuous approximation to a Bernoulli
random variable, whose gradient with respect to its parameters can be computed
via backpropagation; $\tilde{z}_d$ is a deterministic function of $\mu_d$, and the randomness only
comes from $\epsilon$.

The new objective is now

$$R(\theta, \mu) = \mathbb{E}_{X, Y} \mathbb{E}_{\tilde{z}} [L(f_{\theta}(\tilde{z} \odot x), y) + \lambda ||\tilde{z}||_0].$$

The resulting gradient estimator becomes

$$\frac{\partial}{\partial \mu_d} \hat{R}(\theta, \mu) = \frac{1}{K} \sum_{k=1}^{K} \left[ L'(\tilde{z}^k) \frac{\partial \tilde{z}^k_d}{\partial \mu_d} \right],$$

where $K$ is the number of Monte Carlo samples. We note that if we replace
$\frac{\partial \tilde{z}^k_d}{\partial \mu_d}$ with 1, the above gradient estimator is reduced to the Straight-Through esti-
mator Bengio et al. (2013). In feature selection problems, $\frac{\partial \tilde{z}^k_d}{\partial \mu_d}$ carries the critical
information regarding feature’s relevance.

### 3.3 Implementation of the Regularization Term

Under the continuous relaxation we have employed, the expected regularization term
in the objective $R(\theta, \mu)$ is calculated by the sum of the probability of the gate $z_d$
being active for $d = 1, \ldots, D$. Using the cumulative distribution function of
the Gaussian distribution, it is expressed as $\sum_{d=1}^{D} \left[ 1 - \Pr(x_d < \frac{1}{2a}) \right]$, which can be
implemented using the Gauss error function available in a standard machine learning
library such as TensorFlow. However, the direct application of the Gaussian CDF
does not optimize the loss since the gradient vanishes too quickly as $\mu_d$ moves away
from the origin. To address this issue in our implementation, we employ a function
such as $\frac{1}{1+|x|}$ that decays slower than the Gauss error function.

### 4 Connection to Mutual Information

In this section, we demonstrate that replacing the original subset selection prob-
lem (Eq. 2) with the Bernoulli probabilistic setting can be justified from a mutual
information perspective in the feature selection setting. This is motivated by, but
different than the work done by Chen et al. (2018). Recall that the mutual infor-
mation between two random variables can be defined as
\[ I(X; Y) = \mathbb{E}[\log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}] = H(Y) - H(Y|X), \]

where \( H(Y), H(Y|X) \) are the entropy of \( p_Y(y) \) and the conditional entropy of \( p_{Y|X}(y|x) \), respectively \cite{Cover2006}. Next, we present our two assumptions for this section:

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

- **Assumption 2**: \( I(X_{S^*}; Y|X_{S^*}) = 0 \).

The first assumption means that if we do not include an element from \( S^* \), then we can improve our prediction accuracy by adding it. The second assumption means that we only need the variables in \( S^* \) to predict \( Y \). Any additional variables are superfluous. The assumptions are quite benign. For instance they are satisfied if \( X \) is drawn Gaussian with a non-degenerate covariance matrix and \( Y = f(X_{S^*}) + w \), where \( w \) is noise independent of \( X \) and \( f \) is not degenerate. With this in place we can present our results.

**Proposition 1.** Suppose that the above assumptions hold for the model. Then, solving the optimization

\[
\max_{S} I(X \odot S; Y) \quad s.t. \quad \sum_i S_i \leq k \quad \text{and} \quad S_i \in \{0, 1\} \tag{4}
\]

is equivalent to solving the optimization

\[
\max_{0 \leq \pi \leq 1} I(X \odot S; Y) \quad s.t. \quad \sum_i \mathbb{E}[S_i] \leq k, \tag{5}
\]

where the coordinates \( S_i \) are drawn independently at random according to a Bernoulli distribution with parameter \( \pi_i \).

Due to length constraints, we leave the proof of this proposition to the appendix.

## 5 Related Work

The two most related works to this study are \cite{Louizos2017} and \cite{Chen2018}. In \cite{Louizos2017}, they introduce the Hard-Concrete distribution as a continuous surrogate for Bernoulli distributions in the context of model compression. The authors demonstrate how their method leads to fast convergence and improved generalization in deep neural networks due to the sparsification effect. In this study, we focus on a different goal, and demonstrate that a simple relaxation of Bernoulli distributions is sufficient and works better than the Hard-Concrete distribution for feature selection tasks. Unlike the full sparsification framework, our method enables us to increase the feature size up to a number of thousands of features (as shown in...
the Section 6.6); this high dimensional regime is common in a field such as bioinformatics. In Chen et al. (2018), the Gumbel-softmax trick is used to develop a framework for interpreting pre-trained models. Their method is focused on finding a subset of features given a particular instance, and therefore is not appropriate for general feature selection. In the case of gene data, for instance, we are often interested in finding consistent important features.

Some authors tackle embedded feature selection problems by extending LASSO and group LASSO to neural network models. Although Li et al. (2016); Scardapane et al. (2017) and Feng & Simon (2017) have a similar goal as ours, their empirical result does not achieve sufficient sparsity as practitioners would like. Our approach, which utilizes stochastic gates along with the \( \ell_0 \) norm instead of relying on regularizing \( \ell_1 \) norms, has an advantage in terms of achieving high sparsity level while maintaining good performance.

6 Experiments

In this section, we perform a variety of experiments to evaluate the potential of using the proposed Stochastic Gates (STG) for feature selection. In the first two experiments, we generate artificial samples by randomly sampling points based on some parametric distribution; then we assign a label based on a non-linear function. Each data set is concatenated with nuisance noisy coordinates; these coordinates do not hold information regarding the class identity. We compute both classification accuracy and feature weights depicted by our proposed approach. To evaluate the strength of STG, we compare it to feature ranking using support vectors classification (SVC) Chang & Lin (2008), LASSO Tibshirani (1996), and two tree-based methods Rastogi & Shim (2000) and Strobl et al. (2008) which we denote Tree and Random Forests (RF) respectively. We also compare to network based feature selection methods, deep feature selection (DFS), group sparse regularized NN (SG-L1-NN) and to a neural network without any feature selection layer (DNN). In each experiment, all neural network based methods use the same architecture. Each method extracts a weight for the relevance of each feature, we denote this weight by \( W_i \geq 0, i = 1, ..., D \). The wrapper methods, SVC, Tree, and RF are retrained based on the features with highest extracted weights. In the following experiments, we compare classification accuracy as well as the Informative Features Weight Ratio (IFWR). IFWR is defined as the sum of weights \( W_i \) over the informative features divided by the sum over all weights.

The architecture of the neural networks is similar for all experiments. We use between 1 to 5 layers with widely used activation functions such as Linear, Relu or Tanh. For the artificial datasets, the exact architecture is optimized such that the network reaches near-optimal performance when no nuisance variables are added. For the classification experiments, the final layer performs logistic regression. Whereas for the Cox hazard model (Section 6.7) we use the scaled exponential linear units (Selu) and partial likelihood as the loss.

Setting the hyperparameter \( \lambda \) effects the portion of selected features. In the handwriten digits and biological experiments (Sections 6.5 and 6.6) we evaluate the effect of \( \lambda \) on the amount of features selected by the network.
6.1 Two Moons classification with nuisance features

In the first experiment, we construct a dataset based on "two moons" shape classes, concatenated with noisy features. The first two coordinates \(x_1, x_2\) are generated by adding a Gaussian noise with zero mean and the variance of \(\sigma_r^2 = 0.1\) onto two nested half circles, as presented in Fig. 1(a). Nuisance features \(x_i, i = 3, ..., D\), are drawn from a Gaussian distribution with zero mean and variance of \(\sigma_n^2 = 1\). The classification accuracy and the portion of weights assigned to the informative feature (IFWR) are presented in Fig. 1(b) and 1(c). Based on the classification accuracies and assigned weights, it is evident that for a small number of nuisance dimensions all methods correctly identify the most relevant features. The proposed method (STG) and Random Forest (RF) are the only methods that achieve near perfect classification accuracy for a wide range of nuisance dimensions. The other NN based methods (DFS and SG-L1-NN) seem to converge to sub-optimal solutions. STG and LASSO are the only methods which naturally sparsity the feature space. The parameter was set to \(\lambda = 0.05\); this value appears to perform well for a wide range of nuisance dimensions. Moreover, the performance seemed stable for a wide range of \(\lambda\)'s.

![Figure 1](image1.png)

**Figure 1:** (a) Realizations from the "Two moons" shaped binary classification class. \(X_1\) and \(X_2\) are the relevant features, \(X_i, i = 3, ..., D\) are noisy features drawn from a Gaussian with zero mean and variance of 1. (b) Classification accuracy (mean and standard deviation) vs. the number of irrelevant noisy dimension. (c) The portion of relevant weights attributed to the informative feature, this metric is denoted as IFWR.
6.2 Noisy binary XOR classification

The Junta problem was originally suggested by Gödel, the goal is to learn a Boolean function which depends on $k$ variables out of a set of size $D$. Mossel et al. (2003). Here, we consider this problem using a binary XOR for classification task. The first two coordinates $x_1, x_2$ are drawn from a binary "fair" Bernoulli distribution. The response variable is set as an XOR of the first coordinates, such that $y = x_1 \oplus x_2$. The coordinates $x_i, i = 3, ..., D$ are nuisance features also drawn from a binary "fair" Bernoulli distribution. The number of points is $N = 1500$, and 30% are used as a training set while the rest are used for test. The experiment is repeated 50 times for different values of $D$, and the average test classification accuracy and standard deviation are presented in Fig. 2(a). In Fig. 2(b) the average IFWR metric is presented. Although every single feature is statistically independent of $y$, the proposed method manages to reach perfect accuracy for large $D$ and reasonably small $N$.

Figure 2: (a) Classification accuracy (mean and standard deviation) vs. the number of irrelevant noisy dimension ($D$) for the XOR problem. (b) The portion of weights attributed to the informative feature (mean and standard deviation). This metric is denoted as IFWR.

6.3 Regression using synthetic and real datasets

In this section, we evaluate our method for regression tasks against two other embedded feature selection methods: LASSO and Sparse Random Fourier Features Gregorová et al. (2018). Following the same format as Gregorová et al. (2018), the following functions are used to generate synthetic data: (SE1: 100/5) $y = \sin (x_1 + x_3)^2 \sin (x_7 x_8 x_9) + \mathcal{N}(0, 0.1)$. (SE2: 18/5) $y = \log((\sum_{s=1}^{5} x_s)^2) + \mathcal{N}(0, 0.1)$. (SE3: 1000/10) $y = 10(z_1^2 + z_3^3) e^{-2(z_1^4 + z_2^4)} + \mathcal{N}(0, 0.01)$. The numbers next to the experiment code indicate total dimensions/irrelevant dimensions in the feature space. We also evaluate our method using a real dataset (RCP: 21/-), which measures computer systems activity, taken from the LIACC repository of computer systems activity, taken from the LIACC repository. For each dataset, we generate 30 different replications and randomly split the data into train, validation, and test set. For data preparation, we use the code publicly made available by Gregorová et al. (2018). The root mean squared error on the test set averaged

\footnote{http://www.dcc.fc.up.pt/~itorgo/Regression/DataSets.html}
over 30 random replicated datasets are reported in Table 1. Our method outperforms the other two methods for most cases. We note that (SE1) is generated using the sine function, which is in favor of SRFF, the random Fourier feature based method.

6.4 Comparison to Hard-Concrete distribution

To evaluate the strength of the proposed continuous relaxation of the Bernoulli distribution, described in Subsection 3.2, we compare it with the Hard-Concrete distribution Louizos et al. (2017), another continuous surrogate for Bernoulli distributions, which was originally developed for neural network model compression. The details of the Hard-Concrete distribution is deferred to the appendix.

The main difference is that their distribution is based on the logistic distribution, which has a heavier tail than the Gaussian distribution we have employed. The heavy-tailness encourages "exploration" in terms of feature selection during training, but results in the instability as shown in Fig. 3; Our method converges much faster and more reliably than the feature selection method using the Hard-Concrete distribution on a the two-moons dataset (Subsection 6.1).

![Figure 3: Comparison between STG and Hard-Concrete on the Two-Moon dataset. The shaded area represents the standard deviation, calculated by running the same experiment 10 times with different random initializations.](image)

6.5 Sparse Handwritten digits classification

In the following toy example, we attempt to distinguish between images of handwritten digits of 3’s and 8’s using samples from MNIST LeCun et al. (1998). The orientation and location of the digits is more or less the same throughout this dataset, therefore for these two classes (3’s and 8’s), we expect that some of the left side features (pixels) would be sufficient for the separation. The experiment is performed as followed. We omit 90% of the data as the test set, and train on the remaining 10%. We then apply STG and evaluate the classification accuracy and selected features. The experiment was repeated 10 times, the extracted features and accuracies were consistent over 20 trials. We noticed a relatively small number of selected features which are positioned southwest and close to the center of the images achieve very high classification accuracy. An example of 9 randomly selected samples overlaid with the weights of the selected features is presented in Fig. 4(a). In this experiment, we further evaluate the effect of λ on the sparsification and accuracy of the method.
We apply the approach to a randomly sampled training set of size $N = 1500$ and change $\lambda$ in the range of $[0.001, 0.01]$. In Fig. 4(b) we present the accuracy and sparsity level vs. the $\lambda$ parameter. This experiment demonstrates the improved performance of the proposed distribution compared to the Hard-Concrete (HC Louizos et al. (2017)). Not only that the overall accuracy is superior, but it seems that the transition as a function of $\lambda$ is smoother, this suggests that the method is less sensitive to the choice of $\lambda$.

![Figure 4: (a) 9 samples from MNIST (white) overlaid with the subset of 13 features (black) selected by STG. Based on these feature classification accuracy reaches 92.2%. For these 9 randomly selected samples all the 8’s have values within the support of the selected features, whereas for the 3’s there is no intersection. (b) Accuracy and sparsity level performance for $\lambda$ in the range of $[10^{-3}, 10^{-2}]$, a comparison between the performance using the Hard-Concrete (HC) distribution and the proposed distribution (STG).](image)

6.6 Purified populations of peripheral blood monocytes (PBMCs)

Single-cell RNA sequencing (scRNA-seq) is a novel technology that measures gene expression levels of hundreds of thousands of individual cells, simultaneously. This new tool is revolutionizing our understanding of cellular biology as it enables, among other things, the discovery of new cell types as well as the detection of subtle differences between similar but distinct cells. The authors in Zheng et al. (2017), have subjected more than 90,000 purified populations of peripheral blood monocytes (PBMCs) to scRNA-seq analysis. Such blood cells have been thoroughly characterized and studied. Many of these cells fall into well separated clusters (subpopulations). Here we focus on two subpopulations of T-cells, namely the Naive and regulatory T-cells. The Naive CD4+ T-cells are responsible for activating the immune system against antigens, and the regulatory T-cells prevent activation against self antigens (autoimmune diseases). In the following experiment, we use the proposed method to select a subset of genes for which the network discriminates between Naive and regulatory T-cells. We first filter out the genes that are lowly expressed in the cells, this leaves us with $D = 2538$ genes (features). The total number of cells in these two classes is $N = 20742$, of which we only use 10% of the data for training.
This again is a challenging regime for a classification task. We repeat the experiment for different values of $\lambda$ and report the number of selected features and classification accuracy on the test set. Here we compare our performance to Random Forests and LASSO. A scatter plot of the accuracy vs. number of selected features is presented in Fig. 5. For visibility convenience, we have added a least squares polynomial fit to all scatter plots in Fig. 5. We have also evaluated the performance of the Hard-Concrete applied to all layers (HC-Full), following the procedure in Louizos et al. (2017). It seems that using this type of regularization provides inferior capabilities in terms of feature selection. Moreover, when the method converges to a larger subset of features ($d > 50$), it does not generalize at all and the test accuracy is around 0.5.

Figure 5: Classification of T-cells sub-populations. The performance vs. number of features selected by each method. Comparison of the proposed method to Random Forests (RF), LASSO and to the Hard-Concrete applied to all of the weights in the network (HC-Full).

6.7 Cox Proportional Hazard Models for Survival Analysis

In survival analysis, we are interested in building a predictive model for the survival time $T_i$ of an individual $i$ based on the covariates $x_i \in \mathbb{R}^D$. Survival times are assumed to follow a distribution, which is characterized by the survival function $S(t) = P(T > t)$. A hazard function, which measures the instantaneous rate of death, is defined by

$$ h(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} = \frac{p(t)}{S(t)}. $$

We can relate the two functions in the following way: $S(t) = e^{-\int_0^t h(t) dt}$.

Proportional hazard models assume a multiplicative effect of the covariates $x$ on the hazard function such that

$$ h(t|x) = h_0(t)e^{\theta^T x}, $$
where \( h_0(t) \) is a baseline hazard function, which is often the exponential or Weibull distribution, and \( \theta \) is the parameter of interests.

One of the difficulties in estimating \( \theta \) in survival analysis is that a large portion of the available data is censored. However, in order to obtain estimates, Cox observed that it is sufficient to maximize the partial-likelihood, which is defined as follows:

\[
L(\theta) = \prod_{T_i \text{ uncensored}} e^{\theta^T x_i} \frac{1}{\sum_{T_j \geq T_i} e^{\theta^T x_j}}.
\]

In Katzman et al. (2018), the authors propose DeepSurv, which uses a deep neural network model to replace the linear relations between the covariate \( x \) and \( \theta \), demonstrating improvements of survival time prediction over existing models such as CPH and the random survival forest Ishwaran & Kogalur (2007), Ishwaran et al. (2008).

We apply our feature selection method in DeepSurv to see how our procedure improves the performance on the breast cancer dataset called METABRIC Curtis et al. (2012).

The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) uses gene and protein expression profiles to determine new breast cancer subgroups in order to help physicians provide better treatment recommendations. The METABRIC dataset consists of gene expression data and clinical features for 1,980 patients, and \( 57.72\% \) have an observed death due to breast cancer with a median survival time of 116 months.

We pick 16 genes used in the Oncotype DX test, a genomic test that analyzes the activity of genes that could affect breast cancer behavior. We join these genes with the patient’s clinical features (hormone treatment indicator, radiotherapy indicator, chemotherapy indicator, ER-positive indicator, age at diagnosis). To evaluate our feature selection method, we randomly sample genes to construct nuisance features. In total, we obtain 221 features, where 200 features are randomly sampled. We then hold out \( 20\% \) of the patients as the test set.

We evaluate the predictive ability of the learned models based on the concordance index (CI), which measures the agreement between the rankings of the predicted and observed survival times. The concordance index is a standard performance measure for model assessment in survival analysis.

We compared our method (Cox-STG) against three other methods: Cox model with \( \ell_1 \) regularization (Cox-LASSO; available as a python glmnet package), Random Survival Forest (RSF; available as an R package), and the original DeepSurv (DSurv).

The performance in terms of the c-index is reported in Table 1. We see that Cox-STG outperforms all the other methods. We note that the original Cox model using neural nets (DSurv) overfits to the training set, resulting in the poor generalization, whereas Cox-STG successfully shrinks the feature size and achieves good performance on the test set.

7 Conclusion

In this paper, we propose a novel embedded feature selection method based on stochastic gates. It has an advantage over previous \( \ell_1 \) regularization based methods
Table 1: Performance comparison of survival analysis on METABRIC. For DSurv and Cox-STG, we run the same experiment 5 times with random weight initializations and report the mean.

|            | COX-LASSO | RSF | DSURV | COX-STG |
|------------|-----------|-----|-------|---------|
| C-index    | 0.603     | 0.548| 0.627 | 0.650   |

in terms of achieving a high level of sparsity in non-linear models such as neural networks, without hurting the performance. We justify our probabilistic feature selection framework from the information theoretic perspective. In experiments, we demonstrate that our method consistently outperforms existing embedded feature selection methods in both synthetic datasets and real datasets.

8 Conclusion

In this paper, we propose a novel embedded feature selection method based on stochastic gates. It has an advantage over previous $\ell_1$ regularization based methods in terms of achieving a high level of sparsity in non-linear models such as neural networks, without hurting the performance. We justify our probabilistic feature selection framework from the information theoretic perspective. In experiments, we demonstrate that our method consistently outperforms existing embedded feature selection methods in both synthetic datasets and real datasets.

8.1 Acknowledgements

We thank Ronen Basri for helpful discussions. O.L. and Y.K. are supported by NIH grant 1R01HG008383-01A1.

8.2 Appendix

Proof of Proposition 1 We now give the proof of the proposition showing the equivalence between the stochastic optimization (Eq. (5)) and the deterministic one (Eq. (4)). Let $\tilde{S}$ be a subset such that $S^* \setminus \tilde{S} \neq \emptyset$. That is there exists some element in $S^*$ that is not in $\tilde{S}$. For any such set $\tilde{S}$ we have that $I(X_{\tilde{S}}; Y) < I(X; Y)$. Indeed, if we let $i \in S^* \cap \tilde{S}$ then we have

$$I(X_{\tilde{S}}; Y) \leq I(X \setminus \{i\}; Y)$$

$$= I(X; Y) - I(X_i; Y | X \setminus \{i\})$$

$$< I(X; Y),$$

where the final inequality follows by Assumption 1. Assumption 2 also yields that for any set $\tilde{S}$ such that $S^* \subset \tilde{S}$, we have $I(X_{\tilde{S}}; Y) = I(X; Y)$. Now, when we consider the Bernoulli optimization problem we have

$$\max_{\pi} I(X \otimes S; Y) \quad \text{s.t.} \quad \sum_{i} \pi_i \leq k \text{ and } 0 \leq \pi_i \leq 1.$$
The mutual information can be expanded as

$$I(X \odot S; Y) = \sum_s I(X \odot s; Y) p_\pi(S = s)$$

where we have used the fact that $S$ is independent of everything else. Recall that in optimization (Eq. (5)) the coordinates of $S$ are sampled at random. Therefore, the distribution that is being optimized over $p_\pi$ is a product distribution. Our goal is to understand the form of this distribution. To that end, we will consider a problem dropping the independence constraint. If we can show that the distribution found by solving this new optimization problem with less constraints is still a product distribution, then we obtain a solution to the original optimization (Eq. (5)).

Now, from above we know that the optimal value of the optimization is $I(X \odot \tilde{S}; Y)$ for any set $S^* \subset \tilde{S}$. Hence, any unconstrained distribution should place all of its mass on such subsets in order to maximize the mutual information. As a result $\sum_{l \in S^*} p(S_l = 1) = k$. However, there is an optimization constraint that $E[\sum_l S_l] \leq k$. Therefore, $E[S_l] = 0$ for any $l \notin S^*$. Hence, the optimal solution is to select the distribution so that all of the mass is placed on the subset $S^*$ and no mass elsewhere. As this is also a product distribution, this complete the proof of the claim.
The Hard-Concrete distribution

The authors in Louizos et al. (2017) introduce a modification of Binary Concrete, whose sampling procedure is as follows:

\[
\begin{align*}
  u &\sim U(0,1), \quad L = \log(u) - \log(1-u) \\
  s &= \frac{1}{1 + \exp\left(-\frac{\log(\alpha + L)}{\beta}\right)} \\
  \bar{s} &= s(\zeta - \tau) + \tau \\
  z &= \min(1, \max(0, \bar{s}))
\end{align*}
\]

where \((\tau, \zeta)\) is an interval, with \(\tau < 0\) and \(\zeta > 1\). This induces a new distribution, whose support is \([0,1]\) instead of \((0,1)\). With \(0 < \beta < 1\), the probability density concentrates its mass near the end points, since values larger than \(\frac{1-\tau}{\zeta-\tau}\) are rounded to one, whereas values smaller than \(\frac{-\tau}{\zeta-\tau}\) are rounded to zero.

The CDF of \(s\) is

\[
Q_s(s|\beta, \log \alpha) = \text{Sigmoid}(\log s - \log(1-s))\beta - \log \alpha
\]

and so the CDF of \(\bar{s}\) is

\[
Q_{\bar{s}}(\bar{s}|\phi) = \text{Sigmoid}(\log(\bar{s} - \tau) - \log(1 - \bar{s} - \tau))\beta - \log \alpha
\]

where \(\phi = (\beta, \log \alpha, \zeta, \tau)\)

Now, the probability of being the gate \(z_i\) active, which is \(1 - Q_{\bar{s}}(0|\phi)\), can be written as

\[
1 - Q_{\bar{s}}(0|\phi) = \text{Sigmoid}(\log \alpha - \beta \log \frac{-\tau}{\zeta})
\]

References

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

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

Bekkerman, Ron, El-Yaniv, Ran, Tishby, Naftali, and Winter, Yoad. Distributional word clusters vs. words for text categorization. Journal of Machine Learning Research, 3(Mar):1183–1208, 2003.

Bengio, Yoshua, Léonard, Nicholas, and Courville, Aaron C. Estimating or propagating gradients through stochastic neurons for conditional computation. CoRR, abs/1308.3432, 2013.

Chandrashekar, Girish and Sahin, Ferat. A survey on feature selection methods. Computers & Electrical Engineering, 40(1):16–28, 2014.
Chang, Yin-Wen and Lin, Chih-Jen. Feature ranking using linear svm. In *Causation and Prediction Challenge*, pp. 53–64, 2008.

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

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

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

Curtis, Christina, Shah, Sohrab P, Chin, Suet-Feung, Turashvili, Gulisa, Rueda, Oscar M, Dunning, Mark J, Speed, Doug, Lynch, Andy G, Samarajiwa, Shamith, Yuan, Yinyin, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346, 2012.

Davidson, Jennifer L and Jalan, Jaikishan. Feature selection for steganalysis using the mahalanobis distance. In *Media Forensics and Security II*, volume 7541, pp. 754104. International Society for Optics and Photonics, 2010.

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

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

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

Gregorová, Magda, Ramapuram, Jason, Kalousis, Alexandros, and Marchand-Maillet, Stéphane. Large-scale nonlinear variable selection via kernel random features. *arXiv preprint arXiv:1804.07169*, 2018.

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

Hastie, Trevor, Tibshirani, Robert, and Wainwright, Martin. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015. ISBN 1498712169, 9781498712163.

Ishwaran, Hemant and Kogalur, Udaya B. Random survival forests for r, 2007.

Ishwaran, Hemant, Kogalur, Udaya B., Blackstone, Eugene H., and Lauer, Michael S. Random survival forests. *Annals of Applied Statistics*, 2(3):841–860, 9 2008. ISSN 1932-6157. doi: 10.1214/08-AOAS169.

Jang, Eric, Gu, Shixiang, and Poole, Ben. Categorical reparameterization with gumbel-softmax. 2017. URL https://arxiv.org/abs/1611.01144.
Kabir, Md Monirul, Islam, Md Monirul, and Murase, Kazuyuki. A new wrapper feature selection approach using neural network. *Neurocomputing*, 73(16-18):3273–3283, 2010.

Katzman, Jared L., Shaham, Uri, Cloninger, Alexander, Bates, Jonathan, Jiang, Tingting, and Kluger, Yuval. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18, 2018.

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

Koller, Daphne and Sahami, Mehran. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.

Lafferty, John and Wasserman, Larry. Rodeo: Sparse, greedy nonparametric regression. *Ann. Statist.*, 36(1):28–63, 02 2008. doi: 10.1214/009053607000000811. URL https://doi.org/10.1214/009053607000000811.

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

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

Li, Wei, Feng, Jianxing, and Jiang, Tao. 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, Yifeng, Chen, Chih-Yu, and Wasserman, Wyeth W. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

Louizos, Christos, Welling, Max, and Kingma, Diederik P. Learning sparse neural networks through l0 regularization. *CoRR*, abs/1712.01312, 2017.

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

Meinshausen, Nicolai and Yu, Bin. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Statist.*, 37(1):246–270, 02 2009. doi: 10.1214/07-AOS582. URL https://doi.org/10.1214/07-AOS582.

Mossel, Elchanan, O’Donnell, Ryan, and Servedio, Rocco P. Learning juntas. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pp. 206–212. ACM, 2003.

Peng, Hanchuan, Long, Fuhui, and Ding, Chris. 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.
Raskutti, Garvesh, Wainwright, Martin J., and Yu, Bin. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *J. Mach. Learn. Res.*, 13:389–427, February 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2188385.2188398.

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

Ravikumar, Pradeep, Lafferty, John, Liu, Han, and Wasserman, Larry. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

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

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. "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, Debaditya, Murty, K Sri Rama, and Mohan, C Krishna. Feature selection using deep neural networks. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–6. IEEE, 2015.

Scardapane, Simone, Comminiello, Danilo, Hussain, Amir, and Uncini, Aurelio. 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, Le, Smola, Alex, Gretton, Arthur, Borgwardt, Karsten M, and Bedo, Justin. Supervised feature selection via dependence estimation. In *Proceedings of the 24th international conference on Machine learning*, pp. 823–830. ACM, 2007.

Song, Le, Smola, Alex, Gretton, Arthur, Bedo, Justin, and Borgwardt, Karsten. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May):1393–1434, 2012.

Stein, Gary, Chen, Bing, Wu, Annie S, and Hua, Kien 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.

Strobl, Carolin, Boulesteix, Anne-Laure, Kneib, Thomas, Augustin, Thomas, and Zeileis, Achim. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):307, 2008.

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

Verikas, Antanas and Bacauskiene, Marija. Feature selection with neural networks. *Pattern Recognition Letters*, 23(11):1323–1335, 2002.
Yamada, Makoto, Jitkrittum, Wittawat, Sigal, Leonid, Xing, Eric P, and Sugiyama, Masashi. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

Zheng, Grace XY, Terry, Jessica M, Belgrader, Phillip, Ryvkin, Paul, Bent, Zachary W, Wilson, Ryan, Ziraldo, Solongo B, Wheeler, Tobias D, McDermott, Geoff P, Zhu, Junjie, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8:14049, 2017.

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