INVASE+: INSTANCE-WISE VARIABLE SELECTION USING PATH-BASED DERIVATIVES

Tennison Liu
Department of Information Engineering
University of Cambridge, UK
tl522@cam.ac.uk

February 9, 2021

ABSTRACT

This paper presents INVASE+, an instance-wise variable selection method inspired by INVASE [14]. While INVASE uses an actor-critic framework to train a selector function, INVASE+ trains the selector function in an embedded fashion using Gumbel-Softmax approximation and reparameterisation to derive a gradient estimator. The proposed method can be trained end-to-end using backpropagation and achieves comparable performance on feature discovery and predictive performance on synthetic datasets.

1 Introduction

As high-dimensional and big-data level of information become available, complex machine learning methods, such as deep neural networks have achieved state-of-the-art predictive performance on a variety of tasks. However, the decision making process behind such black-box methods are generally difficult to interpret. In domains such as finance, medicine and security, it is essential to verify that the high predictive performance stems from proper problem representation and not exploitation of data artifacts [9]. Interpretability and validation of models are thus key ingredients for full adoption of advanced models.

This paper seeks to shed light on feature importance using an embedded training approach by characterising the contribution of each input feature to a model's prediction in each particular instance. Many studies have proposed methods for selecting globally relevant features, including Sequential Correlation Feature Selection [6], Mutual Information Feature Selection [11] and Knockoff methods [1]. A key limitation of these approaches is that selected features are relevant to prediction on average across the entire dataset (globally relevant) but might not be important for a given sample.

In many cases, such as when the data is heterogeneous, relevant features are likely to differ across samples. As an example, neurologists monitor brain activities in different regions of the brain to investigate epilepsy genesis. The relevant regions behind epileptic episodes differ drastically between patients and also between episodes for the same patient. In such scenarios, instance-wise variable selection becomes more clinically meaningful, compared to discovering the same subset of global features.

This paper extends the instance-wise feature selection method INVASE in a method we termed INVASE+. As opposed to using actor-critic method to derive a surrogate loss for the feature selector function, INVASE+ uses embedded training and a path-based derivative based on Gumbel-Softmax relaxation to backpropagate through sampling. While INVASE is composed of three networks that are trained iteratively, INVASE+ is a single network trained in an end-to-end fashion. Like INVASE, the proposed method is capable of discovering different number of relevant input features for each sample. We compare model performance through feature discovery (true positive and false discovery rate) and predictive performance (area under the curve metrics) on synthetic data.

1.1 Related Work

Instance-wise feature selection methods discover features that are relevant for each sample. Many methods have been proposed for instance-wise selection, which can be separated into two major approaches.

At a high level, feature additive methods assign signed weights to each input feature, indicative of the contributions of the features to prediction. For a model f and instance $\mathbf{x}$, the contributions of each feature i of $\mathbf{x}$ is computed $w_i^{\mathbf{x}}(f)$ such that the sum of all feature contributions approximates $f(\mathbf{x})$, i.e. $\sum_i w_i^{\mathbf{x}} f(\mathbf{x}) \approx f(\mathbf{x})$. LIME [12] linearly regresses predictions on local neighbourhood of the instance to learn contributions. [13] presented DeepLIFT, a feature selection method formulated for deep neural networks by backpropagating contributions back to each feature. [10] provided a unifying interpretation of this class of methods, demonstrating that Shapley values can return feature-additive contributions subject to desired constraints.

Feature selective methods identify subset of features $S(\mathbf{x})$ (ideally small) that approximates the prediction achieved using all features i.e. $f(S(\mathbf{x})) \approx f(\mathbf{x})$. L2X [3] learns $S(\mathbf{x})$ by maximising the mutual information between the subset and the prediction. However, the method assumes a fixed number of important features per instance, which is usually inappropriate in practice. [2] proposes 'sufficient input subsets' and a set of optimisation constraints to search for $S(\mathbf{x})$. While conceptually straightforward, the unconstrained search space is large and can be computationally prohibitive with scale. INVASE [14] addresses the limitations imposed by the previous methods. The model comprises three deep neural networks - selector, predictor and baseline networks. The selector network is trained to predict feature importance based on the input and is optimised through actor-critic framework to allow backpropagation through sampling.

2 Proposed Method

INVASE+ builds closely off of the methods introduced in INVASE. INVASE optimises against the loss function (Equation 1), where the first term (Equation 2) is the KL-divergence between conditional distribution of Y given selected features (particular realisation) and conditional distribution of Y given particular realisation of all features. The second term induces sparsity through l_0 norm.

$$\mathcal{L}(\mathbf{x}, \mathbf{s}) = \mathbb{E}_{\mathbf{x} \sim p_X}[l(\mathbf{x}, S(\mathbf{x})) + \lambda ||S(\mathbf{x})||_0))] \tag{1}$$

where

$$l(\mathbf{x}, \mathbf{s}) = \int_{\mathcal{Y}} p_Y(y|\mathbf{x})[\log(p_Y(y|\mathbf{x}) - \log(p_Y(y|\mathbf{x}^{(s)})]dy \tag{2}$$

Due to the exponential search space of the selector function and the unknown integral in Equation 2, the loss function is estimated with a surrogate loss. To tackle backpropagation through stochastic sampling, INVASE estimates the gradient using score-function (SF) techniques [4]. While the SF estimator only requires that a function $f()$ can be evaluated and is an unbiased estimator, this comes at the cost of much higher variance. Consequently, SF based methods have been found to be empirically slow to converge. In particular, variance of SF estimators scale linearly with the number of dimensions of the sample vector, making it especially challenging for high-dimensional datasets.

The proposed INVASE+ embeds training of selector network with training of the predictor network. Consider a classification task, let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, ..., c\}$ be a discrete label space. Let $D = \{(\mathbf{x}_i, y_i)\}_{i=n}^n$ consists of n i.i.d. realisations i.e. $(\mathbf{x}_i, y_i) \in (\mathcal{X} \times \mathcal{Y})$.

The model consists of a selector function $S^\theta : \mathcal{X} \to [0, 1]^d$, which takes the input features and finds the probability that each feature is important for prediction. Here, S_i^θ denotes probability of selecting feature i.e. $S_i^\theta = p(\mathbf{s}_i = 1)$, where $s = 1$ indicates variable selected. This selector function is approximated with a single neural network and induces a probability distribution over the selection space with the probability of a given joint selection vector $\mathbf{s} \in \{0, 1\}^d$, given by

$$\pi_\theta(\mathbf{x}, \mathbf{s}) = \prod_{i=1}^{d} S_i^\theta(\mathbf{x})^{\mathbf{s}_i}(1 - S_i^\theta(\mathbf{x}))^{1-\mathbf{s}_i} \tag{3}$$

The predictor function $f^\phi : \mathcal{X} \times \{0, 1\}^d \to [0, 1]^c$ takes as input a suppressed feature vector $\mathbf{x}^{(\mathbf{s})}$ based on multiplying the selection vector $\mathbf{s}$ with the input features $\mathbf{x}$ (i.e. $\mathbf{s} \odot \mathbf{x}$) and outputs a probability distribution (using softmax layer) over the c-dimensional output space. The predictor function is implemented as a fully connected neural network. The fully-connected networks used to impelement the selector and predictor function are identical to the architecture proposed in INVASE. Namely, the selector function is a 3-layer network with 100 nodes in the hidden

layers and ReLU activation functions. The predictor function is a 3-layer network with 200 nodes in hidden layers and Batch-Normalisation and ReLU activation after each fully-connected layer.

Both selector and predictor networks are trained in an end-to-end fashion by minimising the loss function in Equation 4. Here, the first term inside the expectation is the cross-entropy loss weighted by class support, where y_i is the ith component of the one-hot encoding of y. The second term is the l_2 penalty to encourage regularisation on the element-wise probabilities predicted by the selector function. $\lambda \in \{1e-4, 5e-4, 1e-3, 5e-3\}$ is a hyperparameter, which adjusts the strength of the penalty term.

$$l(\theta, \phi) = \mathbb{E}_{(\mathbf{x},y)\sim p, \mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)} \left[-\sum_{i=1}^{c} y_i \log(f_i^\phi(\mathbf{x}^{(\mathbf{s})}, \mathbf{s})) + \lambda ||S^\theta(\mathbf{x})||_2 \right] \tag{4}$$

The embedded training strategy takes advantage of the embedded variable selection process to perform feature selection and classification/regression at the same time. As opposed to the iterative training technique adopted in INVASE, embedded training is faster, more accurate and can learn from smaller datasets.

The selection vector $\mathbf{s}$ is sampled from the predicted probabilities $S^\theta(\mathbf{x})$ during operation. Backpropagation cannot be directly applied as the samples are non-differentiable. INVASE overcomes this issue by focusing on score function estimators augmented with variance reduction. Here, an efficient gradient estimator is derived to replace the non-differentiable sample from a Bernoulli distribution with a differential sample from the Gumbel-Softmax distribution [7]. It is important to note that the continuous approximation softens the discrete variables, thereby introducing bias into the estimator. However, the resulting estimator has been found empirically to have lower variance and be more stable.

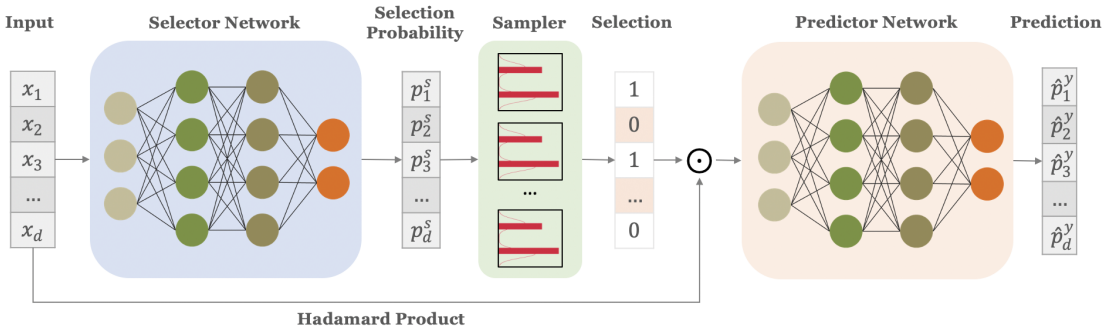


Figure 1: **Block Diagram of INVASE+** Instances are fed into the selector network, which outputs a vector of selection probabilities. The selection vector is then sampled from these probabilities and multiplied with the input features. The suppressed input feature vector is then fed into the predictor network to perform prediction.

Figure 1 illustrates the overall architecture of INVASE+. The network is trained over 2000 iterations with mini-batches of 200 samples using Adam optimiser [8]. Hyperparameters are tuned to maximise predictive performance (as measured through AUROC and AUPRC) on a validation set.

2.1 Gradient Estimator

The Gumbel-Max trick [5] provides a simple reparameterisation to draw samples z from categorical distribution with probabilities π_i:

$$z = \text{one\_hot}(\underset{i}{\text{argmax}}[g_i + \log \pi_i]) \tag{5}$$

where g_i are samples drawn from Gumbel$(0, 1)$. Re-writing Equation 5 for the Bernoulli case gives the below formulation for sampling:

$$\mathbf{s}_i = \begin{cases} 1, & \text{if } \log(S_i^\theta) - \log(1 - S_i^\theta) + g_1 - g_2 > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

As the argmax function is non-differentiable, it is replaced and made differentiable using a soft thresholding operation i.e. sigmoid $\sigma()$ for binary random variables, giving a continuous approximation $\hat{\mathbf{s}}_i$

$$\hat{\mathbf{s}}_i = \frac{\exp\left((\log(S_i^\theta) + g_1)/\tau\right)}{\exp\left((\log(S_i^\theta) + g_1)/\tau\right) + \exp\left((\log(1 - S_i^\theta) + g_2)/\tau\right)} \tag{7}$$

Thus, the sampling operation is performed through a deterministic function dependent on the parameter S^θ and a stochastic component g. The Gumbel-softmax distribution is smooth for $\tau > 0$ and has well-defined gradients $\frac{\partial y}{\partial S^\theta}$ w.r.t. parameter S^θ, allowing backpropagation to be directly applied to compute gradients.

It is important to note that at higher temperature (e.g. $\tau = 10.0$), the approximate distribution converges to a uniform distribution. As $\tau \to 0$, samples from Gumbel-Softmax distribution are identical to those from a Bernoulli distribution. In practice, a bias-variance trade-off exists across different temperatures, where the estimator is unbiased as $\tau \to 0$ but have high variance, and vice-versa. The temperature is annealed from 2.0 to 0.5 using the schedule $\tau = \max(0.5, \exp(-rt) + 1.0)$ of global training step t and $r \in \{1e-5, 1e-4, 1e-3\}$ are hyperparameters.

As the sampled selection vector must be discrete, straight-through estimation is performed on gradients during the backward pass. Specifically, during the forward pass, discretised $\mathbf{s}_i$ are sampled using argmax but the continuous approximation $\hat{\mathbf{s}}_i$ is used to compute gradients in the backward pass. Implementation of INVASE+ can be found at `https://github.com/tennisonliu/invase-net-plus`.

3 Experiments

In this section, INVASE+ is compared against the original INVASE implementation and other state-of-the-art instance-wise variable selection methods on synthetic data. The experiments aim to identify whether the proposed method can discover ground truth relevance and enhance prediction performance.

Six synthetic datasets are generated using the same generative process specified in INVASE. The input features are generated from 11-dimensional independent Gaussian distributions i.e. $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. **Syn1-3** are generated such that labels depend on the same global subset of features, whereas **Syn4-6**, have labels that depend on different features across instances.

3.1 Discovery

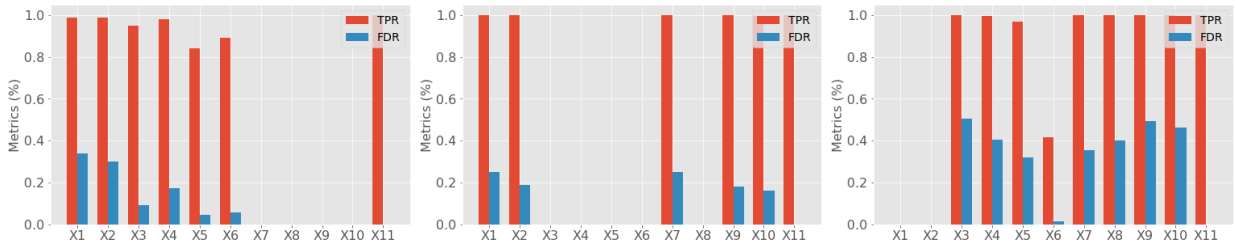
Dataset	**Syn1**		**Syn2**		**Syn3**		**Syn4**		**Syn5**		**Syn6**	
Metrics (%)	TPR	FDR	TPR	FDR	TPR	FDR	TPR	FDR	TPR	FDR	TPR	FDR
INVASE+	**100.0**	**0.0**	**100.0**	**0.0**	**84.4**	**0.0**	**96.6**	**10.1**	**89.8**	**11.3**	**94.5**	**32.6**
INVASE	**100.0**	**0.0**	**100.0**	**0.0**	**92.0**	**0.0**	**99.8**	**10.3**	**84.8**	**1.1**	**90.1**	**7.4**
L2X	100.0	0.0	100.0	0.0	69.4	30.6	79.5	21.8	74.8	26.3	83.3	16.7
LIME	13.8	86.2	100.0	0.0	98.1	1.9	40.7	49.4	41.1	50.6	50.5	49.5
Shapley	60.4	39.6	93.3	6.7	90.1	9.1	65.2	31.9	62.9	33.7	71.2	28.8

Table 1: Relevant feature discovery results for synthetic datasets (11 features)

The quantitative metrics utilised to evaluate discovery are true positive rate (TPR) and false discovery rate (FDR). The discovery performance of INVASE+, compared to other instance-wise selection methods are shown in Table 1. *Note*: apart from INVASE+, all other results are lifted from [14]. Generally, the proposed method is capable of detecting relevant features on a global level, as well as on an instance-wise level. It outperforms all other methods and achieves comparable performance to INVASE in both cases.

On **Syn3**, INVASE+ tends to ignore X_8 as a relevant feature, resulting in a lower TPR. On **Syn5**, **Syn6**, INVASE+ exhibits better performance identifying ground truth relevance features. However, this comes at the expense of higher FDR. The feature-wise discovery performance across **Syn4-6** are plotted below in Figure 2. On **Syn6**, where the false discovery is highest, the logits are generated through two mechanisms. It is evident that the features being falsely discovered correspond to features relevant for alternative data generating process. The same effect exists across **Syn4** and **Syn5** to a lesser degree.

This could be a product of the embedded training of selector and predictor networks. The end-to-end training means that the selector can be influenced by training of the classifier. On **Syn3**, the model routinely ignores X_8 as a relevant metric but the model's predictive performance is on-par with INVASE across all metrics (Table 2). Intuitively, the predictor network learns the influence of X_8 on the prediction and influences the selector to ignore its relevance during selection. On **Syn6**, a similar effect might exist as the predictor influences the selector to select more globally relevant features such that it can learn from the 'richer' input vector.



(a) **Syn4** Discovery (b) **Syn5** Discovery (c) **Syn6** Discovery

Figure 2: Analysis of feature-wise discovery performance (TPR, FDR).

3.2 Prediction

Dataset	AUROC						
	w/o FS	with Global	**INVASE**	**INVASE+**	Tree	L2X	Lasso
Syn1	.578±0.004	.686±0.005	**.690±0.006**	**.697±0.002**	.574±0.101	.498±0.005	.498±0.006
Syn2	.789±0.003	.873±0.003	**.877±0.003**	**.873±0.010**	.872±0.003	.823±0.029	.555±0.061
Syn3	.854±0.004	.900±0.006	**.902±0.003**	**.903±0.003**	.899±0.001	.862±0.009	.886±0.003
Syn4	.558±0.021	.774±0.005	**.787±0.004**	**.773±0.004**	.684±0.017	.678±0.024	.514±0.031
Syn5	.662±0.013	.784±0.005	**.784±0.005**	**.865±0.010**	.741±0.004	.709±0.008	.691±0.024
Syn6	.692±0.015	.858±0.004	**.877±0.003**	**.874±0.005**	.771±0.031	.827±0.017	.727±0.025

Table 2: Predictive performance with and without feature selection methods on synthetic data (11 features)

This section evaluates the predictive performance of models with INVASE+ feature selection as a pre-processing step. As the proposed method contains selector and predictor components in one network, the model's predictive performance is trained in an embedded fashion along with the instance-wise feature selector. Table 2 lists the area under the ROC (AUROC) for model without feature selection and those with global and instance-wise feature selection. *Note:* like before, all metrics except those for INVASE+ are lifted from [14].

Evidently, models with any form of feature selection generally out-perform the model without feature selection. This is likely due to the model overfitting (even with Batch-Normalisation, early-stopping and weight regularisation measures). On **Syn1-3**, the predictive performance between global feature selection and the INVASE variants are comparable. Notably, on **Syn3**, where INVASE+ tends to exclude X_8 as being relevant, the predictive performance is on-par with INVASE. The predictive performance on **Syn4-6** is comparable between the two INVASE variants, while INVASE+ performs significantly better on **Syn5**. This is partially due to the higher TPR, but also the higher FDR, resulting in more irrelevant features. While irrelevant features do not contribute to the prediction, the predictor network can learn these artefacts during training to achieve better predictive performance. Across all synthetic datasets, INVASE+ performs better than Tree, L2X and LASSO based feature selection methods.

4 Discussion

This paper presents INVASE+, an instance-wise feature selection method based on the ideas of INVASE. INVASE+ extends INVASE by training a single network in an end-to-end fashion and uses path-based derivatives to backpropagate error gradients through stochastic sampling layer. This extension was motivated by the high variance and slow convergence observed empirically in many SF based estimators. In place of this unbiased estimator, a biased estimator with lower variance is proposed and derived using Gumbel-Softmax distribution and reparameterisation of the sampling function.

The proposed method demonstrates superior instance-wise feature selection capability when compared to state-of-the-art benchmarks and is comparable to the original INVASE method. The embedded training of the proposed network also achieves strong predictive performance. On feature discovery, INVASE+ is more likely to discover irrelevant features. This is likely due to 1) embedded training of selector network, which can lead to unstable estimation of selector function weights and 2) sub-optimal network design. The selector and predictor network architecture and various model hyperparameters were kept the same as the original INVASE method. As the two are based on different approaches, it is believed that more specified networks and hyperparameter selections can lead to improved performance.

While this work has been limited in scope due to time and computational constraints, further experimentation should be performed to validate the proposed method. This includes moving away from synthetic data and validating on real-world datasets. The proposed method encourages sparsity in the selection vector by placing l_2 penalty on the selection probabilities. A l_1 penalty was placed directly on selection vector for comparison, achieving comparable performance. This penalty can be alternatively implemented as a KL-divergence cost between selection densities and a simple distribution (e.g. Bern$(z;\theta)$ where θ is kept sufficiently small). Lastly, viewed from an attention angle, the sampling of the selection vector can be seen as hard attention. Future works can seek to explicitly use a soft attention mechanism where selection probabilities can be seen as a soft rankings for embedded or post-hoc feature selection.

References

[1] Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: Model-x knockoffs for high-dimensional controlled variable selection. *arXiv preprint arXiv:1610.02351*, 2016.

[2] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576. PMLR, 2019.

[3] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

[4] Michael C Fu. Gradient estimation. *Handbooks in operations research and management science*, 13:575–616, 2006.

[5] EJ Gumbel. The maxima of the mean largest value and of the range. *The Annals of Mathematical Statistics*, pages 76–84, 1954.

[6] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

[7] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Muller, and Wojciech Samek. Analyzing classifiers: Fisher vectors and deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2912–2920, 2016.

[10] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[11] Hanchuan Peng, Fuhui Long, and Chris Ding. 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.

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

[13] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[14] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.