Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Applications Note

OXFORD

Machine Learning

EPS: Automated Feature Selection in Case-Control Studies using Extreme Pseudo-Sampling

Ruhollah Shemirani [1,*], Stephane Wenric [2], Eimear Kenny [2] and José Luis Ambite [1,*]

[1] Information Sciences Institute, University of Southern California, Marina del Rey, US and
[2] Charles Bronfman Institute for Personalized Medicine, Icahn School of Medicine at Mount Sinai, New York, US

* To whom correspondence should be addressed.

Abstract

Summary: Finding informative predictive features in high dimensional biological case-control datasets is challenging. The Extreme Pseudo-Sampling (EPS) algorithm offers a solution to the challenge of *feature selection* via a combination of deep learning and linear regression models. First, using a variational autoencoder, it generates complex latent representations for the samples. Second, it classifies the latent representations of cases and controls via logistic regression. Third, it generates new samples (pseudo-samples) around the extreme cases and controls in the regression model. Finally, it trains a new regression model over the upsampled space. The most significant variables in this regression are selected. We present an open-source implementation of the algorithm that is easy to set up, use, and customize. Our package enhances the original algorithm by providing new features and customizability for data preparation, model training and classification functionalities. We believe the new features will enable the adoption of the algorithm for a diverse range of datasets.
Availability: The software package for Python is available online at https://github.com/roohy/eps
Contact: rshemira@usc.edu

1 Introduction

New biological data collection technologies that yield large numbers of bio-markers in new case-control datasets have made the task of feature selection ever more important and challenging. The two main advantages of feature selection (Hemphill *et al.*, 2014) are (1) bias reduction in very high dimensional datasets, such as the UK Biobank (Sudlow *et al.*, 2015) which contains hundreds of thousands of features, and (2) addressing the curse of dimensionality due to an imbalance between the number of samples and the number of markers, as in datasets like the TCGA (Weinstein *et al.*, 2013) RNA dataset, and ctcRbase (Zhao *et al.*, 2020), which have tens of thousands of markers and only a few thousands of samples. Dimensionality reduction methods, such as PCA, tSNE, and recently, autoencoders are a class of feature selection algorithms that address these issues by creating alternate representations of the data (Hemphill *et al.*, 2014; Tan *et al.*, 2014).

The latent features in autoencoders are derived from complex associations between the original features which increase classification power in bioinformatics applications (Tan *et al.*, 2014). However, extracting the effects of the original feature set on the latent representation of data is challenging due to the non-linearity of such models (Danaee *et al.*, 2017). This disadvantage is especially pronounced in biological applications as classification alone is seldom the sole purpose of the analysis; understanding the causes behind the classification results is often as crucial as the classification itself. The Extreme Pseudo-Sampling (EPS) algorithm, first introduced in Wenric and Shemirani (2018), proposed a solution to address this disadvantage by generating pseudo-samples that highlight the features in the original dataset that are the most influential in case-control prediction. The EPS algorithm was used to extract a gene ranking for cancer survival analysis; and was able to outperform traditional linear methods in 9 out of 12 datasets obtained from the TCGA RNA expression dataset without using the survival data as input.

EPS uses a Variational AutoEncoder (VAE) technique to extract a latent representation of the data. Every sample x_i is assigned a probability distribution $P(z|x)$ with estimated mean $\hat{z_i}$. Unlike encoding, decoding (reconstructing) $\hat{x_i}$ from $\hat{z_i}$ is deterministic. Further, the VAE model ensures that all z_i are located in close proximity. These features ensure that decoding points around each $\hat{z_i}$ would result in new distinct pseudo-samples in the original feature set that have similar features to x_i while following the statistical properties of all samples in the dataset.

Despite its potential, both implementation and customization of the EPS algorithm requires proficiency in deep learning programming and theory. Here, we present an open-source python package to streamline the use of EPS. We have automated non-trivial steps of the algorithm along with the required data management methods into single function calls in

the pipeline. Not only does it offer a ranking of all features, it also enables the usage of its hidden predictive model for classification purposes. Below, we first briefly describe the EPS algorithm and then describe the features of our enhanced EPS package.

2 EPS Algorithm

The main steps of the EPS algorithm are shown in Supplementary Figure 1. First, a multi-layered Variational AutoEncoder (VAE; Kingma and Welling (2013)) is trained using the `train` function. This VAE uses fully connected layers to cover associations between any possible set of features. When trained, the VAE offers a compressed representation of the samples in a latent space, where cases and controls can be separated by a hyper plane (Wenric and Shemirani, 2018). In the second step of the pipeline, the `generate` function fits a logistic regressor on the latent representations of the samples based on the case-control labels to find the best separating hyperplane. On each side of the regressor hyper plane, a user-defined number of samples with the highest distance from the hyper plane are selected as seeds for randomized generation of pseudo-samples not present in the original dataset. The pseudo-samples have the same labels as their seeds. The pseudo-samples are generated using a multivariate normal distribution with a seed as the mean. Third, representations of the pseudo-samples in the original feature space are generated using the decoder of the VAE trained in the first step. Unlike the original samples, the new pseudo-samples can easily be classified by a logistic regression model. Informative predictive features in the original feature space are made salient by the pseudo-sample-enriched regression. EPS provides a ranking of the features based on their fitted weights of this new logistic regressor.

3 EPS Software Features

We describe select capabilities of the EPS package below:

- The EPS package uses the Tensorflow[1] library to build the VAE. Thus, through the configuration of the Tensorflow software on the host machine, it can also be run on graphical processing units (GPUs) for higher efficiency.
- The EPS package accepts input data in a Numpy array format. This format is supported by most python data packages, which makes the process convenient.
- The input should be standardized so that their range is between zero and one. Thus, the EPS package normalizes the data by default, unless it is instructed otherwise.
- The Tensorflow graph is automatically reset when switching between models to avoid model errors in the Tensorflow library. To reduce the memory load caused by this switching process, EPS also clears the memory when switching between models using the garbage collection mechanism.
- The VAE uses the RMSProp optimization algorithm while the logistic regression models use Adam optimizer (Kingma and Ba, 2014). The learning rate for these optimizers can be set separately to increase statistical power through parameter optimization.
- The VAE architecture in our software can be modified both in terms of number of layers of the network, and number of nodes in each layer using the `set_layers` function. Further, users can also select the activation function for middle layers from all the activation functions available in the Tensorflow library.
- The batch sizes and the number of epochs for training can be set separately for every training function (VAE, latent regression, main regression). The EPS automatically randomizes the dataset between epochs.
- The number of selected seeds and extreme pseudo-samples generated for each of them can be passed to the pipeline as arguments. The variance of the pseudo-sampling process can be adjusted to control the divergence of the pseudo-samples from the seed data.
- The pseudo-sample dataset is balanced (equal number of generated cases and controls) to improve the regression performance and overcome the imbalance observed in some case-control studies, such as the TCGA gene expression dataset.
- The linear regressor in the latent space can also be used for classification purposes, although it is not recommended for high dimensional data (number of features $> 20,000$) on personal computing devices due to memory limitations.
- Pseudo-samples are now accessible to the users upon generation, facilitating the augmentation of other feature selection algorithms via pseudo-sampling.
- When using the `generate` function to generate the pseudo-samples using the regressor, the EPS can use a subset of VAE training data and labels for the process. This enables the usage of multiple case-control datasets to train the VAE, both to reduce the VAE bias and to enable the analysis of small datasets. Assigning differentiating labels to each dataset is not required.

[1] https://github.com/tensorflow/tensorflow

4 Conclusion

The EPS package provides a feature selection process that takes complex data associations into account, while abstracting away the non-trivial technical challenges of deep learning libraries and model and data management in machine learning. We hope this enables new applications for this algorithms. We plan to add more customization options in the future. The new features, such the ability to choose the random distribution for the pseudo-samples, will further increase the adaptability of the software.

Conflict of interest: none declared.

References

Danaee, P. *et al.* (2017). A deep learning approach for cancer detection and relevant gene identification. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2017*, pages 219–229. World Scientific.

Hemphill, E. *et al.* (2014). Feature selection and classifier performance on diverse bio-logical datasets. In *BMC bioinformatics*, volume 15, page S4. Springer.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Sudlow, C. *et al.* (2015). Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *Plos med*, **12**(3), e1001779.

Tan, J. *et al.* (2014). Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. In *Pacific Symposium on Biocomputing Co-Chairs*, pages 132–143. World Scientific.

Weinstein, J. N. *et al.* (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, **45**(10), 1113.

Wenric, S. and Shemirani, R. (2018). Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, **9**, 297.

Zhao, L. *et al.* (2020). ctcrbase: the gene expression database of circulating tumor cells and microemboli. *Database*, **2020**.