STab: Self-supervised Learning for Tabular Data

Ehsan Hajiramezanali† Nathaniel Diamant Gabriele Scalia Max W. Shen

Department of Artificial Intelligence and Machine Learning,
Genentech Research and Early Development
†{hajiramezanali.ehsan}@gene.com

Abstract

Self-supervised learning in tabular data is understudied:

- Unlike its image and language counterparts which have unique spatial or semantic structure information, it is difficult to design an effective augmentation method generically beneficial to downstream tasks in the tabular setting, owing to its *lack of common structure and diverse nature*.
- Most existing augmentation methods are *domain-specific* (such as rotation in vision, token masking for NLP, and edge dropping for graphs), making them less effective for real-world tabular data.

This significantly limits tabular self-supervised learning and hinders progress in this domain. Aiming to fill this crucial gap, we propose **STab**, an augmentation-free self-supervised representation learning based on stochastic regularization techniques that does not rely on negative pairs, to capture highly heterogeneous and non-structured information in tabular data.

Our experiments show that **STab** achieves state-of-the-art performance compared to existing contrastive and pretext task self-supervised methods.

Introduction

Human learning in the real world builds mental representations that are robust to different views or distortions of an identity. With this in mind, when designing algorithms that imitate the human learning process, we seek a multi-view learning model that can learn representations invariant to a family of viewing conditions.

Contrastive learning between multiple views of the data often fits such a description well by bringing two views of the same scene together in the representation space, while pushing those of different scenes (*negative samples*) apart.

However, their performance critically depends on the choice of input augmentations. In addition, these methods rely on memory banks, large batch sizes, or customized mining strategies to retrieve the negative pairs.

The augmentation steps to generate views or corruptions are mostly domain-specific (e.g. cropping, rotation, color transformation in vision, token masking in NLP, node/edge dropping in graph), making them less effective in the tabular data commonly used in many fields such as healthcare, advertisement, finance, etc.

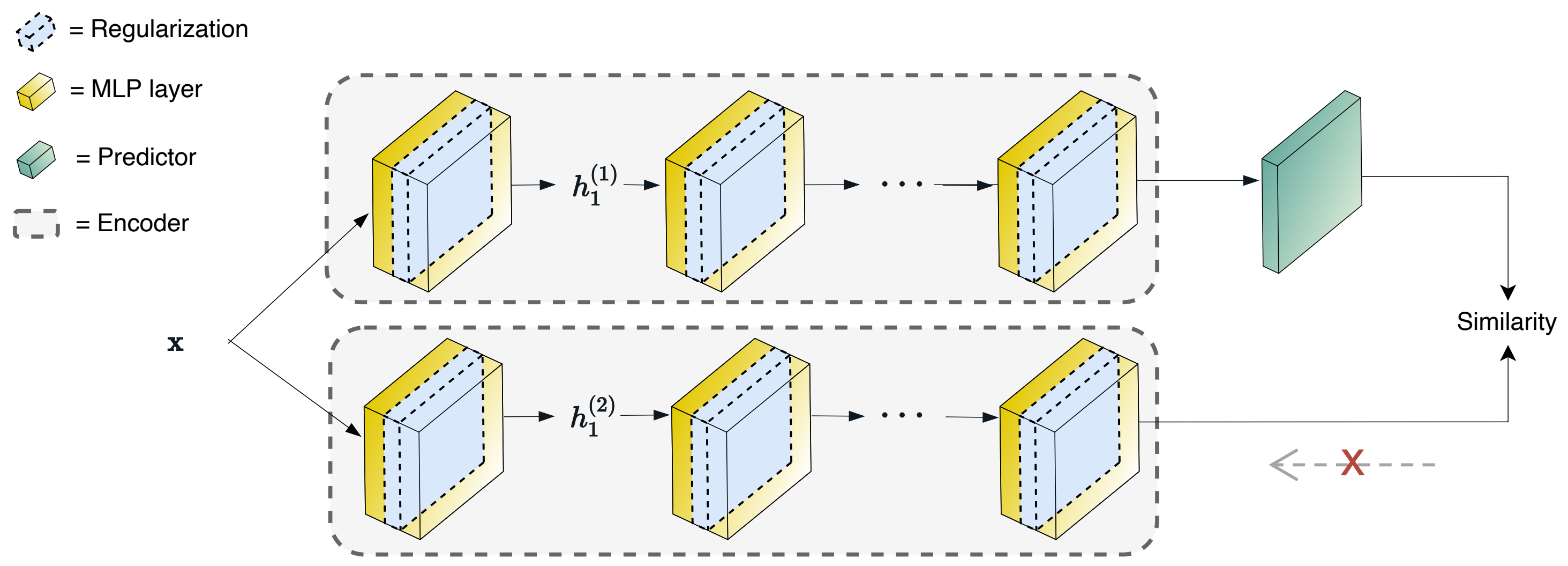
Instead of applying augmentations over the input samples to make different views of data for contrastive learning, we propose to apply augmentations to every layer of encoders. As a result, the proposed **Self-supervised learning for Tabular data (STab)** relies on two (or multiple) weight-sharing neural networks with different regularizations applied to a single input.

By exploiting the stop-gradient [2] operation technique, the proposed weight-sharing networks can model invariance with respect to more complicated regularizations while it will not converge to an undesired trivial solution.

Method

STab takes an unlabeled tabular sample $\mathbf{x} \in \mathbb{R}^M$ as input. The input sample is then processed by two encoder multi-layer perceptron (MLP) networks f_1 and f_2. While the weight parameters of the encoders are shared, they have two different stochastic regularizations. In addition, a projection head g, which is a MLP, transforms the output of one encoder and matches it to the output of the other encoder.

Figure 1: A schematic illustration of the proposed augmentation-free self-supervised learning for tabular data.



Denoting the two output vectors by $\mathbf{y}_1 = g(f_1(\mathbf{x}))$ and $\mathbf{z}_2 = f_2(\mathbf{x})$, we use the negative cosine distance as a measure of similarity:

$$\mathcal{D}(\mathbf{y}_1, \mathbf{z}_2) = -\frac{\mathbf{y}_1}{||\mathbf{y}_1||_2} \cdot \frac{\mathbf{z}_2}{||\mathbf{z}_2||_2},$$

where $||\cdot||_2$ is l_2-norm. We optimize the following symmetric loss:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(\mathbf{y}_1, \mathbf{z}_2) + \frac{1}{2}\mathcal{D}(\mathbf{y}_2, \mathbf{z}_1).$$

To avoid converging to trivial solution, similar to [2], we need to make sure the encoder f_2 receives no gradient from $\mathbf{z}_2$ in the first term, but it receives gradients from $\mathbf{y}_2$ in the second term (and vice versa for f_1).

In order to regularize each encoder, we impose dynamic sparsity within the model. Specifically, similar to DropConnect [5], the fully-connected layers become a sparsely connected layer in which the connections are chosen at random during the training. Please note that this is different from considering the weights of the linear layer to be a fixed sparse matrix during training.

Let's denote the output of hidden layers for view i by $\mathbb{H}^{(i)} = \{\mathbf{h}_j^{(i)}\}_{j=0}^{L}$ with $\mathbf{h}_0^{(1)} = \mathbf{h}_0^{(2)} = \mathbf{x}$ being the input data and L as the number of layers. For each layer of the encoders, the output is given as:

$$\mathbf{h}_j^{(i)} = \sigma\left(\left(\mathbf{M}_j^{(i)} \odot \mathbf{W}_j\right)\mathbf{h}_{j-1}^{(i)}\right), \quad \text{for} \quad i = 1, 2$$

where $\mathbf{M}_j^{(i)}$ is a binary matrix encoding the connection information for f_i and $\mathbf{M}_{j,mn}^{(i)} \sim \text{Bernoulli}(p_j^{(i)})$, $\mathbf{W}$ is the shared weight parameters across encoders, and σ is a non-linear activation function. Note that each element of the mask $\mathbf{M}_j^{(i)}$, i.e. $\mathbf{M}_{j,mn}^{(i)}$, is drawn independently for each sample during training, essentially instantiating a different connectivity for each sample seen.

Results

We compare our STab with existing self-supervised learning SOTA methods for tabular data. VIME-self and SubTab [4] can be categorized as an autoencoder-based model, while SCARF [1] is a contrastive model based on InfoNCE loss.

Following the experimental setting in SCARF [1], all encoders are four-layer [256, 256, 256, 256] dimensional fully-connected NN while the projection head is a two-layer [256, 256] dimensional fully-connected NN.

For all models, we train and evaluate them with 10 different random seeds. Evaluation of these models is done by training a logistic regression model using the embeddings of the training set (i.e. 80% of the data), and by testing it using the embeddings of the test set (20% of the data).

Similar to SCARF, we use ReLU as activation functions for all experiments. Please note that we follow SubTab for the preprocessing of each dataset.

Table 1: Performance of our STab and baselines in terms of classification accuracy (in %). * In STab w/ DropOut we used DropOut to mask weight instead of DropConnect.

Method	Income	Gesture	Robot	Theorem
Raw features	82.28±0.08	46.93±1.05	68.46±1.34	46.96±0.1
VIME-self	82.43±0.16	46.08±0.37	74.23±1.21	44.99±0.9
SubTab	83.97±0.31	52.03±0.98	88.21±0.72	50.81±0.76
SCARF	83.96±0.23	52.28±1.04	83.51±0.86	51.06±1.09
STab w/ DropOut *	81.37±1.13	51.81±0.95	81.28±0.85	48.88±1.22
STab	**84.53 ±0.11**	**53.08 ±0.91**	**88.40 ±0.82**	**55.06 ±0.28**

Related Works

A few recent works have explored applying augmentations to the model weights instead of the data. MetAug applies meta-learned augmentations to the output of the encoder portion of their neural networks. It focuses on one layer of the neural network and the results are focused on natural image data, whereas STab augments multiple layers and is intended for non-structured data.

Similar to STab, the sentence embedding method SimSCE uses dropout on model weights rather than input augmentations. STab builds on this approach by removing the need for negative views by using stop-gradient. STab also replaces DropOut with DropConnect, which we found led to better results

Conclusions

We seek a well-designed augmentation-free self-supervised representation learning to capture highly heterogeneous and non-structured information in tabular data.

The stochastic regularization techniques used in STab is a more effective approach for modeling invariance than random augmentation over the input such as the one in SCARF.

Comparing DropConnect with DropOut in STab framework is demonstrating that DropConnect, which generalizes Dropout to the entire connectivity structure of a fully connected neural network layer, is a more powerful augmentation in the tabular settings and critical to achieve better performance compared to other baselines.

A different interpretation of STab is through the lens of stochastic functions. It is well-known that neural networks with stochastic regularization are random functions [3]. We can interpret STab as siamese model with two different stochastic functions as encoders. Different stochasticities in the encoders produces different views of data. One possible avenue for future works is employing other classes of stochastic functions such as neural processes as encoder.

Another avenue for further improvements is learning the drop rates of binary masks throughout a hierarchical Bayesian model or bi-level optimization which leads to a more flexible and versatile model.

References

[1] Dara Bahri et al. “SCARF: Self-supervised contrastive learning using random feature corruption”. In: *ICLR*. 2021.

[2] Xinlei Chen and Kaiming He. “Exploring simple siamese representation learning”. In: *CVPR*. 2021.

[3] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[4] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. “SubTab: Subsetting Features of Tabular Data for Self-Supervised Representation Learning”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[5] Li Wan et al. “Regularization of neural networks using dropconnect”. In: *ICML*. 2013.