《UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING RANDOM DISTANCES》

Introduction

目前提出大部分的无监督方法都是自监督方法，但都是文本/图像/视频数据，不适合表哥数据，因为表格数据不包含所需要的时间/空间监控信息。

本文重点研究高维表格数据的无监督表示学习

传统方法可以很好处理表格数据，但是通常计算成本太高

- 随机投影
- 主成分分析
- 流形学习

这篇文章中，我们介绍了一个随机距离预测(RDP)模型，它训练神经网络预测随机投影空间中的数据距离

当距离信息在数据中捕获非固有的类结构时，通过优化表示学习器来学习类结构，使预测误差最小化。

随机映射是一种高效且经过理论证明的近似保持距离的方法。因此，我们利用投影空间中的距离来学习所需的特征。

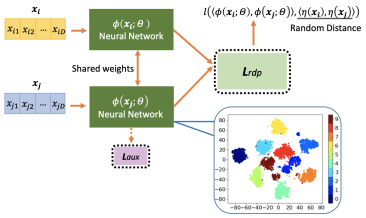
从直观上看，随机映射在底层数据分布假设不准确的情况下，保留了丰富的局部接近信息，但也可能保持了误导性的接近;通过最小化随机距离预测误差，RDP实质上利用了保留的数据接近性和神经网络的能力来学习全局的接近性和纠正不一致的接近性信息，从而得到一个比原始空间更好的表示空间。

contributions

我们提出了一个随机距离预测公式，它是非常简单的，但提供了一个高效的监督信号学习表达特征表示，优化了随机投影中的距离。所学习的特征是足够通用的，并且能够很好地完成不同的下游学习任务，

们的公式灵活地纳入了任务相关的辅助损失，这些损失是随机距离预测的组成部分，以进一步增强已学习的特征，也就是说，特征是专门为下游任务优化的，同时尽可能保留一般接近性。

因此，在19个真实的高维表格数据集上，我们的实例化模型称为RDP，它在两个关键的非监督任务(异常检测和聚类)中，比最先进的竞争方法具有更大的性能。



model

将原始数据顺到一个权值共享的学生网络中，学到一个从D维度到M维度的表示学习器

$$\phi(\mathbf{x};\Theta).\ \phi:\mathbb{R}^D\mapsto\mathbb{R}^M$$

表示学习神经网络损失函数如下

$$L_{rdp}(\mathbf{x}_i,\mathbf{x}_j)=l(\langle\phi(\mathbf{x}_i;\Theta),\phi(\mathbf{x}_j;\Theta)\rangle,\langle\eta(\mathbf{x}_i),\eta(\mathbf{x}_j)\rangle),$$

l是一个比较函数，n是一个固定的映射函数

在本文中，我们利用随机投影空间中的内积作为距离/相似性的来源，因为它是非常有效的，并且在保持真实距离信息方面有很强的理论支持

$$L_{rdp}(\mathbf{x}_i,\mathbf{x}_j)=\langle\phi(\mathbf{x}_i;\Theta)\cdot\phi(\mathbf{x}_j;\Theta)-\eta(\mathbf{x}_i)\cdot\eta(\mathbf{x}_j)\rangle^2,$$

$$L_{aux}^{ad}(\mathbf{x})=\left(\phi(\mathbf{x};\Theta)-\eta(\mathbf{x})\right)^2.$$

异常检测任务的损失函数

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In ICLR, 2019.

异常点在这个损失函数中会比正常点得到更大的值，这个值可以直接用于异常检测

模型的理论分析

随机投影是一种简单而有效的线性特征映射技术，证明了其距离保持能力。

$$\Pr\left(|\hat{\mathbf{x}}_i\cdot\hat{\mathbf{x}}_j-f(\hat{\mathbf{x}}_i)\cdot f(\hat{\mathbf{x}}_j)|\geq\epsilon\right)\leq 4e^{\frac{-(\epsilon^2-\epsilon^3)K}{4}},$$

上公式指出随机投影空间中的内积可以在很大程度上保留原空间中的内积，特别是当投影维数K较大时。

相关工作

自监督学习

自监督是近年来兴起的一种最流行、最有效的表征学习方法

许多自我监督方法通过预测某种"上下文"信息。如空间或时间邻域信息来学习高级表示

NLP中流行的分布式表示学习技术，如CBOW/skip-gram

NLP

短语/句子嵌入技术

预测文本片段(例如，单词/短语/句子)，使用其周围的片段作为联合文本来学习表征

应用

CV

对缺失像素patch的预测

两个patch的相对位置

学习顺序帧的时间顺序　　视频数据

GAN

其他方法

PCA或多种学习方法的一个臭名昭著的问题是，由于代价高昂的邻域搜索和/或特征分解，它们在处理大规模高维数据时的计算成本令人望而却步

随机投影是一种计算效率高的方法，被证明的距离保存理论如Johnson- lindenstrauss引理支持[Johnson & Lindenstrauss, 1984]。实验表明，通过随机投影获取的保留距离可以有效地监督表示学习。

自编码器网络是另一种广泛使用的有效的特征学习方法，它通过最小化重构误差来学习低维表示。自动编码器的一个主要问题是它们只关注于保持全局信息，这可能会导致局部结构信息的丢失。

一些表示学习方法是专门为异常检测设计的[Pang et al., 2018a;贾等人，2018年;Burda等人，2019年]。相比之下，我们的目标是学习通用的表示，同时灵活地选择性地添加任务依赖的损失，以学习特定于任务的语义丰富的表示。

EXPERIMENTS

我们的RDP模型与五种最先进的方法进行了比较，包括iForest 、autoencoder (AE)、REPEN、DAGMM 和RND。iForest和AE是两个最流行的基线。另外三种方法学习了用于异常检测的表示法。

Table 1: AUC-ROC (mean±std) performance of RDP and its five competing methods on 14 datasets.

Data Characteristics				Our Method RDP and Its Five Competing Methods					
Data	N	D	Anomaly (%)	iForest	AE	REPEN	DAGMM	RND	RDP
DDoS	464,976	66	3.75%	0.880 ± 0.018	0.901 ± 0.000	0.933 ± 0.002	0.766 ± 0.019	0.852 ± 0.011	**0.942 ± 0.008**
Donors	619,326	10	5.92%	0.774 ± 0.010	0.812 ± 0.011	0.777 ± 0.075	0.763 ± 0.110	0.847 ± 0.011	**0.962 ± 0.011**
Backdoor	95,329	196	2.44%	0.723 ± 0.029	0.806 ± 0.007	0.857 ± 0.001	0.813 ± 0.035	**0.935 ± 0.002**	0.910 ± 0.021
Ad	3,279	1,555	13.99%	0.687 ± 0.021	0.703 ± 0.000	0.853 ± 0.001	0.500 ± 0.000	0.812 ± 0.002	**0.887 ± 0.003**
Apascal	12,695	64	1.38%	0.514 ± 0.051	0.623 ± 0.005	0.813 ± 0.004	0.710 ± 0.020	0.685 ± 0.019	**0.823 ± 0.007**
Bank	41,188	62	11.26%	0.713 ± 0.021	0.666 ± 0.000	0.681 ± 0.001	0.616 ± 0.014	0.690 ± 0.006	**0.758 ± 0.007**
Celeba	202,599	39	2.24%	0.693 ± 0.014	0.735 ± 0.002	0.802 ± 0.002	0.680 ± 0.067	0.682 ± 0.029	**0.860 ± 0.006**
Census	299,285	500	6.20%	0.599 ± 0.019	0.602 ± 0.000	0.542 ± 0.003	0.502 ± 0.003	**0.661 ± 0.003**	0.653 ± 0.004
Creditcard	284,807	29	0.17%	0.948 ± 0.005	0.948 ± 0.000	0.950 ± 0.001	0.877 ± 0.005	0.945 ± 0.001	**0.957 ± 0.005**
Lung	145	3,312	4.13%	0.893 ± 0.057	0.953 ± 0.004	0.949 ± 0.002	0.830 ± 0.087	0.867 ± 0.031	**0.982 ± 0.006**
Probe	64,759	34	6.43%	0.995 ± 0.001	0.997 ± 0.000	0.997 ± 0.000	0.953 ± 0.008	0.975 ± 0.000	**0.997 ± 0.000**
R8	3,974	9,467	1.28%	0.841 ± 0.023	0.835 ± 0.000	**0.910 ± 0.000**	0.760 ± 0.066	0.883 ± 0.006	0.902 ± 0.002
Secom	1,567	590	6.63%	0.548 ± 0.019	0.526 ± 0.000	0.510 ± 0.004	0.513 ± 0.010	0.541 ± 0.006	**0.570 ± 0.004**
U2R	60,821	34	0.37%	**0.908 ± 0.001**	0.987 ± 0.000	0.978 ± 0.000	0.945 ± 0.028	0.981 ± 0.001	0.986 ± 0.001

Table 2: AUC-PR (mean±std) performance of RDP and its five competing methods on 14 datasets.

Data	iForest	AE	REPEN	DAGMM	RND	RDP
DDoS	0.141 ± 0.020	0.248 ± 0.001	0.300 ± 0.012	0.038 ± 0.000	0.110 ± 0.013	**0.301 ± 0.028**
Donors	0.124 ± 0.006	0.138 ± 0.007	0.120 ± 0.032	0.070 ± 0.024	0.201 ± 0.033	**0.432 ± 0.061**
Backdoor	0.045 ± 0.007	0.065 ± 0.004	0.129 ± 0.001	0.034 ± 0.023	**0.433 ± 0.015**	0.305 ± 0.008
Ad	0.363 ± 0.061	0.479 ± 0.000	0.600 ± 0.002	0.140 ± 0.000	0.473 ± 0.009	**0.726 ± 0.007**
Apascal	0.015 ± 0.002	0.023 ± 0.001	0.041 ± 0.001	0.023 ± 0.009	0.021 ± 0.005	**0.042 ± 0.003**
Bank	0.293 ± 0.023	0.264 ± 0.001	0.276 ± 0.001	0.150 ± 0.020	0.258 ± 0.006	**0.364 ± 0.013**
Celeba	0.060 ± 0.006	0.082 ± 0.001	0.081 ± 0.001	0.037 ± 0.017	0.068 ± 0.010	**0.104 ± 0.006**
Census	0.071 ± 0.004	0.072 ± 0.000	0.064 ± 0.005	0.061 ± 0.001	0.081 ± 0.001	**0.086 ± 0.001**
Creditcard	0.145 ± 0.031	**0.382 ± 0.004**	0.359 ± 0.014	0.010 ± 0.012	0.290 ± 0.012	0.363 ± 0.011
Lung	0.379 ± 0.092	0.565 ± 0.022	0.429 ± 0.005	0.042 ± 0.003	0.381 ± 0.104	**0.705 ± 0.028**
Probe	0.923 ± 0.011	0.944 ± 0.002	**0.964 ± 0.000**	0.409 ± 0.153	0.609 ± 0.014	0.955 ± 0.002
R8	0.076 ± 0.018	0.097 ± 0.006	0.083 ± 0.000	0.019 ± 0.011	0.134 ± 0.031	**0.146 ± 0.017**
Secom	0.106 ± 0.007	0.093 ± 0.000	0.091 ± 0.001	0.066 ± 0.002	0.086 ± 0.002	**0.096 ± 0.001**
U2R	0.180 ± 0.018	0.230 ± 0.004	0.116 ± 0.007	0.025 ± 0.019	0.217 ± 0.011	**0.261 ± 0.005**

此外，一个推进过程被用来过滤掉5%可能的异常每次迭代，以迭代地改进RDP的建模。这是因为当出现异常时，模型在很大程度上是有偏差的。在4.1.3节的消融研究中，我们将展示所有这些组件的作用。

Table 3: AUC-ROC results of anomaly detection (see Appendix C for similar AUC-PR results).

	Decomposition				Supervision Signal	
Data	RDP	RDP\ L_{rdp}	RDP\ L_{aux}^{ad}	RDP\Boosting	Org_SS	SRP_SS
DDoS	**0.942 ± 0.008**	0.852 ± 0.011	0.931 ± 0.003	0.866 ± 0.011	0.924 ± 0.006	0.927 ± 0.005
Donors	**0.962 ± 0.011**	0.847 ± 0.011	0.737 ± 0.006	0.910 ± 0.013	0.728 ± 0.005	0.762 ± 0.016
Backdoor	0.910 ± 0.021	0.935 ± 0.002	0.872 ± 0.012	**0.943 ± 0.002**	0.875 ± 0.002	0.882 ± 0.010
Ad	**0.887 ± 0.003**	0.812 ± 0.002	0.718 ± 0.005	0.818 ± 0.002	0.696 ± 0.003	0.740 ± 0.008
Apascal	**0.823 ± 0.007**	0.685 ± 0.019	0.732 ± 0.007	0.804 ± 0.021	0.604 ± 0.032	0.760 ± 0.030
Bank	**0.758 ± 0.007**	0.690 ± 0.006	0.684 ± 0.004	0.736 ± 0.009	0.684 ± 0.002	0.688 ± 0.015
Celeba	**0.860 ± 0.006**	0.682 ± 0.029	0.709 ± 0.005	0.794 ± 0.017	0.667 ± 0.033	0.734 ± 0.027
Census	0.653 ± 0.004	**0.661 ± 0.003**	0.626 ± 0.006	0.661 ± 0.001	0.636 ± 0.006	0.560 ± 0.006
Creditcard	**0.957 ± 0.005**	0.945 ± 0.001	0.950 ± 0.000	0.956 ± 0.003	0.947 ± 0.001	0.949 ± 0.003
Lung	**0.982 ± 0.006**	0.867 ± 0.031	0.911 ± 0.006	0.968 ± 0.018	0.884 ± 0.018	0.928 ± 0.008
Probe	0.997 ± 0.000	0.975 ± 0.000	**0.998 ± 0.000**	0.978 ± 0.001	0.995 ± 0.000	0.997 ± 0.001
R8	0.902 ± 0.002	0.883 ± 0.006	0.867 ± 0.003	0.895 ± 0.004	0.830 ± 0.005	**0.904 ± 0.005**
Secom	**0.57 ± 0.004**	0.541 ± 0.006	0.544 ± 0.011	0.563 ± 0.008	0.512 ± 0.007	0.530 ± 0.016
U2R	0.986 ± 0.001	0.981 ± 0.001	0.987 ± 0.000	**0.988 ± 0.002**	0.987 ± 0.000	0.981 ± 0.002
#wins/draws/losses (RDP vs.)		13/0/1	13/0/1	12/0/2	10/2/2	6/0/8