北京交通大學

当生理时间序列分类遇上深度学习

贾子钰

北京交通大学 计算机与信息技术学院

ziyujia@bjtu.edu.cn

目 录

- 时间序列概述
- 生理时间序列概述
- 相关深度学习模型
  - GraphSleepNet （IJCAI 2020)
  - SST-EmotionNet (ACM MM 2020)
- 总结

何为时间序列？

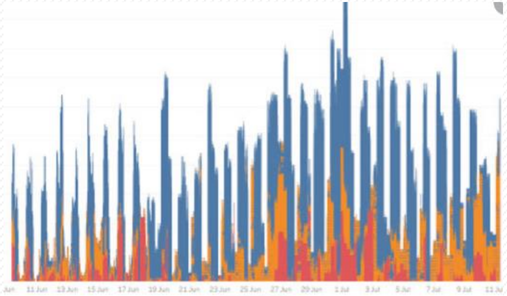
时间序列

◆时间序列

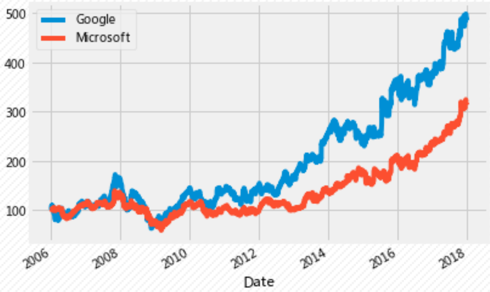
时间序列是按照时间排序的一组随机变量，它通常是在相等间隔的时间段内依照给定的采样率对某种潜在过程进行观测的结果。

◆时间序列的分析与挖掘

在过去的二十年中，时间序列的分析与挖掘被认为是数据挖掘中最具挑战性的问题之一[1]。



交通流量时间序列



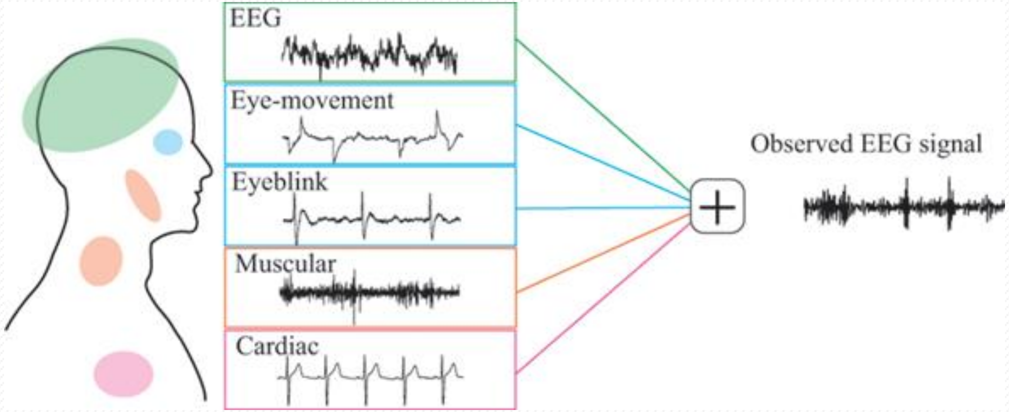
金融时间序列

[1] Fawaz, Hassan Ismail, et al. "Deep learning for time series classification: a review." Data Mining and Knowledge Discovery 33.4 (2019): 917-963.

何为生理时间序列？

生理时间序列

心电ECG



脑电EEG

生理时序-应用

◆医疗健康

重获肢体能力、运动康复、与周围环境进行交流、基于可穿戴设备进行健康评估

生理时序-应用

- 教育

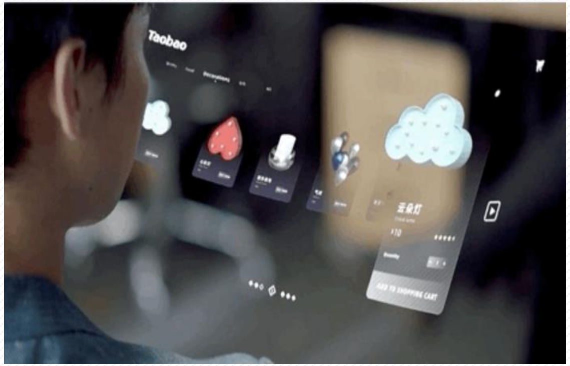
学生注意力值的实时探测和训练

- 军事

脑控无人机、无人车、机器人

- 智能家居
- 刑侦审讯
- 娱乐（游戏）

- 电子商务



淘宝造物节黑科技-淘宝意念购

- 交通安全

疲劳检测、驾驶员精神状况监测

GraphSleepNet: Adaptive Spatial-Temporal Graph Convolutional Networks for Sleep Stage Classification

论文链接：https://www.ijcai.org/Proceedings/2020/184

论文代码：https://github.com/ziyujia/GraphSleepNet

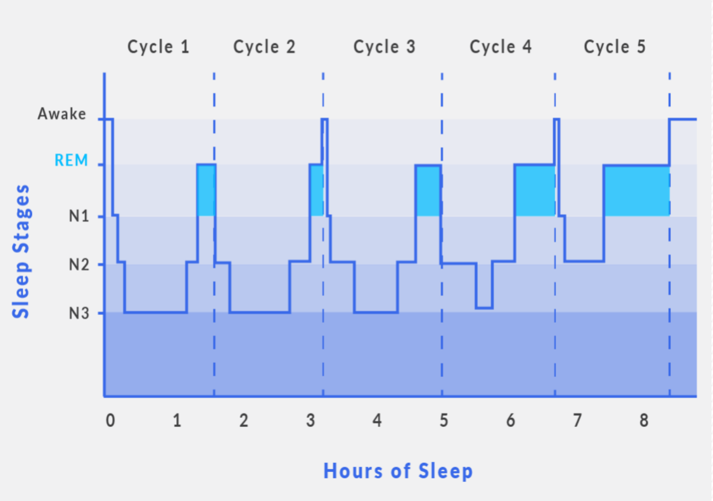
Introduction

睡眠：

- 人的一生中约有三分之一的时间都在睡眠中度过，睡眠质量的好坏直接影响到人类的身心健康；
- 睡眠分期是评估睡眠质量和诊断睡眠障碍的重要手段。

人工睡眠分期：

- 睡眠专家会根据睡眠分期标准和多导睡眠图（PSG）对睡眠阶段进行划分；
- 人工睡眠分期是一项繁琐且耗时的任务；
- 睡眠专家的主观性和可变性易影响睡眠分期的结果。



自动睡眠分期：

- 提高传统睡眠分期的效率；
- 具有重要的临床价值。

Related Work

睡眠分期

- **传统机器学习：**

◆ 支持向量机和随机森林等方法。

◆ 需要手工设计特征，且要求大量的先验领域知识。

- **卷积神经网络和递归神经网络：**

◆ FDCCNN[1]，SeqSleepNet[2]，DeepSleepNet[3]等。

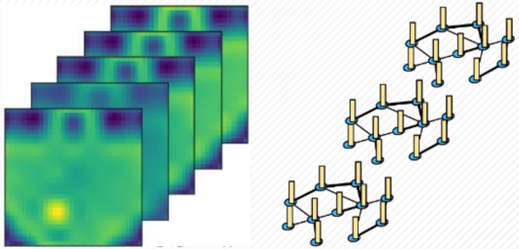
◆ 输入必须是网格数据 (类似于图像)。

Motivation & Challenge

· **网格数据的局限性：**

◆ 大脑区域间的连接关系被忽视。

◆ 由于大脑处于非欧氏空间，因此图是最适合用于表示大脑连接性的数据结构。



· **对大脑的功能连接进行建模：**

◆ 图卷积神经网络在处理图数据中展示了相当优异的表现[4,5]。

◆ 现有的工作往往使用的是固定的图结构，但是睡眠是一个动态的过程。

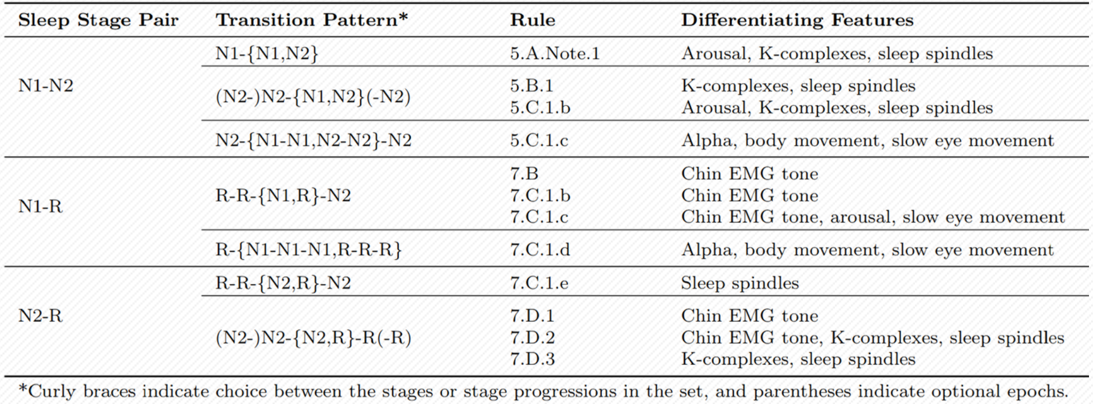
◆ 人类对大脑的理解是有限的。

挑战1：如何为睡眠分期确定合适的图结构。

Motivation & Challenge

挑战2：如何有效地提取时空特征。

- 在睡眠期间，大脑区域的空间特性是不同的。
- 在时间维度上，睡眠阶段之间存在着过渡规则。



Sleep Stage Pair	Transition Pattern*	Rule	Differentiating Features
N1-N2	N1-{N1,N2}	5.A.Note.1	Arousal, K-complexes, sleep spindles
	(N2-)N2-{N1,N2}(-N2)	5.B.1 5.C.1.b	K-complexes, sleep spindles Arousal, K-complexes, sleep spindles
	N2-{N1-N1,N2-N2}-N2	5.C.1.c	Alpha, body movement, slow eye movement
N1-R	R-R-{N1,R}-N2	7.B 7.C.1.b 7.C.1.c	Chin EMG tone Chin EMG tone Chin EMG tone, arousal, slow eye movement
	R-{N1-N1-N1,R-R-R}	7.C.1.d	Alpha, body movement, slow eye movement
N2-R	R-R-{N2,R}-N2	7.C.1.e	Sleep spindles
	(N2-)N2-{N2,R}-R(-R)	7.D.1 7.D.2 7.D.3	Chin EMG tone Chin EMG tone, K-complexes, sleep spindles K-complexes, sleep spindles

*Curly braces indicate choice between the stages or stage progressions in the set, and parentheses indicate optional epochs.

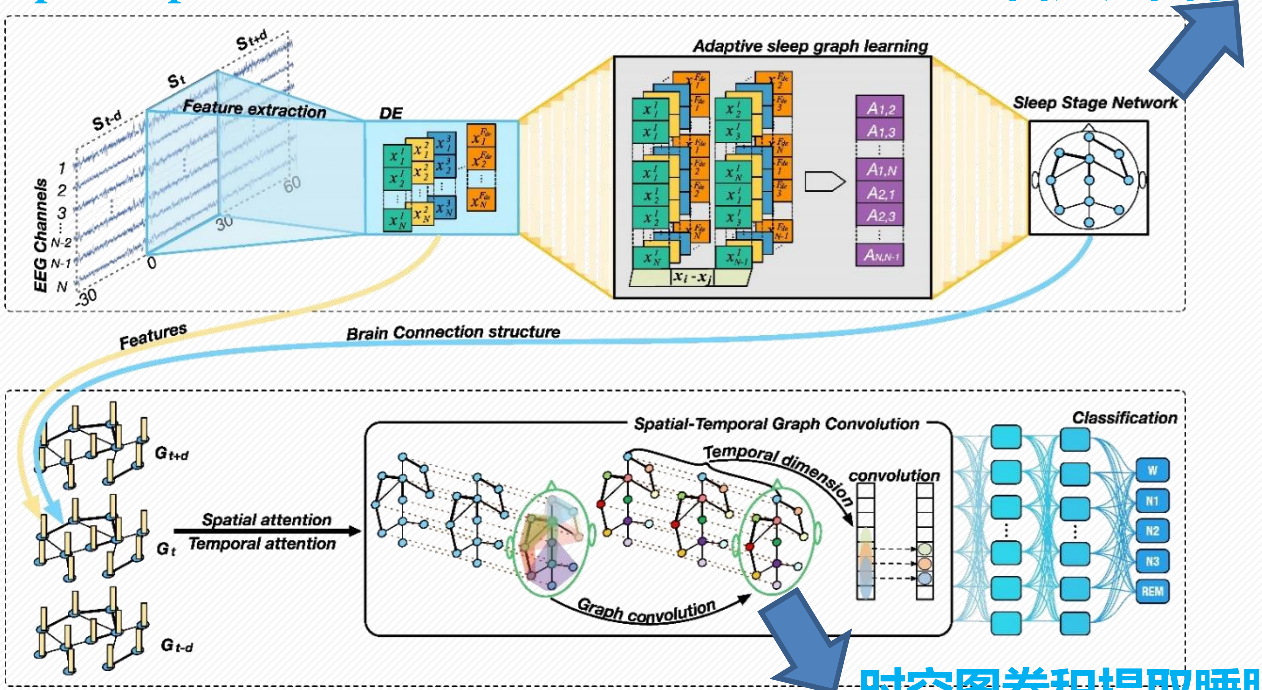
AASM睡眠分期标准中的睡眠阶段过渡规则[6]

- **挑战2.1**：如何将图卷积有效的应用于睡眠分期。
- **挑战2.2**：如何利用相邻睡眠阶段之间的过渡规则。

Methods

GraphSleepNet: 自适应时空图卷积网络

自适应图学习模块，解决网络构建问题



时空图卷积提取睡眠时空特征

贡献：

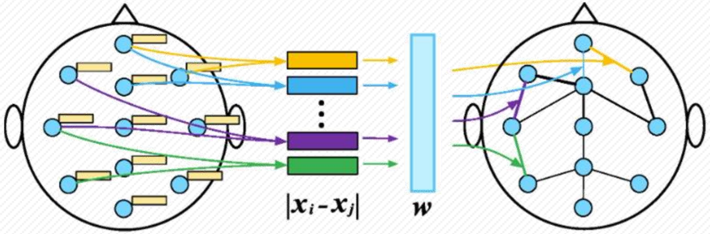
- ◆第一次将时空图卷积用于睡眠分期任务。
- ◆一种新颖的自适应睡眠图学习机制。
- ◆我们设计了一种时空图卷积。
- ◆实现了睡眠分期领域中的最优结果。

Methods

挑战1：如何为睡眠分期确定合适的图结构。

方法1：我们提出了一种自适应睡眠图学习机制.

- 与时空图卷积集成在统一的架构中。
- 动态的构造邻接矩阵A。
- 利用损失函数中的第二项以控制邻接矩阵阵**A的稀疏性**。



$$A_{mn} = g(\boldsymbol{x}_m, \boldsymbol{x}_n) = \frac{\exp(\mathrm{ReLU}(\boldsymbol{w}^T|\boldsymbol{x}_m - \boldsymbol{x}_n|))}{\sum_{n=1}^{N} \exp\left(\mathrm{ReLU}\left(\boldsymbol{w}^T\left|\boldsymbol{x}_m - \boldsymbol{x}_n\right|\right)\right)}$$

$$\mathcal{L}_{\text{graph\_learning}} = \sum_{m,n=1}^{N} \|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2^2 A_{mn} + \lambda\|\boldsymbol{A}\|_F^2$$

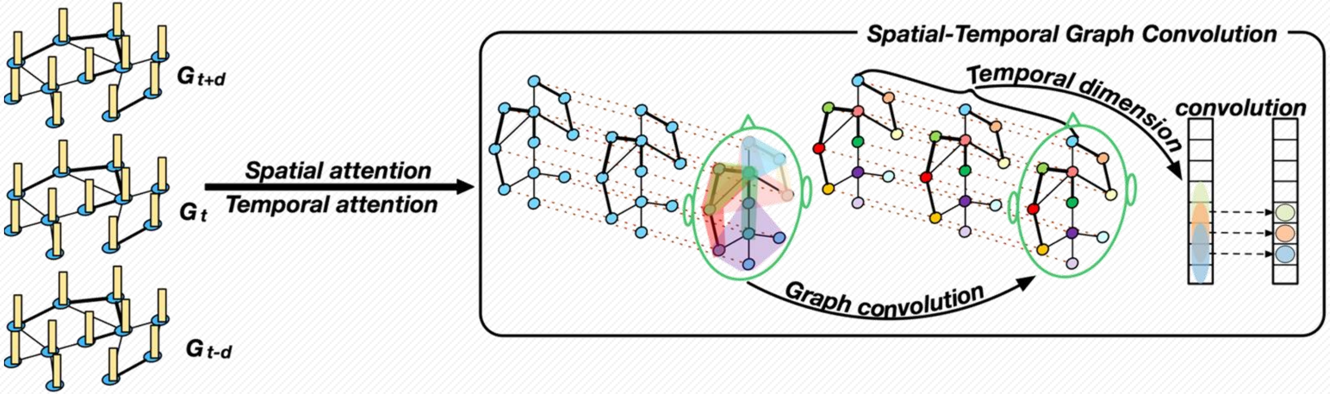
Methods

挑战2：如何有效地提取时空特征。

方法2：我们设计了一种时空图卷积架构。

a) 空间维度：利用图卷积聚合空间信息。

- 使用基于谱图理论的图卷积方法。
- 利用图拉普拉斯算子的切比雪夫展开来降低计算复杂性。



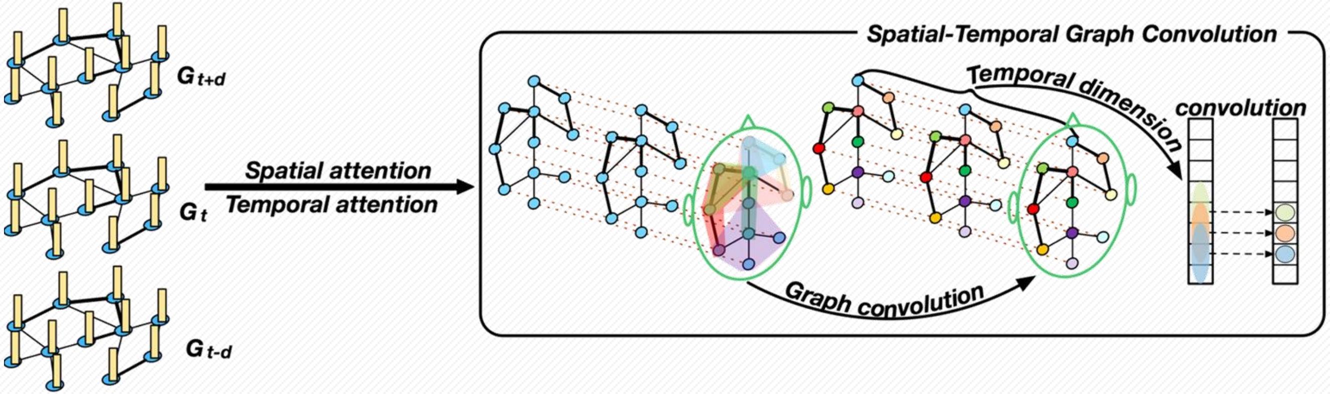
Methods

挑战2：如何有效地提取时空特征。

方法2：我们设计了一种时空图卷积架构。

b) 时间维度：使用卷积神经网络进行卷积操作以提取睡眠阶段间的过渡规则。

c) 时空注意力：自动提取有价值的信息。



Experiments

数据集：

Montreal Archive of Sleep Studies (MASS)-SS3 dataset [7]

- 该数据集包含来自62位健康受试者（28位男性和34位女性）的PSG记录。
- 专家根据AASM标准将这些PSG记录分为五个睡眠阶段（W，N1，N2，N3和REM）。
- 我们从原始信号中的每个通道中提取微分熵（ DE ）特征。

Stage	W	N1	N2	N3	REM	Total
Samples	6357	4829	29777	7651	10566	**59180**
Ratio	10.7%	8.2%	50.3%	12.9%	17.9%	**100%**

Number of samples for each sleep stage

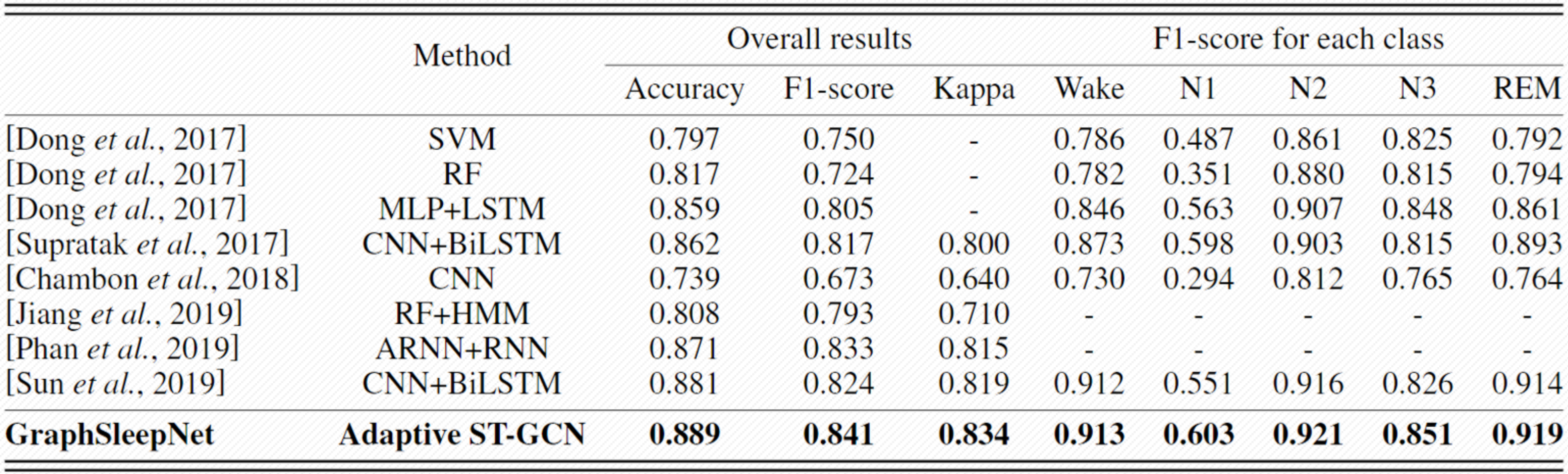
Experiments

基准方法:

- [Dong *et al.*, 2017][8]: 一种混合神经网络，它结合了多层感知机和长短期记忆，此外，我们还比较了其随机森林和支持向量机的性能。
- [Supratak *et al.*, 2017][3]: 一个结合了卷积神经网络和双向长短期记忆来捕捉时间不变量特征和睡眠阶段间的过渡规则的模型。
- [Chambon *et al.*, 2018][9]: 利用多变量和多模态时间序列进行时间睡眠阶段分类。
- [Phan *et al.*, 2019][2]: 通过使用基于注意力的双向递归神经网络和递归神经网络，将单一的睡眠阶段分类问题改变为序列到序列的分类问题。
- [Sun *et al.*, 2019][10]: 一种分别学习综合特征和时间序列的层次神经网络。.
- [Jiang *et al.*, 2019][11]: 使用多模态分解和基于隐马尔科夫模型优化的泛用睡眠阶段分类。

Experiments

与SOTA方法的对比结果：

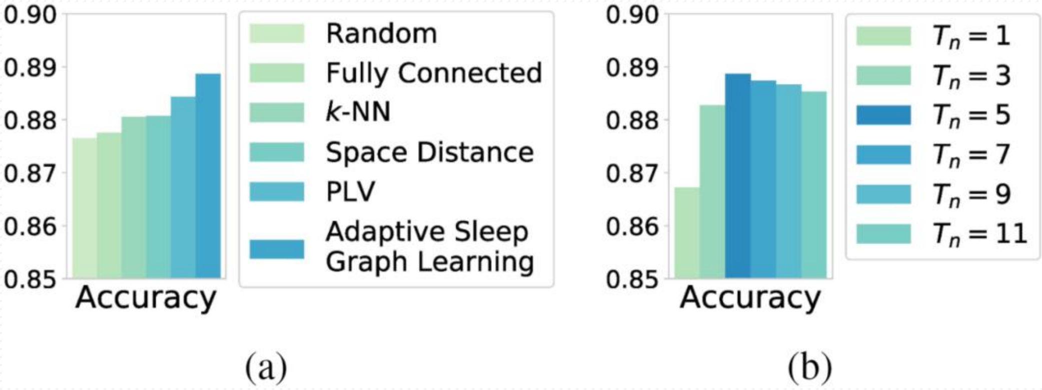


Method		Overall results			F1-score for each class				
		Accuracy	F1-score	Kappa	Wake	N1	N2	N3	REM
[Dong *et al.*, 2017]	SVM	0.797	0.750	-	0.786	0.487	0.861	0.825	0.792
[Dong *et al.*, 2017]	RF	0.817	0.724	-	0.782	0.351	0.880	0.815	0.794
[Dong *et al.*, 2017]	MLP+LSTM	0.859	0.805	-	0.846	0.563	0.907	0.848	0.861
[Supratak *et al.*, 2017]	CNN+BiLSTM	0.862	0.817	0.800	0.873	0.598	0.903	0.815	0.893
[Chambon *et al.*, 2018]	CNN	0.739	0.673	0.640	0.730	0.294	0.812	0.765	0.764
[Jiang *et al.*, 2019]	RF+HMM	0.808	0.793	0.710	-	-	-	-	-
[Phan *et al.*, 2019]	ARNN+RNN	0.871	0.833	0.815	-	-	-	-	-
[Sun *et al.*, 2019]	CNN+BiLSTM	0.881	0.824	0.819	0.912	0.551	0.916	0.826	0.914
GraphSleepNet	**Adaptive ST-GCN**	**0.889**	**0.841**	**0.834**	**0.913**	**0.603**	**0.921**	**0.851**	**0.919**

Table 2: The performance comparison of the state-of-the-art approaches on the MASS dataset

Experiments

实验分析:



- **邻接矩阵**: 我们提出的自适应图学习的结果优于所有固定图的结果。
- **输入睡眠阶段网络的数量** T_n : 分类表现随着 T_n 增加而提高, 并且在 T_n = 5时达到最高的分类精度。

Conclusion

结论：

- 据我们所知，这是首次将**时空图卷积应用到了自动睡眠阶段分类任务**。此外，我们还提出了一种**新型的自适应睡眠图学习机制**，该机制与时空图卷积同时集成在一个统一的网络架构中。
- 我们设计了一种时空卷积，它包括用于捕捉空间特征的图积卷和用于捕捉不同睡眠阶段之间过渡规则的时空卷积。
- 实验结果表明，GraphSleepNet在睡眠阶段分类中实现了**最先进的性能**。

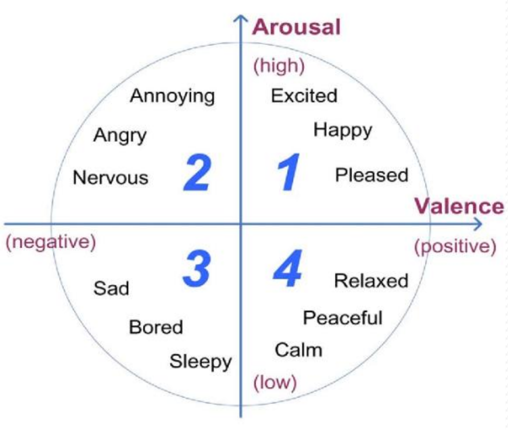
展望：

- 我们所提出的模型是一个**多变量生理时间序列的通用框架**。
- 它还可以应用于**时间序列的分类、预测**和其他相关领域。

SST-EmotionNet: Spatial-Spectral-Temporal based Attention 3D Dense Network for EEG Emotion Recognition

论文链接：https://dl.acm.org/doi/abs/10.1145/3394171.3413724

基于EEG信号的情绪识别



情绪:

情绪与许多精神疾病相关联，如自闭症和抑郁症[1, 2]；
情绪被用作评估患者精神障碍的参考[3]。

基于EEG信号的情绪识别:

与面部表情等情绪识别方法相比，脑电信号可以客观地反映
是一种能够识别真实情绪的可靠方法[4] 。

[1] Al-Kaysi, et al. (2017). Predicting tDCS treatment outcomes of patients with major depressive disorder using automated EEG classification. Journal of affective disorders, 208, 597-603.
[2] Bocharov, et al. (2017). Depression and implicit emotion processing: An EEG study. Neurophysiologie Clinique/Clinical Neurophysiology, 47(3), 225-230.
[3] Zhong, et al. (2020). EEG-Based Emotion Recognition Using Regularized Graph Neural Networks. IEEE Transactions on Affective Computing.
[4] Zheng, et al. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. IEEE Transactions on Autonomous Mental Development, 7(3), 162-175.

相关工作

- **频域特征:**

◆ DE [5, 6], PSD [7, 8], DASM [9], RASM [10], DCAU [4] , etc.

- **时域特征:**

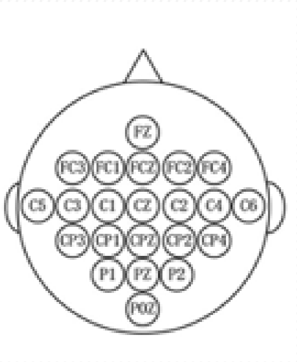
◆ LSTM [11], MMResLSTM [12] , etc.

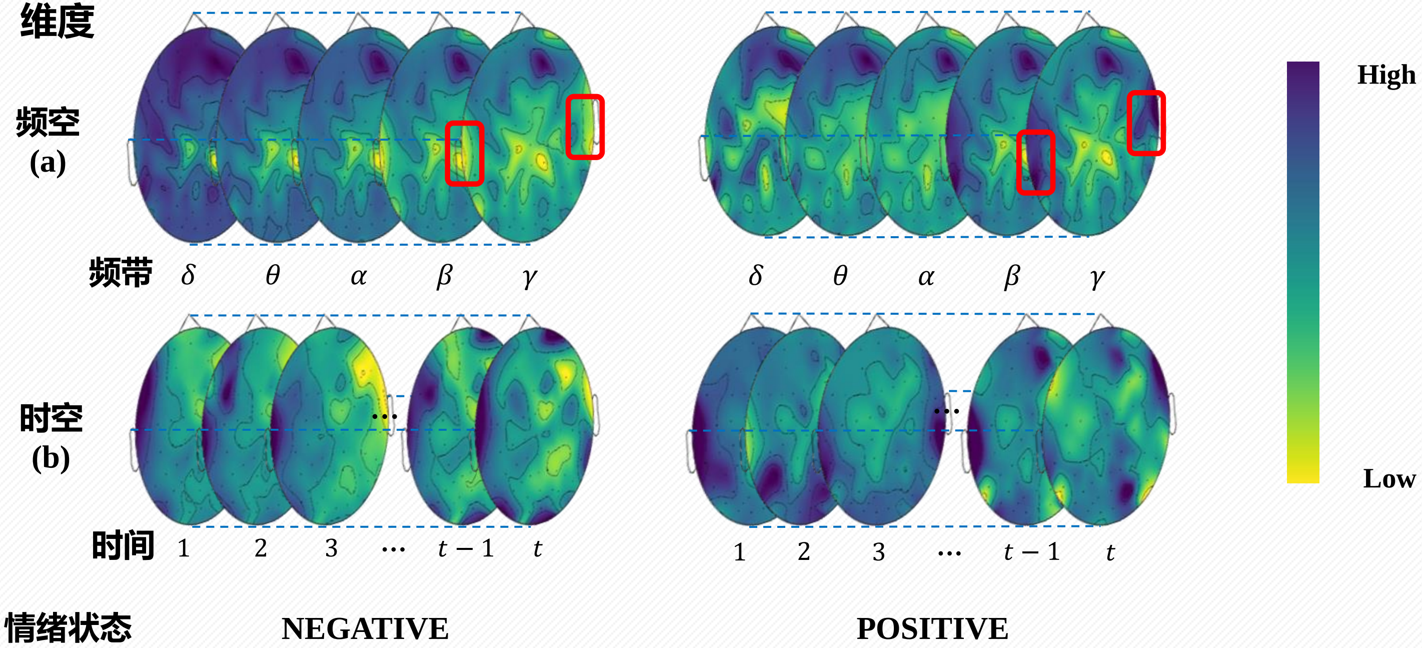
- **空域特征:**

◆ CNN [13, 14], GCN [15, 16] , etc.

◆ 大部分现有的情绪识别方法仅考虑了以上一种或两种特征。

脑电信号在不同情绪状态下的时频空特征激活:

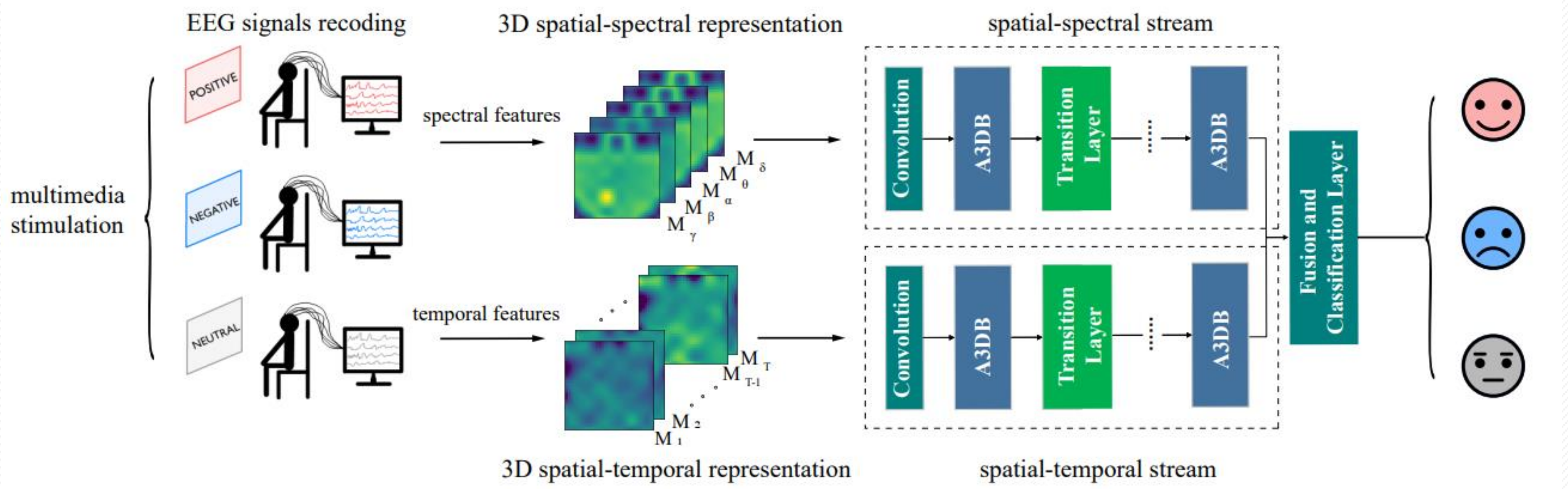




C1: **如何利用被现有方法所忽略的脑电时、频、空特征之间的互补性。**

C2: **如何捕获情绪识别任务中脑电信号的局部时频空特征。**

流程图



情绪的产生

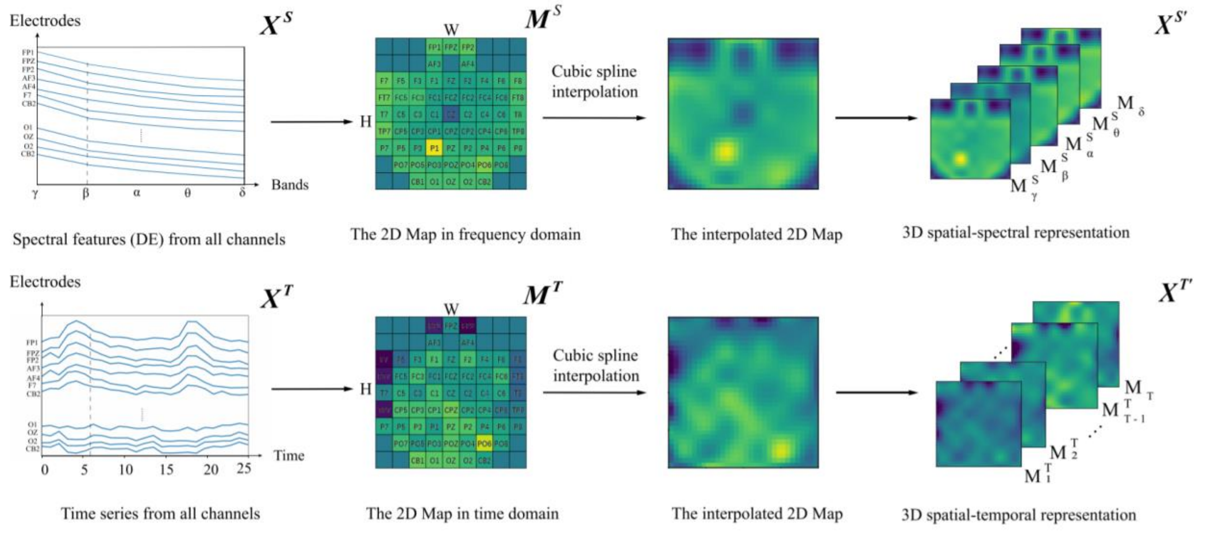
3D表示构建

分类模型

方法

C1: **如何利用不同特征间的互补性?**

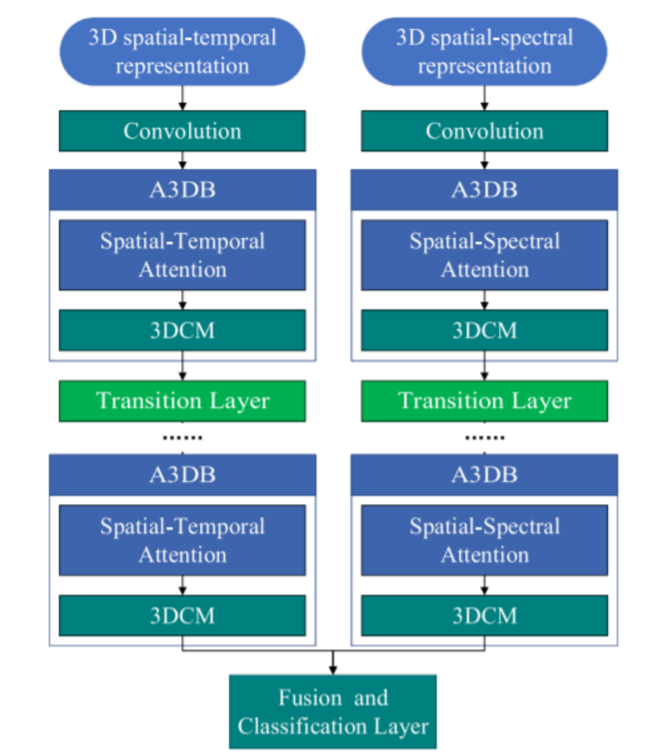
S1.1: **构建脑电信号的3D时频空表示。**



方法

*C1:*如何利用不同特征间的互补性?

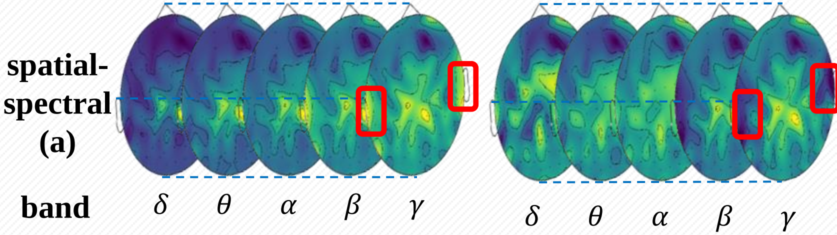
*S1.2:*提出了一个双流的3D密集连接网络，基于已经构建好的脑电3D表示，在一个统一的网络框架下同时融合了脑电的时频空信息。

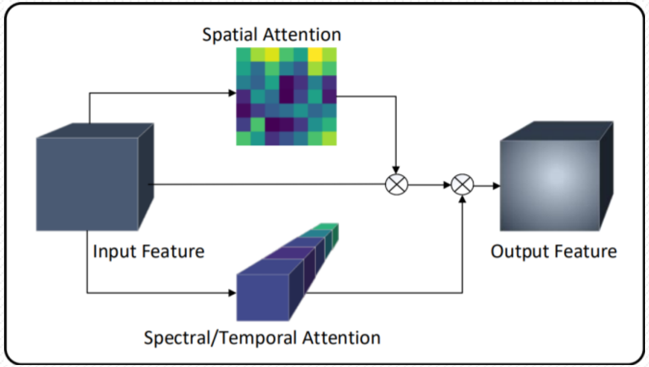

3D spatial-temporal representation
3D spatial-spectral representation
Convolution
Convolution
A3DB
A3DB
Spatial-Temporal Attention
Spatial-Spectral Attention
3DCM
3DCM
Transition Layer
Transition Layer
....
....
A3DB
A3DB
Spatial-Temporal Attention
Spatial-Spectral Attention
3DCM
3DCM
Fusion and Classification Layer

方法

*C2:*如何捕获时频空特征中的局部特征?

*S2:*设计了一种时频空注意力机制，用于动态捕获时、频、空域下对于情绪识别任务有价值的局部特征模式。





实验结果

与SOTA模型做对比:

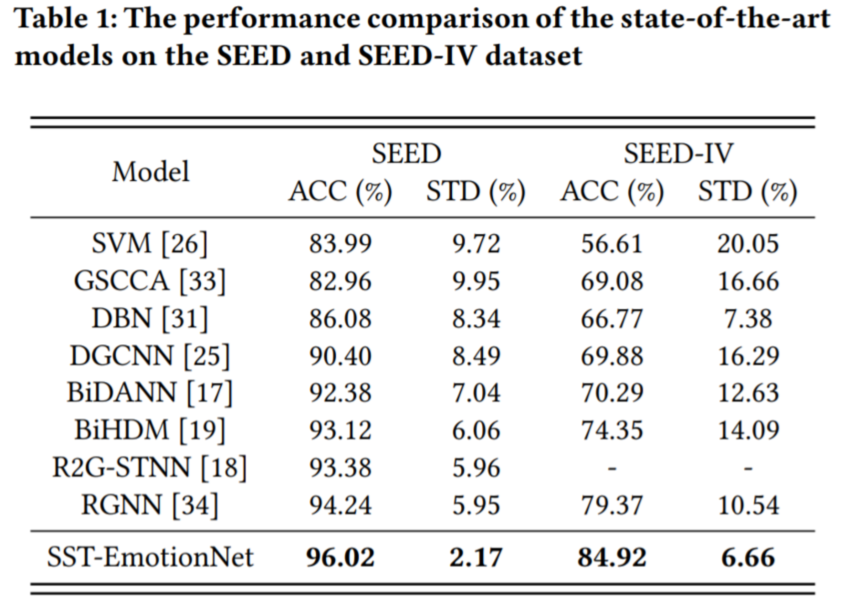


Table 1: The performance comparison of the state-of-the-art models on the SEED and SEED-IV dataset

Model	SEED		SEED-IV	
	ACC (%)	STD (%)	ACC (%)	STD (%)
SVM [26]	83.99	9.72	56.61	20.05
GSCCA [33]	82.96	9.95	69.08	16.66
DBN [31]	86.08	8.34	66.77	7.38
DGCNN [25]	90.40	8.49	69.88	16.29
BiDANN [17]	92.38	7.04	70.29	12.63
BiHDM [19]	93.12	6.06	74.35	14.09
R2G-STNN [18]	93.38	5.96	-	-
RGNN [34]	94.24	5.95	79.37	10.54
SST-EmotionNet	**96.02**	**2.17**	**84.92**	**6.66**

总结

贡献:

- 我们提出了一个双流3D密集网络，它使用脑电信号的3D时频空表示在一个**统一的网络框架下融合脑电信号的时频空特征。**
- 我们提出了**一种时频空注意力机制**，用于动态捕捉时、频、空下有辨别力的局部模式。
- 我们两个数据集上进行了实验，实验结果显示我们的SST-EmotionNet表现优于其他SOTA模型。

References

[1] Junming Zhang and Y an Wu. A new method for automatic sleep stage classification. *IEEE transactions on biomedical circuits and systems*, 11(5):1097–1110, 2017.

[2] Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y Chén, and Maarten De Vos. Seqsleepnet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410, 2019.

[3] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.

[4] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[5] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[6] Iber, C., S. Ancoli-Israel, A. Chesson, and S. F. Quan. *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. Westchester, IL:American Academy of Sleep Medicine, 2007.

[7] Christian O'Reilly, Nadia Gosselin, Julie Carrier, and Tore Nielsen. Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of sleep research*, 23(6):628–635, 2014.

[8] Hao Dong, Akara Supratak, Wei Pan, Chao Wu, Paul M Matthews, and Yike Guo. Mixed neural network approach for temporal sleep stage classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):324–333, 2017.

[9] Stanislas Chambon, Mathieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018.

[10] Chenglu Sun, Chen Chen, Wei Li, Jiahao Fan, and Wei Chen. A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning. *IEEE journal of biomedical and health informatics*, 2019.

[11] Dihong Jiang, Y a-nan Lu, MA Y u, and W ANG Y uanyuan. Robust sleep stage classification with single-channel eeg signals using multimodal decomposition and hmm-based refinement. *Expert Systems with Applications*, 121:188–203, 2019.

相关资源

Selected Paper

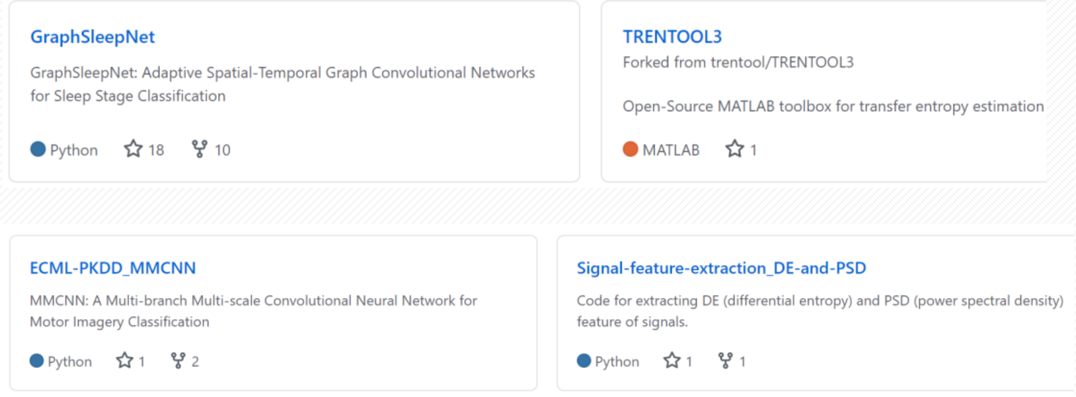
- **MMCNN: A Multi-branch Multi-scale Convolutional Neural Network for Motor Imagery Classification (ECML-PKDD 2020, CCF B, Oral)**
- **Refined nonuniform embedding for coupling detection in multivariate time series. Physical Review E 101 (2020) 062113.(SCI-II)**
- **Detecting Causality in Multivariate Time Series via Non-Uniform Embedding. Entropy 21(12) (2019): 1233. (SCI-III)**
- **Sleep Stage Classification Model Based on Deep Convolutional Neural Network. Journal of Zhejiang University (Engineering Science). (Chinese Journal EI)**

更多论文详见个人主页：https://ziyujia.github.io/

相关资源

■ 论文、数据集、代码

- ■ GraphSleepNet模型：https://github.com/ziyujia/GraphSleepNet
- ■ MMCNN模型：https://github.com/ziyujia/ECML-PKDD_MMCNN
- ■ SST-EmotionNet模型：正在整理，后续会公开



GraphSleepNet

GraphSleepNet: Adaptive Spatial-Temporal Graph Convolutional Networks for Sleep Stage Classification

● Python ☆ 18 ⑂ 10

TRENTOOL3

Forked from trentool/TRENTOOL3

Open-Source MATLAB toolbox for transfer entropy estimation

● MATLAB ☆ 1

ECML-PKDD_MMCNN

MMCNN: A Multi-branch Multi-scale Convolutional Neural Network for Motor Imagery Classification

● Python ☆ 1 ⑂ 2

Signal-feature-extraction_DE-and-PSD

Code for extracting DE (differential entropy) and PSD (power spectral density) feature of signals.

● Python ☆ 1 ⑂ 1

邮箱：ziyujia@bjtu.edu.cn

微信：

谢谢！

KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction

Xuan Lin[1], Zhe Quan[1], Zhi-Jie Wang[2], Tengfei Ma[1]
and Xiangxiang Zeng[1]

[1]Hunan University[2]Chongqing University

Outline

□ Background and Motivation

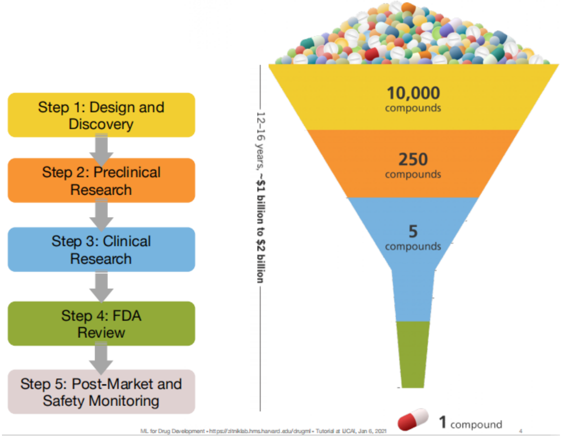
□ Knowledge Graph Neural Network

□ Experimental Results

□ Conclusion and Future Work

Background

- Drug Discovery[1]


Step 1: Design and Discovery
Step 2: Preclinical Research
Step 3: Clinical Research
Step 4: FDA Review
Step 5: Post-Market and Safety Monitoring
12–16 years, ~$1 billion to $2 billion
10,000 compounds
250 compounds
5 compounds
1 compound

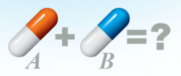
ML for Drug Development • https://zitniklab.hms.harvard.edu/drugml • Tutorial at IJCAI, Jan 6, 2021 4

[1] M. Zitnik, et al. Machine Learning for Drug Development. *IJCAI 20'*

Background

- Drug-Drug Interaction (DDI)

  Drug-drug interactions occur when two or more drugs react with each other. This DDI may cause you to experience an unexpected side effect.



- Why Need Drug Drug Interaction Prediction?

  For example, mixing a drug you take to help you sleep (*a sedative*) and a drug you take for allergies (*an antihistamine*) can slow your reactions and make driving a car or operating machinery dangerous.

Motivation

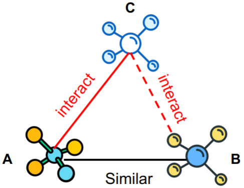
▶ Limitation of Previous Methods

- Molecule representation

  Intuition: drugs with similar representations will perform similar DDIs

  Goal: learn better drug similarity based on multi-view drug features[2]

  Limitation: design specialized drug representation

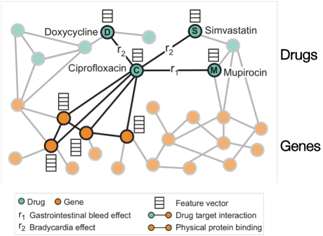


- Network embedding-based methods

  Intuition: drug combination leads to polypharmacy side effect

  Goal: predict labeled edges between drugs[3]

  Limitation: single relation



[2] T. Ma, et al. Drug Similarity Integration Through Attentive Multi-view Graph Auto-Encoders. *IJCAI 18'*

[3] M. Zitnik, et al. Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *Bioinformatics 18'*

Motivation

- Our Solution: KGNN

- Knowledge graph

  provide abundant information

  - structural relations among multiple entities
  - semantic relations associated with each node

- Graph neural network

  recursively learn from neighboring information

  - neighborhood sampling
  - aggregation

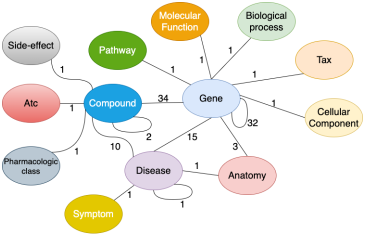
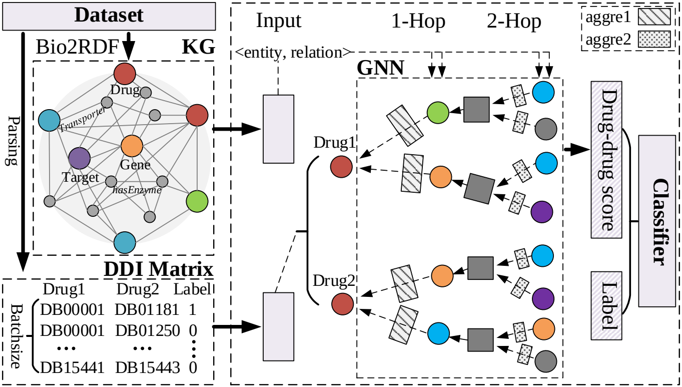


Figure: Interactions in the DRKG[4]

[4] https://github.com/gnn4dr/DRKG

Knowledge Graph Neural Network

- Framework



Drug1	Drug2	Label
DB00001	DB01181	1
DB00001	DB01250	0
⋯	⋯	⋮
DB15441	DB15443	0

- DDI extraction and KG construction
- KGNN layer
- Drug-drug interaction prediction

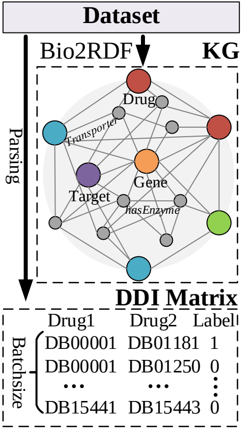
Knowledge Graph Neural Network

- DDI Extraction and KG Construction

  - Download and parse the dataset
  - Bio2RDF[5]

Table: The detailed description of KG.

	DrugBank	KEGG-drug
Drugs	2,578	1,925
Interactions	612,388	56,983
Entities	2,129,712	129,910
Relation Types	72	167
KG Triples	7,852,852	362,870



Drug1	Drug2	Label
DB00001	DB01181	1
DB00001	DB01250	0
···	···	⋮
DB15441	DB15443	0

[5] Bio2RDF: https://github.com/bio2rdf/bio2rdf-scripts/wiki

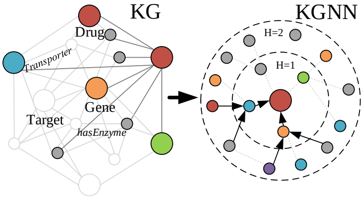
Knowledge Graph Neural Network

- KGNN Layer

- Local receptive: 2-hop

- Neighborhood sampling

  $N_{neigh}(e)$: the entity connects directly to a drug
  $S(e)$: a fixed size set, $S(e) < N_{neigh}(e)$



- Aggregation

  aggregate the entity e and its

  neighborhood representation $e^{i}_{S(e)}$

$$aggre_{sum} = \sigma(W \cdot (e + e^{i}_{S(e)}) + b)$$

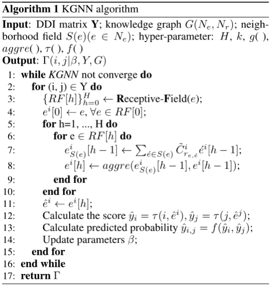
$$aggre_{concat} = \sigma(W \cdot concat(e, e^{i}_{S(e)}) + b)$$

$$aggre_{neighbor} = \sigma(W \cdot e^{i}_{S(e)} + b)$$

Knowledge Graph Neural Network

- Drug-Drug Interaction Prediction

  - Extract the DDI data sources and construct the corresponding KG
  - Obtain the features of drug and its neighboring of related entities
  - Concatenate all the representations and feed them to output the interaction value



Algorithm 1 KGNN algorithm

Input: DDI matrix **Y**; knowledge graph $G(N_e, N_r)$; neighborhood field $S(e)(e \in N_e)$; hyper-parameter: H, k, $g(\,)$, $aggre(\,)$, $\tau(\,)$, $f(\,)$

Output: $\Gamma(i, j|\beta, Y, G)$

```
while KGNN not converge do
    for (i, j) ∈ Y do
        $\{RF[h]\}_{h=0}^{H} \leftarrow$ Receptive-Field($e$);
        $e^i[0] \leftarrow e, \forall e \in RF[0]$;
        for h=1, ..., H do
            for e ∈ $RF[h]$ do
                $e^i_{S(e)}[h-1] \leftarrow \sum_{\acute{e} \in S(e)} \tilde{C}^i_{r_{e,\acute{e}}} \acute{e}^i[h-1]$;
                $e^i[h] \leftarrow aggre(e^i_{S(e)}[h-1], e^i[h-1])$;
            end for
        end for
        $\hat{e}^i \leftarrow e^i[h]$;
        Calculate the score $\hat{y}_i = \tau(i, \hat{e}^i), \hat{y}_j = \tau(j, \hat{e}^j)$;
        Calculate predicted probability $\hat{y}_{i,j} = f(\hat{y}_i, \hat{y}_j)$;
        Update parameters $\beta$;
    end for
end while
return Γ
```

Figure: KGNN algorithm

- Loss Function

$$Loss = \sum_{(i,j) \in Y (i,j \in N_d, j \neq i)} -y_{i,j} log y_{i,j} - (1 - y_{i,j}) log(1 - \hat{y}_{i,j})$$

Experimental Results

- Dataset
  - DrugBank (V5.1.4)[6]
    obtains 2,578 approved small molecule drugs and 612,388 unique approved DDIs spanning 13,339 drugs.
  - KEGG-drug[7]
    collects 1,925 approved drugs and 56,983 approved interactions spanning 11,147 drugs and 324,183 interactions respectively.

  randomly divide *all approved DDIs* as positive samples into training, validation and testing sets in a 8/1/1 ratio
- Metrics
  - ACC, AUPR, AUC-ROC, F1 scores
- Baselines
  - Matrix Factorization (MF), Random Walk (RW), Neural Network (NN), Deep Learning (DL), Knowledge Graph (KG)

[6] DrugBank: https://go.drugbank.com/releases/latest

[7] KEGG-drug: https://www.kegg.jp/kegg/drug/

Experimental Results

- Results and Analysis
  compare the performance of the proposed method with the baselines

- Ablation Study
  test the performance of multiple aggregation operations

Methods / Metrics	MF-based		RW-based		NN-based			DL-based	KG-based	**KGNN$_x$**		
	Laplacian	GreRep	DeepWalk	struc2vec	LINE	SDNE	GAE	DeepDDI	KG-ddi	*neighbor*	*sum*	*concat*
ACC	0.7183	0.8443	0.8349	0.7882	0.8280	0.8303	0.7491	0.8123	0.7867	0.9354	0.9538	**0.9561**
	0.8029	0.8718	0.8547	0.8436	0.8655	0.8674	0.7586	0.8229	0.8154	0.8846	0.8882	**0.8950**
AUPR	0.7533	0.9115	0.9070	0.8672	0.8915	0.8782	0.7403	0.9193	–	0.9801	0.9890	**0.9892**
	0.8261	0.9055	0.9011	0.8861	0.8968	0.8967	0.7571	0.8442	–	0.9207	0.9247	**0.9533**
AUC-ROC	0.7966	0.9230	0.9181	0.8735	0.9092	0.9029	0.8085	0.9261	0.7867	0.9824	0.9902	**0.9912**
	0.8736	0.9305	0.9208	0.9086	0.9264	0.9249	0.8334	0.8994	0.8154	0.9418	0.9453	**0.9518**
F1	0.7270	0.8461	0.8357	0.7962	0.8318	0.8373	0.7889	0.8466	0.7843	0.9366	0.9544	**0.9566**
	0.8079	0.8748	0.8570	0.8476	0.8695	0.8704	0.7888	0.7966	0.8152	0.8869	0.8909	**0.8982**

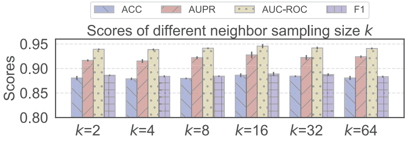
Table: Performance of KGNN against comparative approaches. First/second row of each method corresponds to results reported on **DrugBank** and **KEGG-drug** dataset respectively.

Experimental Results

- Impact of Key Parameters

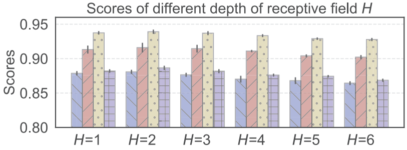
  - Neighborhood size k

    $k=16$, achieves the best performance



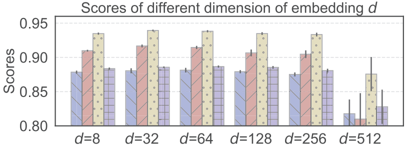
  - Depth of receptive field H

    $H = 3$, performance decreases



  - Dimension of embedding d

    d =32 or 64, boost the performance



Conclusion and Future Work

- Conclusion
  - An novel framework for drug-drug interaction prediction.
  - Extends spatial-based GNN methods to the knowledge graph.
  - Provides new insights into the study of jointly considering
    - topological structure information of drug
    - semantic relation of knowledge graph
- Future Work
  - Large-scale knowledge graph
  - Neighborhood sampling
  - Multi-typed DDI prediction

Paper: https://www.ijcai.org/Proceedings/2020/0380.pdf
Code: https://github.com/jacklin18/KGNN
About me: https://jacklin18.github.io/

DrugAI

IJCAI-PRICAI
YOKOHAMA
2021

International Joint Conference
on Artificial Intelligence - Pacific
Rim International Conference
on Artificial Intelligence.
January 7-15, 2021.
Yokohama, Japan.

IGNITE: A Minimax Game Toward Learning Individual Treatment Effects from Networked Observational Data

Ruocheng Guo[1], Jundong Li[2], Yichuan Li[3], K. S. Candan[1], Adrienne Raglin[4], Huan Liu[1]

1 Arizona State University
2 University of Virginia
3 Worcester Polytechnic Institute
4 Army Research Laboratory

Introduction

What is causality?

A definition with random variables

- Given two random variables T and Y, we say T causes Y iff changing the value of T would cause a change in the value of Y with the values of all the other variables fixed.

Introduction

Why do we care about causal effects?

- They are crucial for decision making
  - A/B tests in tech companies
  - Clinical trials for medicines

Why do we study networked observational data?

- Network information is ubiquitous.
  - Social networks
  - Branch networks of banks
- Network information can be useful, but how?

Introduction

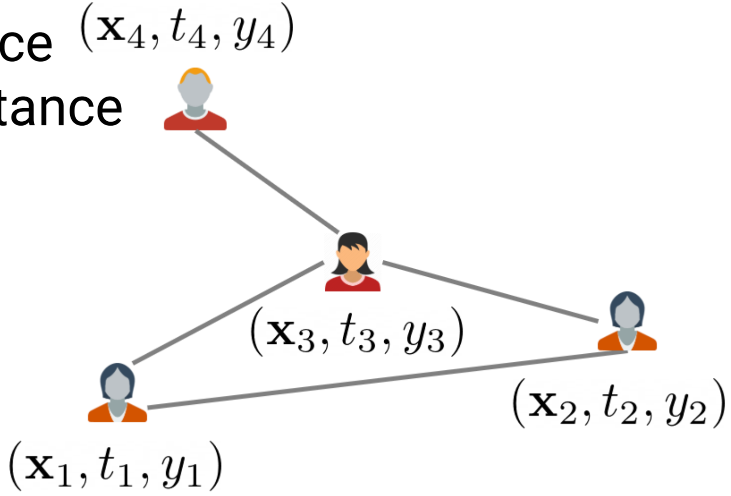
Networked observational data $(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^N, \mathbf{A})$

$\mathbf{x}_i$ - feature vector of an instance

t_i - binary observed treatment of an instance

y_i - an observed factual outcome of an instance

$\mathbf{A}$ - network information


$(\mathbf{x}_4, t_4, y_4)$
$(\mathbf{x}_3, t_3, y_3)$
$(\mathbf{x}_2, t_2, y_2)$
$(\mathbf{x}_1, t_1, y_1)$

$t = 1 :$ take medicine

$t = 0 :$ take no medicine

$y = 1 :$ good health outcome

$y = 0 :$ bad health outcome

Challenge and Motivation

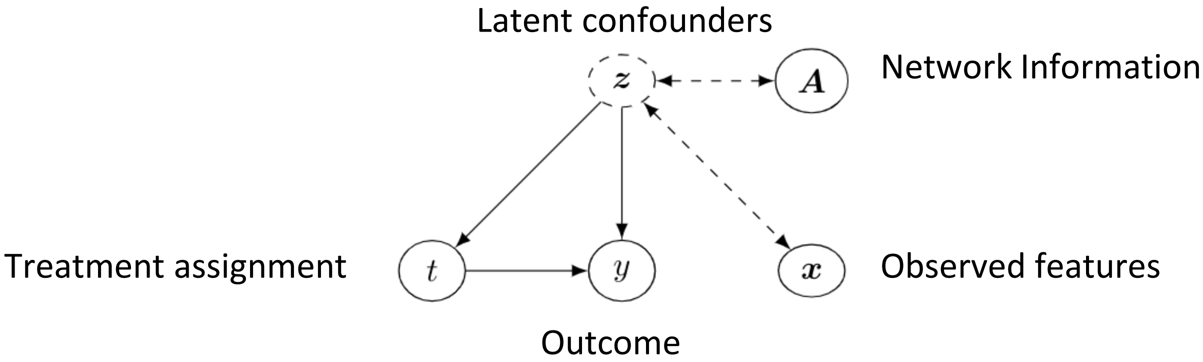
- Challenge: Hidden confounders
  - Without controlling hidden confounders -> biased estimates
- Motivation: two heuristics in existing work
  - Balancing the representation of confounders [1]
  - Predicting the treatment assignments [2]
  - Can we benefit from properly combining them?

[1] Shalit, Uri, Fredrik D. Johansson, and David Sontag. "Estimating individual treatment effect: generalization bounds and algorithms." In International Conference on Machine Learning, pp. 3076-3085. PMLR, 2017.
[2] Veitch, Victor, Yixin Wang, and David Blei. "Using embeddings to correct for unobserved confounding in networks." In *Advances in Neural Information Processing Systems*, pp. 13792-13802. 2019.

Identification

Identification: write causal estimands as probability quantities.

In this work, we follow the theory of measurement bias [1] to identify causal effects based on the following causal graph.



Latent confounders

Network Information

Treatment assignment

Observed features

Outcome

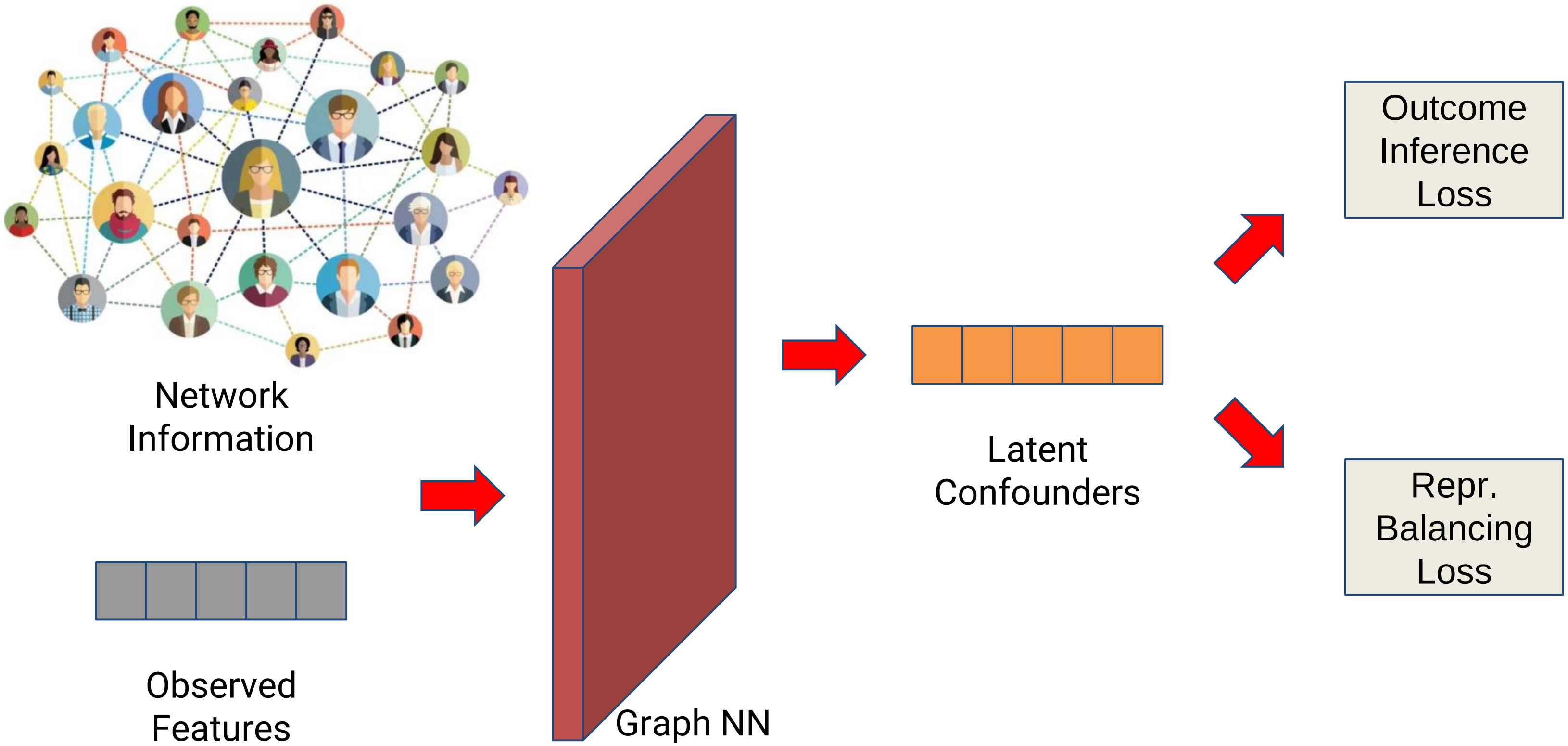
Specifically, with the conditional independence $y^1, y^0 \perp t | \mathbf{z}$, we can identify the causal effect using

$$E[y^1 - y^0|z] = E[y^1|z] - E[y^0|z] = E[y^1|z, t=1] - E[y^0|z, t=0] = E[y|z, t=1] - E[y|z, t=0$$

[1] Kuroki, Manabu, and Judea Pearl. "Measurement bias and effect restoration in causal inference." *Biometrika* 101, no. 2 (2014): 423-437.

IGNITE

A Minimax game for learning latent confounders: overview



Network Information

Observed Features

Graph NN

Latent Confounders

Outcome Inference Loss

Repr. Balancing Loss

IGNITE

Critic based representation balancing

$$\min_{g}\max_{D} \mathcal{L}_{CB} = \frac{1}{n^1}\sum_{i:t_i=1} D(\hat{\mathbf{h}}_i) - \frac{1}{n^0}\sum_{i:t_i=0} D(\hat{\mathbf{h}}_i)$$

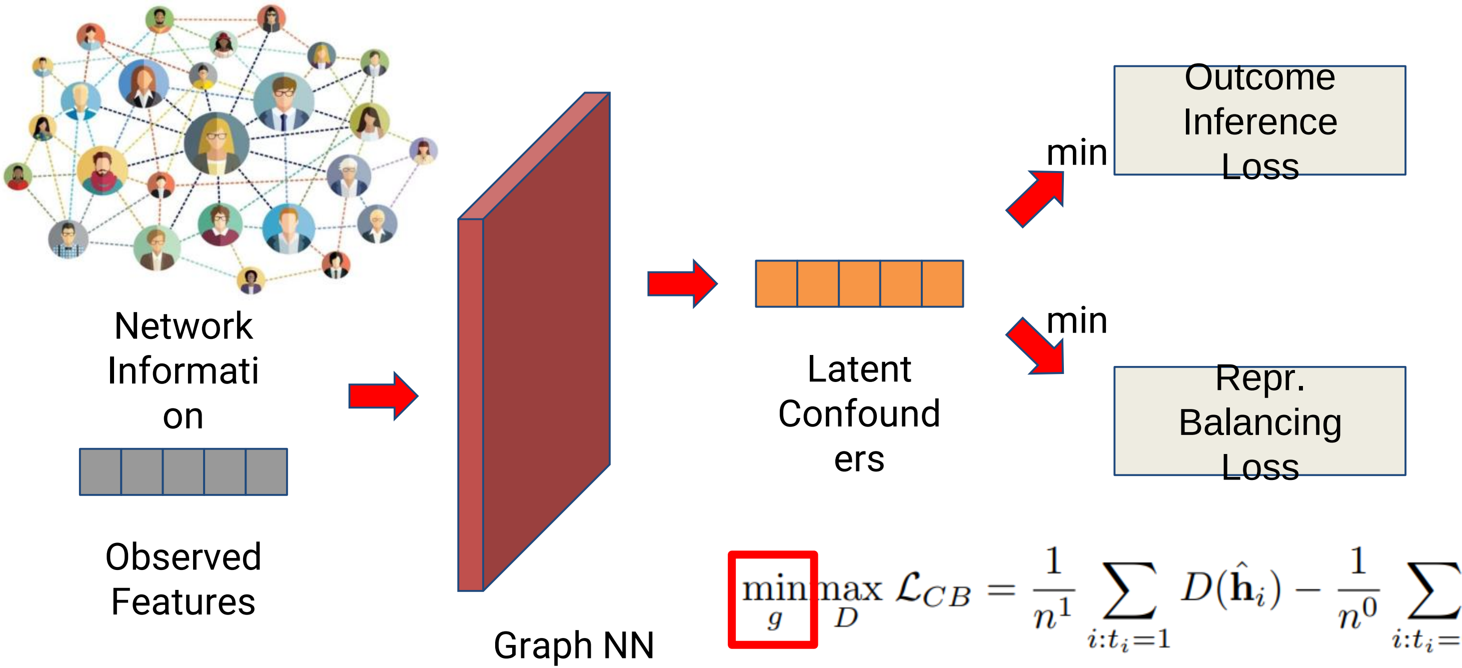
Gradient penalty [1]

$$\mathcal{L}_{GP} = -\frac{1}{n'}\sum_{i=1}^{n'} \lambda(|| \nabla_{\tilde{\mathbf{h}}_i} D(\tilde{\mathbf{h}}_i)||_2 - 1)^2$$

[1] Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. "Improved training of wasserstein gans." In *Advances in neural information processing systems*, pp. 5767-5777. 2017.

IGNITE

A Minimax game for learning latent confounders: min step



Network Information

Observed Features

Graph NN

Latent Confounders

min

Outcome Inference Loss

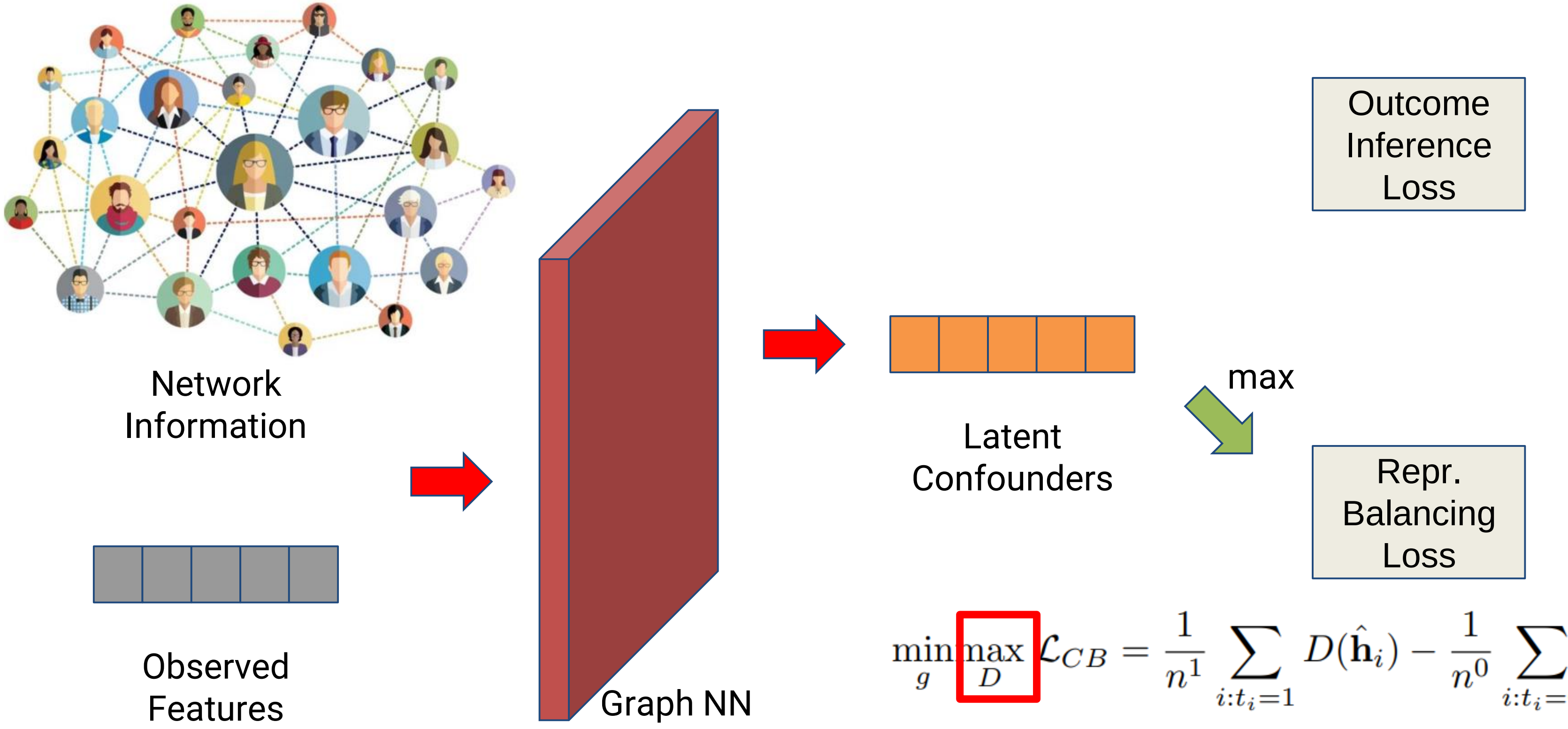
min

Repr. Balancing Loss

$$\boxed{\min_g} \max_D \mathcal{L}_{CB} = \frac{1}{n^1}\sum_{i:t_i=1} D(\hat{\mathbf{h}}_i) - \frac{1}{n^0}\sum_{i:t_i=0} D(\hat{\mathbf{h}}_i)$$

IGNITE

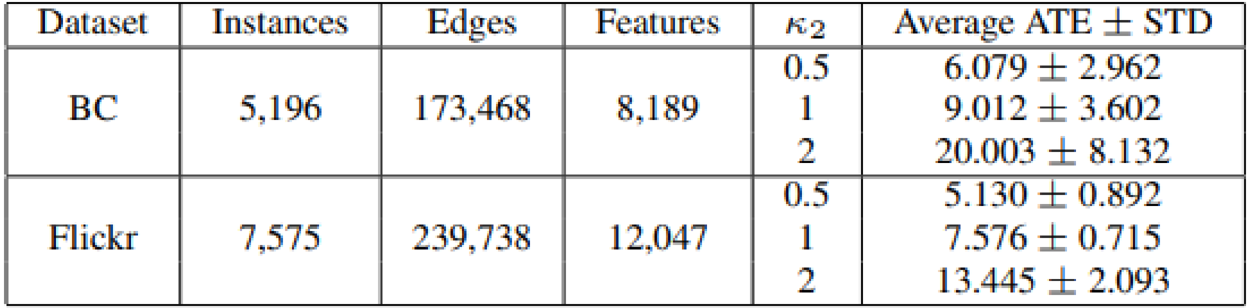
A Minimax game for learning latent confounders: max step



$$\min_{g} \boxed{\max_{D}} \mathcal{L}_{CB} = \frac{1}{n^1} \sum_{i:t_i=1} D(\hat{\mathbf{h}}_i) - \frac{1}{n^0} \sum_{i:t_i=0} D(\hat{\mathbf{h}}_i).$$

Experimental Setup

- Semi-synthetic datasets
  - We obtain features and network information from real-world datasets.
  - We synthesize treatments and outcomes similar to the News dataset in [1].
  - We consider various strength of hidden confounding controlled by parameter κ_2
  - Different from [2], we randomly sample edge weights to reflect real-world cases.



Dataset	Instances	Edges	Features	κ_2	Average ATE $\pm$ STD
BC	5,196	173,468	8,189	0.5	6.079 ± 2.962
				1	9.012 ± 3.602
				2	20.003 ± 8.132
Flickr	7,575	239,738	12,047	0.5	5.130 ± 0.892
				1	7.576 ± 0.715
				2	13.445 ± 2.093

Table 1: Statistics of the Datasets

Training/validation/test = 60% : 20% : 20%

[1] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." ICML. 2016.

[2] Guo, Ruocheng, Jundong Li, and Huan Liu. "Learning individual causal effects from networked observational data." In Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 232-240. 2020.

Experimental Setup

- Baselines
  - SOTA neural network based and ensemble based causal inference methods
    - Ablation models: GATD+, GATD and GATDT
    - Network Deconfounder [3] and Causal Network Embedding [5]
    - CFRNet [1], CEVAE [2], and Causal Forest [4]
- Evaluation:
  - compare the estimated causal effects with the ground truth
  - robustness under various hidden confounding strength
- Metrics:
  - RMSE on estimated ITEs

$$\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{n}\sum_{i=1}(\hat{\tau}_i - \tau_i)^2},$$

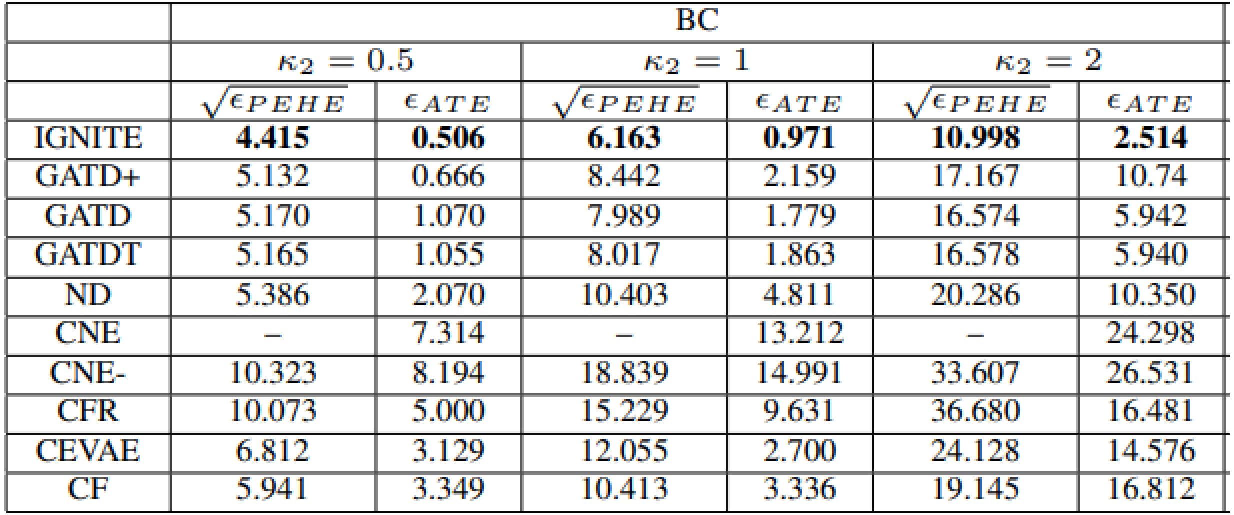
  - MAE on the estimated average treatment effect (ATE)

$$\epsilon_{ATE} = |\frac{1}{n}\sum_{i=1}(\hat{\tau}_i) - \frac{1}{n}\sum_{i=1}(\tau_i)|,$$

[1] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." ICML. 2016.
[2] Louizos, Christos, et al. "Causal effect inference with deep latent-variable models." In NeurIPS, 2017.
[3] Guo, Ruocheng, Jundong Li, and Huan Liu. "Learning individual causal effects from networked observational data." In Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 232-240. 2020.
[4] Wager, Stefan, and Susan Athey. "Estimation and inference of heterogeneous treatment effects using random forests." *JASA.* 2018
[5] Veitch, Victor, Yixin Wang, and David Blei. "Using embeddings to correct for unobserved confounding in networks." In Advances in Neural Information Processing Systems, pp. 13792-13802. 2019.

Results

- IGNITE outperforms the ablation models and the state-of-the-art methods as it combines the benefit of the two heuristics.
- The error of IGNITE increases the least as the influence of hidden confounding (κ_2) increases.



	BC					
	$\kappa_2 = 0.5$		$\kappa_2 = 1$		$\kappa_2 = 2$	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
IGNITE	**4.415**	**0.506**	**6.163**	**0.971**	**10.998**	**2.514**
GATD+	5.132	0.666	8.442	2.159	17.167	10.74
GATD	5.170	1.070	7.989	1.779	16.574	5.942
GATDT	5.165	1.055	8.017	1.863	16.578	5.940
ND	5.386	2.070	10.403	4.811	20.286	10.350
CNE	–	7.314	–	13.212	–	24.298
CNE-	10.323	8.194	18.839	14.991	33.607	26.531
CFR	10.073	5.000	15.229	9.631	36.680	16.481
CEVAE	6.812	3.129	12.055	2.700	24.128	14.576
CF	5.941	3.349	10.413	3.336	19.145	16.812

More to find

The paper can be found at https://www.ijcai.org/Proceedings/2020/0625.pdf

You can also reach out to me through rguo12@asu.edu

Code is available upon request for now and will be released soon.

The Challenge

With observational data, what can we estimate?
Probabilistic quantities: joint, conditional and marginal distributions of observed variables.

Causal effect

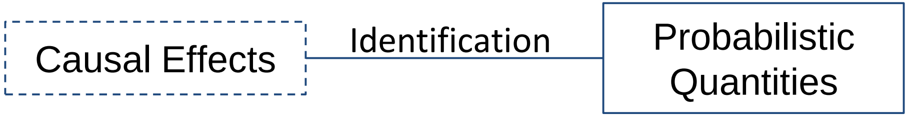
- In potential outcome framework
  - Potential outcomes $y_i^t, t \in \{0, 1\}$
  - Individual treatment effect (ITE) $\tau_i = y_i^1 - y_i^0$
  - Conditional average treatment effect (CATE) $E[\tau|\mathbf{x}]$
  - Average treatment effect $E[\tau]$
  - Not directly estimable from data

How can **network information** help connect probabilistic quantities to causal effects?

Causal Identification

Causal Identification

- With causal assumptions, we can identify causal effects by writing them as functions of probabilistic quantities.



Causal Effects

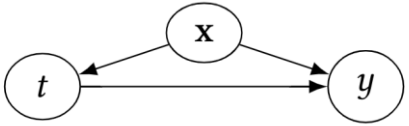
Identification

Probabilistic Quantities

Causal Identification

Strong ignorability

- It assumes that
  - all the **confounders** have been measured as the observed features **x**,
  - each instance's probability to receive treatment (true propensity score) is between 0 and 1.
- In a causal graph



- In the potential outcome framework $y^1, y^0 \perp t|\mathbf{x}$
- How it works in identifying CATE/ITE

$$
\begin{aligned}
E[\tau|\mathbf{x}] &= E[y^1 - y^0|\mathbf{x}] \\
&= E[y^1|\mathbf{x}] - E[y^0|\mathbf{x}] \\
&= [y^1|\mathbf{x}, t = 1] - [y^0|\mathbf{x}, t = 0] \\
&= \boxed{E[y|\mathbf{x}, t = 1]} - \boxed{E[y|\mathbf{x}, t = 0]},
\end{aligned}
$$

Can be estimated!

$$
\begin{aligned}
&= E[y^1 - y^0|\mathbf{x}] \\
&= E[y^1|\mathbf{x}] - E[y^0|\mathbf{x}] \\
&= [y^1|\mathbf{x}, t = 1] - [y^0|\mathbf{x}, t = 0] \\
&= E[y|\mathbf{x}, t = 1] - E[y|\mathbf{x}, t = 0],
\end{aligned}
$$

Causal Identification

Strong ignorability can be untenable given observational data

- There can exist hidden confounders (e.g., socio-economic status)
- Using strong ignorability can lead to **confounding bias**.

Relaxed strong ignorability assumption with latent confounders **z**

$$y^1, y^0 \perp t | \mathbf{z}$$

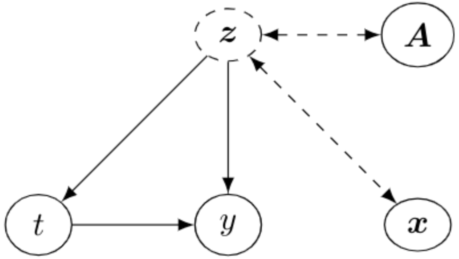
- Latent confounders **z** are not observable, we only assume its existence.
- We can learn **z** from data via machine learning models.

Causal Identification

We propose to use **network information** and observed features to improve the learned latent confounders.

- Network information can compensate for hidden confounders.
- *Homophily*: similar individuals are more likely to connect with each other.

The causal graph



Latent confounders

Network Information

Treatment assignment

Observed features

Outcome

Two Problems

- Learning individual treatment effects with networked observational data [1]

- Counterfactual evaluation of treatment assignment functions with networked observational data [2]

[1] Guo, Ruocheng, Jundong Li, and Huan Liu. "Learning Individual Causal Effects from Networked Observational Data." WSDM 2020.
[2] Guo, Ruocheng, Jundong Li, and Huan Liu. "Counterfactual Evaluation of Treatment Assignment Functions with Networked Observational Data." SDM 2020.

Data Mining &
Machine Learning

Learning Individual Causal Effects with Networked Observational Data

Problem Definition

Problem: learning ITEs with networked observational data

Given: networked observational data $(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^{N}, \mathbf{A})$

Find: ITE $\hat{\tau}_i = \hat{y}_i^1 - \hat{y}_i^0$ of an instance given its features and the network information.

Existing Methods

Neural network based methods that relies on strong ignorability

- CFRNet [1]

Neural network based methods that learns latent confounders with variational inference

- CEVAE [2]

Ensembles of trees that also rely on strong ignorability

- BART [3] Causal Forest [4]

None of them utilizes network information

[1] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." ICML. 2016.
[2] Louizos, Christos, et al. "Causal effect inference with deep latent-variable models." In NeurIPS, 2017.
[3] Hill, Jennifer L. "Bayesian nonparametric modeling for causal inference." *Journal of Computational and Graphical Statistics.* 2011.
[4] Wager, Stefan, and Susan Athey. "Estimation and inference of heterogeneous treatment effects using random forests." *JASA.* 2018.

Network Deconfounder

How to leverage network information to learn latent confounders?

- Use observed features and network information through Graph Convolutional Networks (GCN).

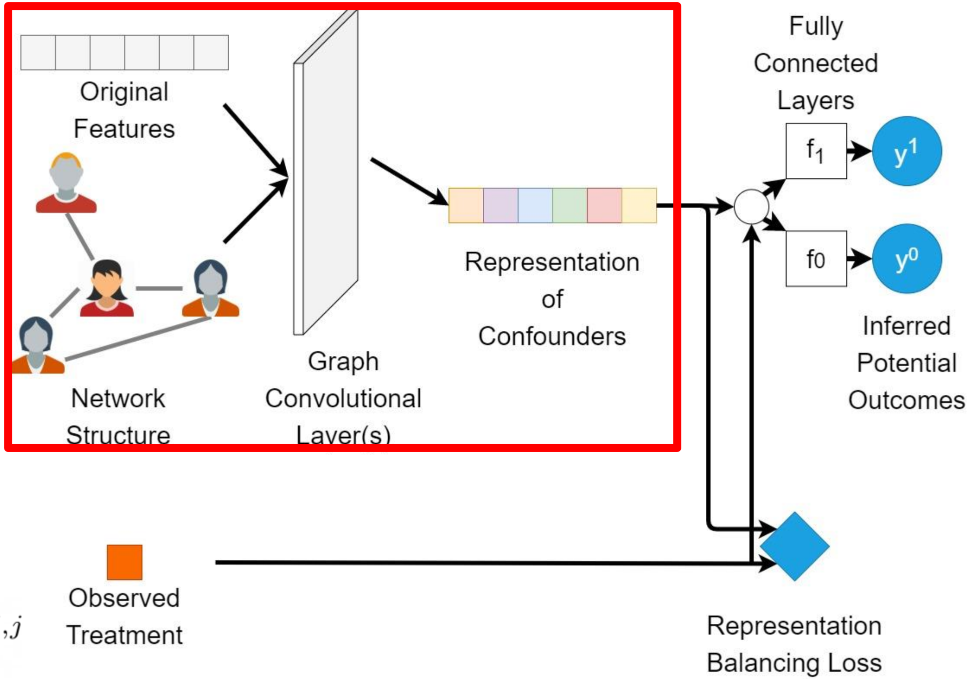
$$\hat{\mathbf{z}}_i \quad = g(\mathbf{x}_i, \mathbf{A}) = \sigma((\hat{\mathbf{A}}\mathbf{X})_i\mathbf{U}),$$

$\hat{\mathbf{z}}_i$ Learned latent confounders

$\hat{\mathbf{A}}$ Normalized adjacency matrix with renormalization trick [1]

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}} \quad \tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n \quad \tilde{\mathbf{D}}_{j,j} = \sum_j \tilde{\mathbf{A}}_{j,j}$$

$\mathbf{U}$ Weight matrix of the GCN layer (parameters to be learned)


Original Features
Network Structure
Graph Convolutional Layer(s)
Representation of Confounders
Fully Connected Layers
f_1
f_0
y^1
y^0
Inferred Potential Outcomes
Observed Treatment
Representation Balancing Loss

[1]Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).

Network Deconfounder

How to leverage network information to learn latent confounders?

- Using the supervision of observed potential outcomes.

Minimize the MSE on factual outcomes

$$\min \frac{1}{n} \sum_{i=1}^{N} (\hat{y}_i^{t_i} - y_i)^2.$$

Inferred outcome

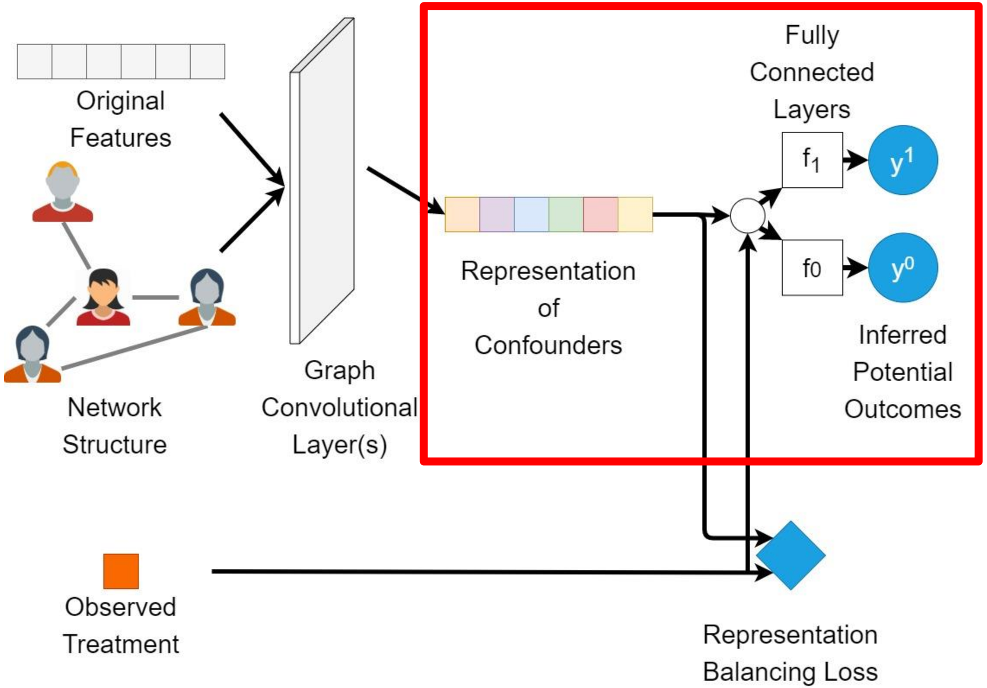
$$\hat{y}_i^t = f(g(\mathbf{x}_i, \mathbf{A}), t),$$

Outcome inference function

$$f(\hat{\mathbf{z}}_i, t) = \begin{cases} f_1(\hat{\mathbf{z}}_i) \text{ if } t = 1 \\ f_0(\hat{\mathbf{z}}_i) \text{ if } t = 0 \end{cases},$$

Fully connected layers for regression

$$f_1 = \mathbf{w}^1 \sigma(\mathbf{W}_L^1 ... \sigma(\mathbf{W}_1^1 \hat{\mathbf{z}}_i)),$$
$$f_0 = \mathbf{w}^0 \sigma(\mathbf{W}_L^0 ... \sigma(\mathbf{W}_1^0 \hat{\mathbf{z}}_i)),$$


Original Features
Network Structure
Graph Convolutional Layer(s)
Representation of Confounders
Fully Connected Layers
f_1
$f0$
y^1
y^0
Inferred Potential Outcomes
Observed Treatment
Representation Balancing Loss

Network Deconfounder

How to leverage network information to learn latent confounders?

- Representation balancing: mitigate confounding bias/domain shift problem.

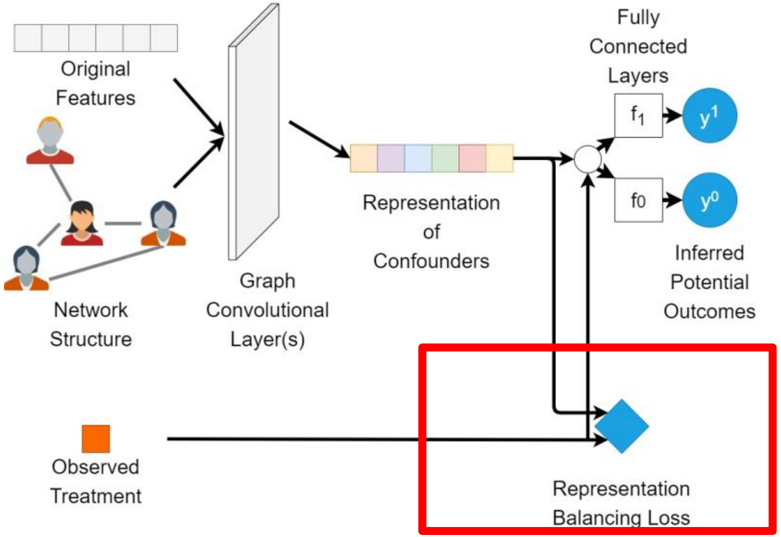
$$\min \rho_{\mathcal{Z}}(P, Q)$$

Minimize W-1 distance between latent confounder distribution of the treated (P) and the controlled (Q)

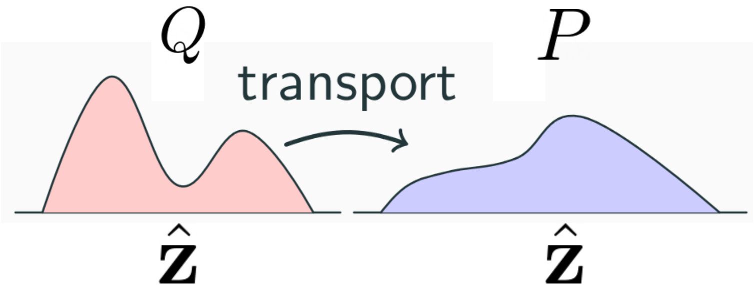
$$\rho_{\mathcal{Z}}(P, Q) = \inf_{k \in \mathcal{K}} \int_{\hat{\mathbf{z}} \in \{\hat{\mathbf{z}}_i\}_{i:t_i=1}} ||k(\hat{\mathbf{z}}) - \hat{\mathbf{z}}|| P(\hat{\mathbf{z}}) d\hat{\mathbf{z}}$$

W-1 distance is the solution of the optimal transport problem between two distributions.

$$\mathcal{K} = \{k | k : R^d \rightarrow R^d \ s.t. \ Q(k(\hat{\mathbf{z}})) = P(\hat{\mathbf{z}})\}$$



Original Features
Network Structure
Graph Convolutional Layer(s)
Representation of Confounders
Fully Connected Layers
f_1
f_0
y^1
y^0
Inferred Potential Outcomes
Observed Treatment
Representation Balancing Loss



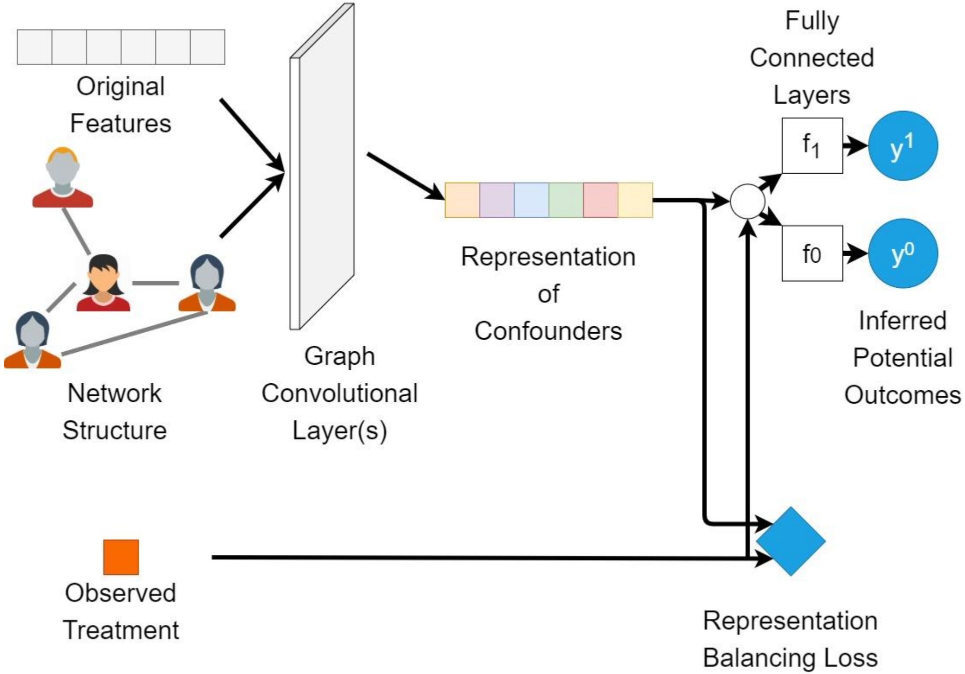
Network Deconfounder

How to leverage network information to handle confounding bias?

- Loss function

$$\mathcal{L}(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^n, \mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^{t_i} - y_i)^2 + \boxed{\alpha\rho_{\mathcal{Z}}(P, Q)} + \lambda||\theta||_2^2,$$

The W-1 distance (representation balancing penalty) and its gradients can be approximated using the efficient algorithm proposed by [1].

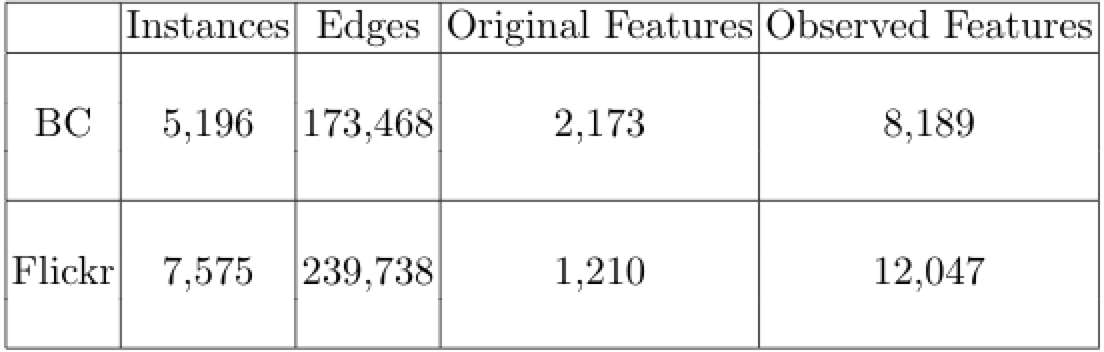


Original Features
Network Structure
Graph Convolutional Layer(s)
Representation of Confounders
Fully Connected Layers
f1
f0
y1
y0
Inferred Potential Outcomes
Observed Treatment
Representation Balancing Loss

[1] Cuturi, Marco, and Arnaud Doucet. "Fast Computation of Wasserstein Barycenters." In *International Conference on Machine Learning*, pp. 685-693. 2014.

Experimental Setup

- Semi-synthetic datasets
  - We obtain features and network information from real-world datasets.
  - We synthesize treatments and outcomes similar to the news dataset in [1].
  - We consider various strength of hidden confounding controlled by parameter κ_2



	Instances	Edges	Original Features	Observed Features
BC	5,196	173,468	2,173	8,189
Flickr	7,575	239,738	1,210	12,047

Code & Data:
https://github.com/rguo12/network-deconfounder-wsdm20

Training/validation/test = 60% : 20% : 20%

[1] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." ICML. 2016.

Experimental Setup

- Baselines
  - SOTA neural network based and ensemble based causal inference methods
    - CFRNet [1]
    - CEVAE [2]
    - BART [3]
    - Causal Forest [4]
- Evaluation:
  - compare the estimated causal effects with the ground truth
  - robustness under various hidden confounding strength
- Metrics:
  - RMSE on estimated ITEs

$$\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{n}\sum_{i=1}(\hat{\tau}_i - \tau_i)^2},$$

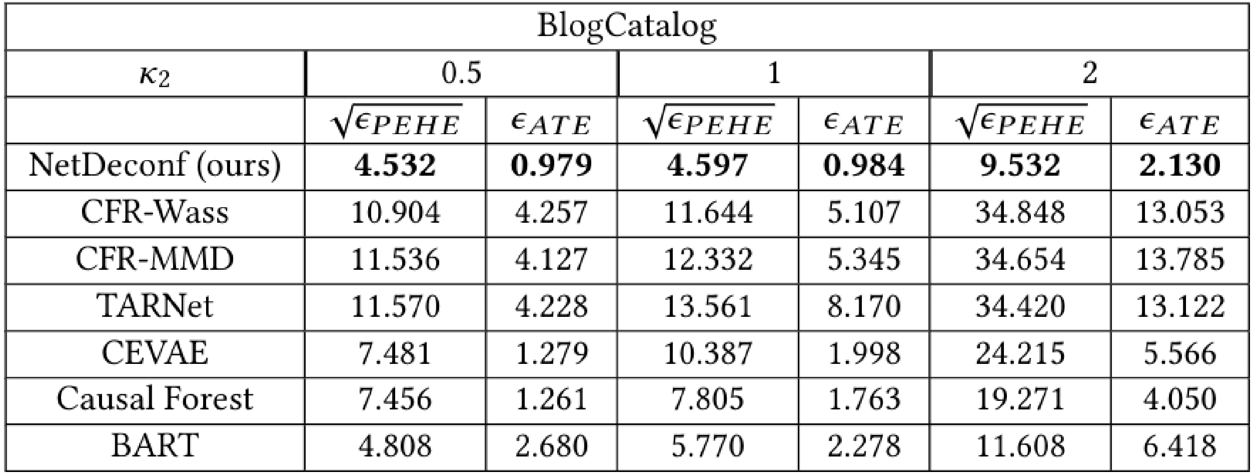
  - MAE on the estimated average treatment effect (ATE)

$$\epsilon_{ATE} = |\frac{1}{n}\sum_{i=1}(\hat{\tau}_i) - \frac{1}{n}\sum_{i=1}(\tau_i)|,$$

[1] Johansson, Fredrik, Uri Shalit, and David Sontag. "Learning representations for counterfactual inference." ICML. 2016.
[2] Louizos, Christos, et al. "Causal effect inference with deep latent-variable models." In NeurIPS, 2017.
[3] Hill, Jennifer L. "Bayesian nonparametric modeling for causal inference." *Journal of Computational and Graphical Statistics.* 2011.
[4] Wager, Stefan, and Susan Athey. "Estimation and inference of heterogeneous treatment effects using random forests." *JASA.* 2018.

Results

- Network Deconfounder outperforms the state-of-the-art methods as it recognizes patterns of hidden confounders from network information.
- The error of Network Deconfounder increases the least as the influence of hidden confounding (κ_2) increases.



BlogCatalog						
κ_2	0.5		1		2	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
NetDeconf (ours)	**4.532**	**0.979**	**4.597**	**0.984**	**9.532**	**2.130**
CFR-Wass	10.904	4.257	11.644	5.107	34.848	13.053
CFR-MMD	11.536	4.127	12.332	5.345	34.654	13.785
TARNet	11.570	4.228	13.561	8.170	34.420	13.122
CEVAE	7.481	1.279	10.387	1.998	24.215	5.566
Causal Forest	7.456	1.261	7.805	1.763	19.271	4.050
BART	4.808	2.680	5.770	2.278	11.608	6.418

Counterfactual Evaluation of Treatment Assignment Functions with Networked Observational Data

Problem Definition

Given: Networked observational data $(\{\mathbf{x}_i, t_i, y_i\}_{i=1}^{N}, \mathbf{A})$ and a **treatment assignment function** $\pi : \mathcal{X} \times \mathcal{A} \rightarrow (0, 1)$

Find: estimate of the true utility of the treatment assignment function π on the given data

$$\tau(\pi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t \in \{0,1\}} \pi^t(\mathbf{x}_i, \mathbf{A}) y_i(t).$$

Existing Methods

There are three types of classic estimators

Direct Method [1] $\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t}\pi^{t}(\mathbf{x}_i)\hat{y}_i(t)$

Weighted Estimator [2, 3] $\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}\hat{w}(\mathbf{x}_i, t_i)y_i,$

- Inverse Propensity Scoring (IPS) [2] $\hat{w}_{IPS}(\mathbf{x}_i, t_i) = \frac{\pi^{t_i}(\mathbf{x}_i)}{P(t = t_i|\mathbf{x})},$
- Self Normalized IPS (SNIPS) [3]

Doubly Robust Estimator (combination of the aforementioned two) [4]

$$\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}[\sum_{t}\pi^{t}(\mathbf{x}_i)\hat{y}_i(t) + \hat{w}_{IPS}(\mathbf{x}_i, t_i)(y_i - \hat{y}_i(t_i))].$$

- Individual causal effect modeling
- Propensity score modeling

Network information has not been used!

[1] Beygelzimer, A., Dasgupta, S., & Langford, J.. Importance weighted active learning. ICML 2009.

[2] Swaminathan, A., & Joachims, T.. Counterfactual risk minimization: Learning from logged bandit feedback. ICML 2015.

[3] Swaminathan, A., & Joachims, T. (2015). The self-normalized estimator for counterfactual learning. NIPS 2015.

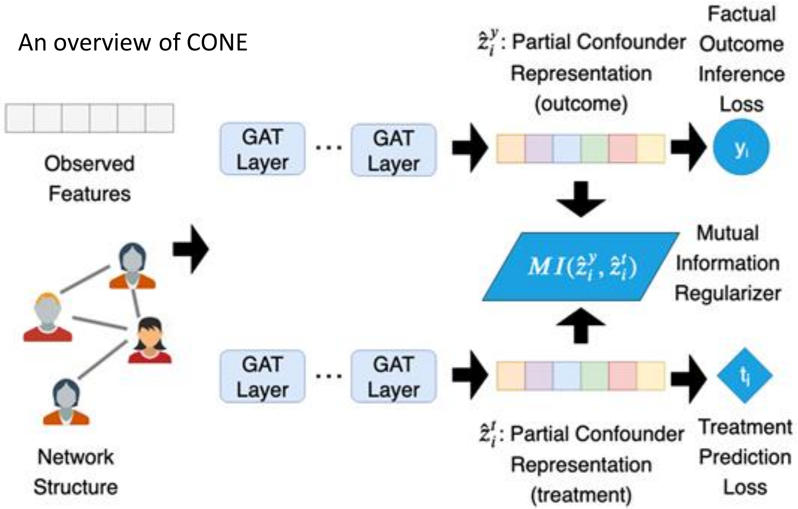
[4] Dudík, M., Langford, J., & Li, L.. Doubly robust policy evaluation and learning. ICML 2011.

Proposed Framework

COunterfactual Network Evaluator (CONE)

- It learns two partial representations of latent confounders for
  - Individual causal effect modeling
  - Propensity score modeling
- It maximizes the mutual information between the two partial representations to capture latent confounders

An overview of CONE



Observed Features
Network Structure
GAT Layer
...
GAT Layer
GAT Layer
...
GAT Layer
$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
t_i
Treatment Prediction Loss

CONE Framework

Compute partial representations with GAT layers [1]

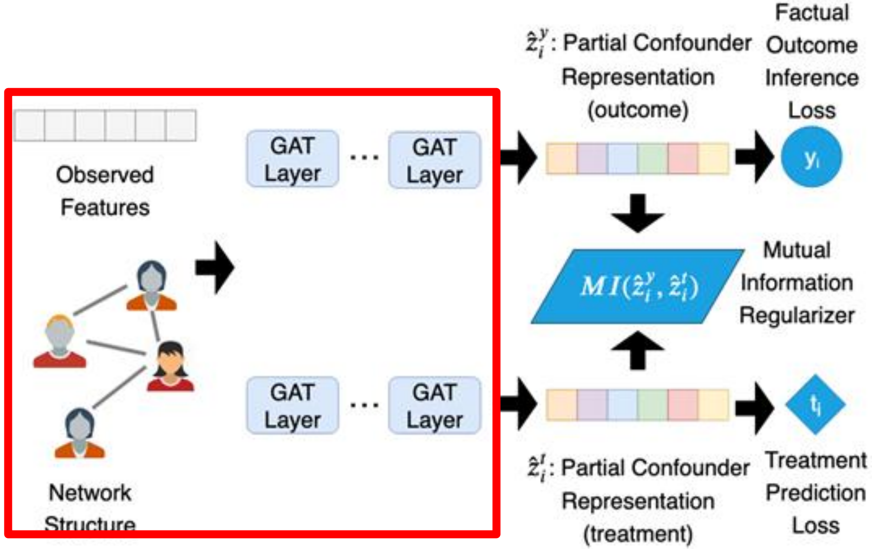
- They capture importance of each edge

$$\hat{\mathbf{z}}_i^t = g^t(\mathbf{x}_i, \mathbf{A}) = \|_{k=1}^{K} \delta(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{x}_j)$$

$$\hat{\mathbf{z}}_i^y = g^y(\mathbf{x}_i, \mathbf{A}) = \|_{k=1}^{K} \delta(\sum_{j \in \mathcal{N}_i} \beta_{ij}^k \mathbf{U}^k \mathbf{x}_j)$$

$$\alpha_{ij}^k = \frac{\exp(\delta'(\mathbf{a}^T[\mathbf{W}^k \mathbf{x}_i \parallel \mathbf{W}^k \mathbf{x}_j]))}{\sum_{l \in \mathcal{N}_i} \exp(\delta'(\mathbf{a}^T[\mathbf{W}^k \mathbf{x}_i \parallel \mathbf{W}^k \mathbf{x}_l]))}$$

$$\beta_{ij}^k = \frac{\exp(\delta'(\mathbf{b}^T[\mathbf{U}^k \mathbf{x}_i \parallel \mathbf{U}^k \mathbf{x}_j]))}{\sum_{l \in \mathcal{N}_i} \exp(\delta'(\mathbf{b}^T[\mathbf{U}^k \mathbf{x}_i \parallel \mathbf{U}^k \mathbf{x}_l]))}$$



Observed Features
Network Structure
GAT Layer
...
GAT Layer
GAT Layer
...
GAT Layer
$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
t_i
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
Treatment Prediction Loss

a and **b** are weight vectors

W^k and **U**^{k} is the weight matrix of the k-th head

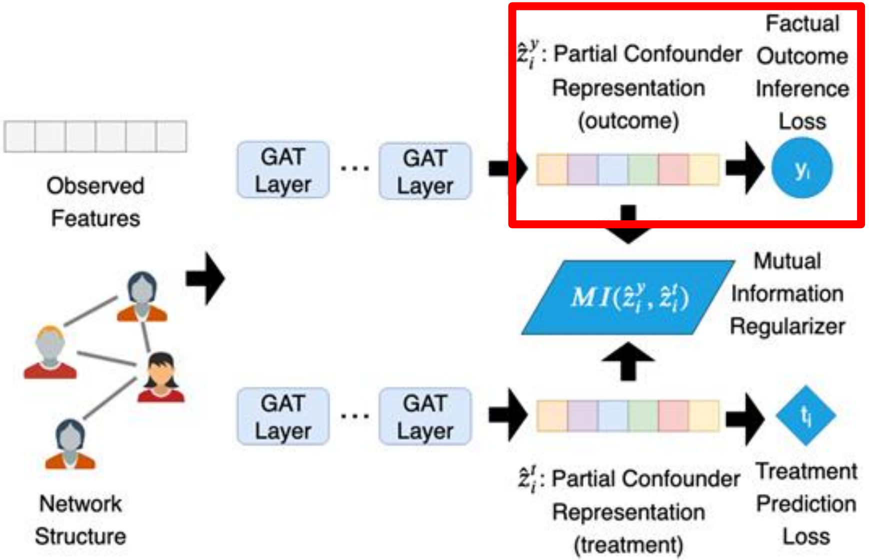
[1] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y.. Graph attention networks. ICLR 2018.

CONE Framework

Factual Outcome Inference Loss

- Let the partial representation of latent confounders predict factual outcome
- We use fully connected NN with ELU activation and MSE penalty

$$\mathcal{L}^y = \frac{1}{N}\sum_{i=1}^{N}(f^y(\hat{\mathbf{z}}_i^y) - y_i)^2.$$


Observed Features
Network Structure
GAT Layer
...
GAT Layer
$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
GAT Layer
...
GAT Layer
t_i
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
Treatment Prediction Loss

CONE Framework

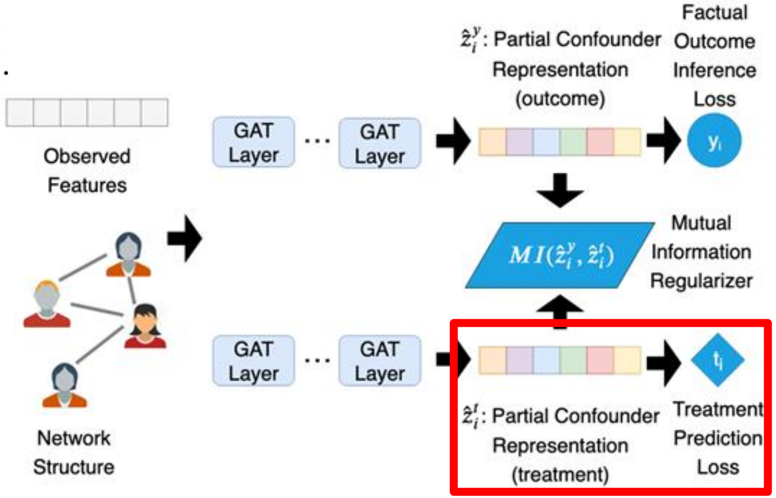
Treatment Prediction Loss

- CONE models propensity scores with the other partial representation
- CONE infers propensity score by a fully connected NN with sigmoid activation

$$\hat{P}(t = 1|\hat{\mathbf{z}^t}) = f^t(\hat{\mathbf{z}^t}) = \sigma(\mathbf{v}^T \hat{\mathbf{z}}_i^t + c)$$

- CONE uses cross-entropy loss for the propensity score model

$$\mathcal{L}^t = -\frac{1}{N}\sum_i t_i \log(\hat{P}(t = 1|\hat{\mathbf{z}^t})) + (1 - t_i)\log(\hat{P}(t = 0|\hat{\mathbf{z}^t})).$$



Observed Features
Network Structure
GAT Layer
...
GAT Layer
$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
GAT Layer
...
GAT Layer
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
t_i
Treatment Prediction Loss

CONE Framework

Mutual Information Maximization

Intuition

- Latent confounders should influence treatment and outcome.

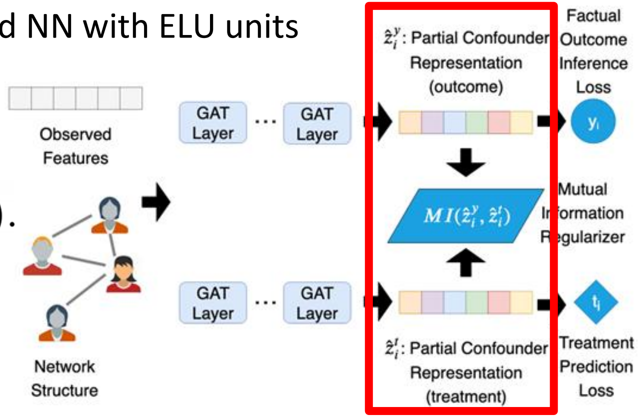
A tight lower bound of mutual information is computed by an NN estimator [1].

$$MI(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y}) = D_{KL}(P(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})||P(\hat{\mathbf{z}^t}) \otimes P(\hat{\mathbf{z}^y})) = \sup_{h \in \mathcal{H}} E_{P(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})}[h(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})] - \log(E_{P(\hat{\mathbf{z}^y}) \otimes P(\hat{\mathbf{z}^y})}[e^{h(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})}]),$$

h: fully connected NN with ELU units

The penalty term is formulated to minimize the negative mutual information

$$\mathcal{L}^{MI} = -E_{P(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})}[h(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})] + \log(E_{P(\hat{\mathbf{z}^y}) \otimes P(\hat{\mathbf{z}^y})}[\exp^{h(\hat{\mathbf{z}^t}, \hat{\mathbf{z}^y})}]).$$


$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
Observed Features
GAT Layer
...
GAT Layer
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
GAT Layer
...
GAT Layer
t_i
Network Structure
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
Treatment Prediction Loss

[1] Belghazi, Mohamed Ishmael, Sai Rajeswar, Aristide Baratin, Devon Hjelm, and Aaron Courville. "MINE: Mutual Information Neural Estimation." ICML 2018.

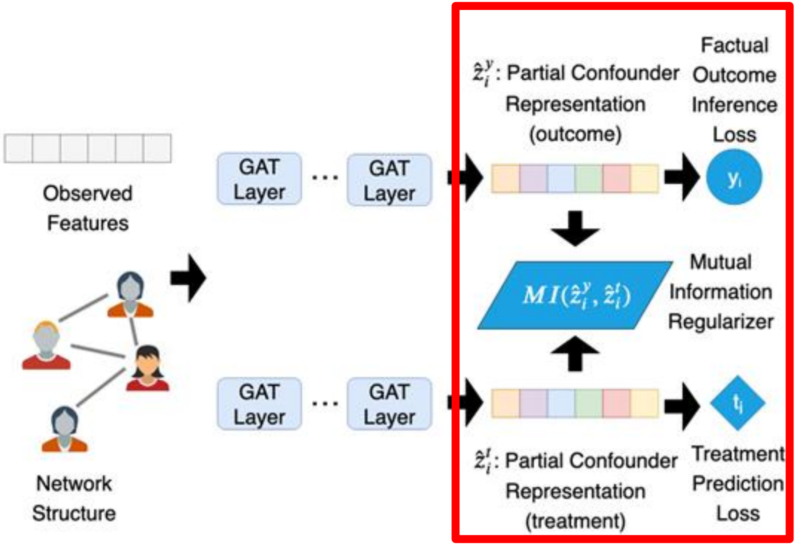
CONE Framework

Training CONE with a combination of the loss functions

$$\arg\min_{\boldsymbol{\theta}_{-h}, \boldsymbol{\theta}_h} \mathcal{L} = \mathcal{L}^y + \gamma\mathcal{L}^t + \zeta\mathcal{L}^{MI},$$

θ_h - the NN mutual information estimator's parameters

θ_{-h} - other model parameters


Observed Features
Network Structure
GAT Layer
...
GAT Layer
GAT Layer
...
GAT Layer
$\hat{z}_i^y$: Partial Confounder Representation (outcome)
Factual Outcome Inference Loss
y_i
$MI(\hat{z}_i^y, \hat{z}_i^t)$
Mutual Information Regularizer
$\hat{z}_i^t$: Partial Confounder Representation (treatment)
t_i
Treatment Prediction Loss

Counterfactual Evaluation with CONE

Goal: estimating utility of a treatment assignment function

We combine the two partial representations and adopt the doubly robust estimator with SNIPS weights

$$\hat{\mathbf{z}}_i = concat([\hat{\mathbf{z}}_i^y, \hat{\mathbf{z}}_i^t])$$

Inferred outcomes

$$\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}[\sum_{t}\pi^t(\mathbf{x}_i, \mathbf{A})\boxed{\hat{y}_i(\hat{\mathbf{z}}_i, t)}$$
$$+\boxed{\hat{w}_{SNIPS}(\hat{\mathbf{z}}_i, t_i)}(y_i - \hat{y}_i(\hat{\mathbf{z}}_i, t_i))],$$

By simple direct method in [1]

SNIPS weights $\hat{w}_{SNIPS}(\hat{\mathbf{z}}_i, t_i) = \frac{\hat{w}_{IPS}(\hat{\mathbf{z}}_i, t_i)}{\sum_{i=1}^{N}\hat{w}_{IPS}(\hat{\mathbf{z}}_i, t_i)},$

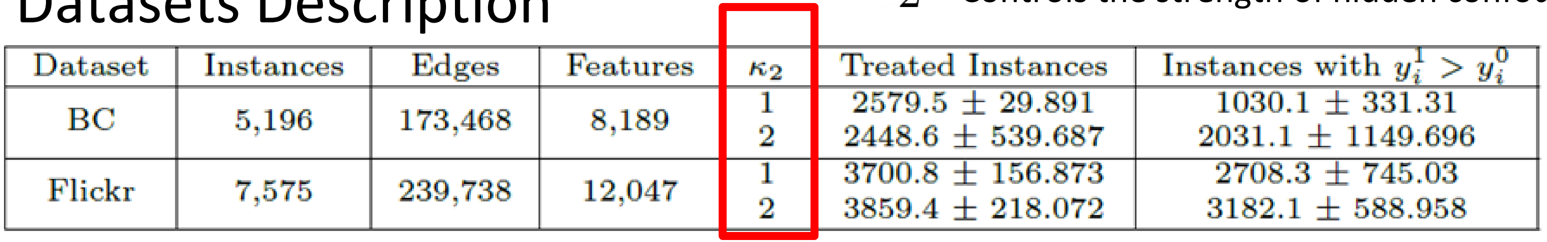
With propensity scores by a logistic regression model

[1] Bennett, Andrew, and Nathan Kallus. "Policy evaluation with latent confounders via optimal balance." In Advances in Neural Information Processing Systems, pp. 4827-4837. 2019.

Experimental Settings

Datasets Description

κ_2 Controls the strength of hidden confounding



Dataset	Instances	Edges	Features	κ_2	Treated Instances	Instances with $y_i^1 > y_i^0$
BC	5,196	173,468	8,189	1	2579.5 ± 29.891	1030.1 ± 331.31
				2	2448.6 ± 539.687	2031.1 ± 1149.696
Flickr	7,575	239,738	12,047	1	3700.8 ± 156.873	2708.3 ± 745.03
				2	3859.4 ± 218.072	3182.1 ± 588.958

Compared to the one used in network deconfounder, we standardize the outcomes into [0,1] and introduce negative ITEs.

Training/validation/test = 60% : 20% : 20%

Evaluation

Given treatment assignment functions with randomly sampled weights:

$$\pi^t_{rw}(\mathbf{x}_i, \mathbf{A}) = \frac{\exp({\psi^t}^T \mathbf{x}_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} {\delta^t}^T \mathbf{x}_j)}{\sum_t \exp({\psi^t}^T \mathbf{x}_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} {\delta^t}^T \mathbf{x}_j)}$$

We compare the estimated utility of a given treatment assignment function with the group truth

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{\tau}_k(\pi) - \tau_k(\pi))^2}$$

$$MAE = \frac{1}{K}\sum_{k=1}^{K}|\hat{\tau}_k(\pi) - \tau_k(\pi)|,$$

K is the number of randomly sampled treatment assignment functions.

Experimental Setup

Baseline Methods: 9 SOTA baseline methods

Optimal Kernel Balancing (OKB)

Inverse Propensity Scoring (IPS-X)

Self-Normalized Inverse Propensity Scoring (SNIPS-X)

Direct Method (OLS1, OLS2, DM-X)

Linear regression methods: OLS1, OLS2

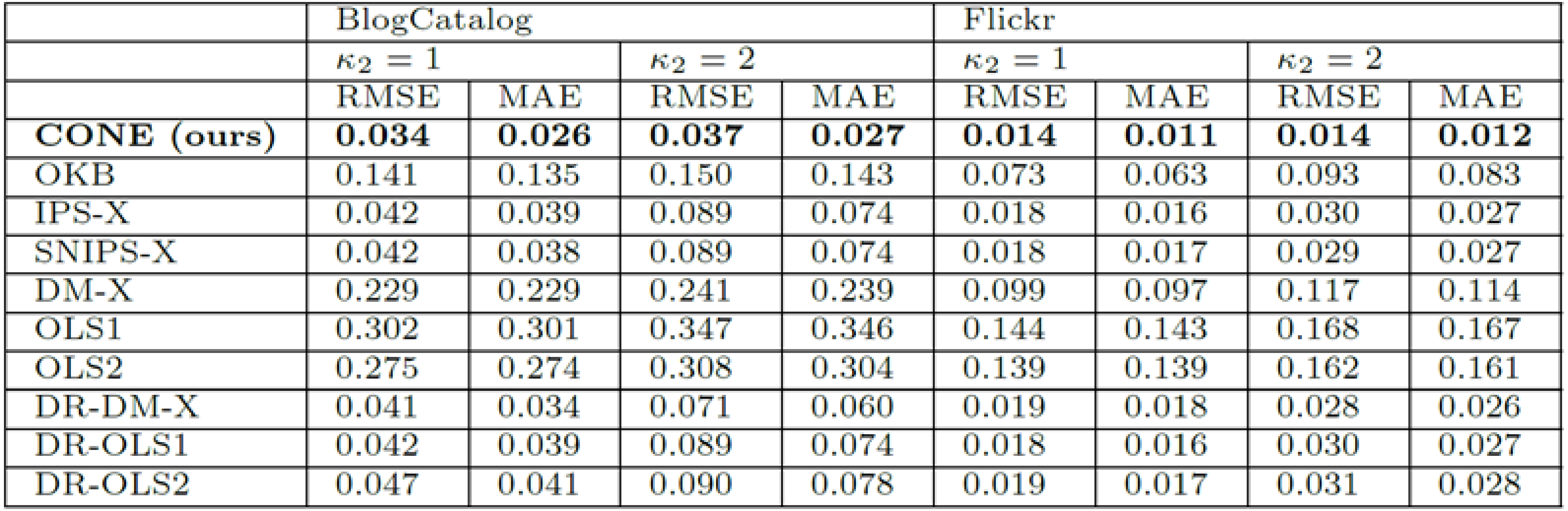
NN-based: DM-X

Doubly Robust Estimators (DR-OLS1, DR-OLS2, DR-DM-X)

Effectiveness

Observations

- CONE outperforms baselines consistently and significantly
- CONE worsens less than baselines when hidden confounding effect (κ_2) grows
- Using network information helps counterfactual evaluation



	BlogCatalog				Flickr			
	$\kappa_2 = 1$		$\kappa_2 = 2$		$\kappa_2 = 1$		$\kappa_2 = 2$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
CONE (ours)	**0.034**	**0.026**	**0.037**	**0.027**	**0.014**	**0.011**	**0.014**	**0.012**
OKB	0.141	0.135	0.150	0.143	0.073	0.063	0.093	0.083
IPS-X	0.042	0.039	0.089	0.074	0.018	0.016	0.030	0.027
SNIPS-X	0.042	0.038	0.089	0.074	0.018	0.017	0.029	0.027
DM-X	0.229	0.229	0.241	0.239	0.099	0.097	0.117	0.114
OLS1	0.302	0.301	0.347	0.346	0.144	0.143	0.168	0.167
OLS2	0.275	0.274	0.308	0.304	0.139	0.139	0.162	0.161
DR-DM-X	0.041	0.034	0.071	0.060	0.019	0.018	0.028	0.026
DR-OLS1	0.042	0.039	0.089	0.074	0.018	0.016	0.030	0.027
DR-OLS2	0.047	0.041	0.090	0.078	0.019	0.017	0.031	0.028

Conclusion

We propose

- a causal identification strategy
- two novel frameworks
- to solve two causal inference problems with networked observational data.

Empirical results support the hypothesis

- **with network information, we can improve the learned latent confounders**
- for causal effect estimation and counterfactual evaluation.

Proposed Work for Dissertation Defense

Fairness of Treatment Assignment Functions

Fairness is important in decision making

- Resources of beneficial treatments are often limited.
- For example, In COVID-19, Small Business Administration (SBA) needs to distribute limited amount of loans.

When we adapt fairness metrics from machine learning to causal inference, we find some of them depend on counterfactuals.

- They lead to challenging causal identification problems

Advanced Representation Balancing

Two desiderata in learning latent confounders

- Balancing
- Treatment prediction

Existing methods are developed toward one of them, but not both.

Representation balancing has implication in causality-aware machine learning tasks.

- Invariant risk minimization

Timeline

May 2020
Dissertation Prospectus Defense

June 2020 - November 2020
Investigating the Proposed Problems

November - December 2020
Dissertation Writing

December 2020 - January 2021
Dissertation Defense

Selected Publications

[1] **Ruocheng Guo**, Lu Cheng, Jundong Li, P. Richard Hahn and Huan Liu, “A Survey of Learning Causality with Data: Problems and Methods”, ACM Computing Surveys

[2] **Ruocheng Guo**, Jundong Li, Yichuan Li, K. Selçuk Candan, Adrienne Raglin and Huan Liu, “IGNITE: A Minimax Game Toward Learning Individual Treatment Effects from Networked Observational Data”, IJCAI 2020

[3] **Ruocheng Guo**, Jundong Li and Huan Liu, “Learning Individual Treatment Effects from Networked Observational Data”, WSDM 2020

[4] **Ruocheng Guo**, Jundong Li and Huan Liu, “Counterfactual Evaluation of Treatment Assignment Functions with Networked Observational Data”, SDM 2020

[5] **Ruocheng Guo**, Jundong Li and Huan Liu, “INITIATOR: Noise-contrastive Estimation for Marked Temporal Point Process”, IJCAI 2018

[6] Vineeth Rakesh*, **Ruocheng Guo***, Raha Moraffah, Nitin Agarwal and Huan Liu (* Equal Contribution), “Linked Causal Variational Autoencoder for Inferring Paired Spillover Effects”, CIKM 2018

[7] **Ruocheng Guo**, Hamidreza Alvari and Paulo Shakarian, “Strongly Hierarchical Factorization Machines and ANOVA Kernel Regression”, SDM 2018

[8] **Ruocheng Guo** and Paulo Shakarian, “A Comparison of Methods for Cascade Prediction”, ASONAM 2016

[9] **Ruocheng Guo**, Elham Shaabani, Abhinav Bhatnagar and Paulo Shakarian, ”Toward Early and Order-of-magnitude Cascade Prediction in Social Networks”, Springer SNAM

[10] **Ruocheng Guo**, Elham Shaabani, Abhinav Bhatnagar and Paulo Shakarian, “Towards Order-of-magnitude Cascade Prediction”, ASONAM 2015

[11] Ghazaleh Beigi, Ahmadreza Mosallanezhad, **Ruocheng Guo**, Alex Nou, Hamidreza Alvari and Huan Liu “Privacy-Aware Recommendation with Private-Attribute Protection using Adversarial Learning”, WSDM 2020

[12] Jundong Li, **Ruocheng Guo**, Chenghao Liu and Huan Liu, “Adaptive Unsupervised Feature Selection on Attributed Networks”, KDD 2019

Full list can be found at my google scholar page/ DBLP.

Acknowledgement

Ph.D. dissertation committee members

Dr. Huan Liu

Dr. K. Selçuk Candan

Dr. Guoliang Xue

Dr. Emre Kiciman

Collaborators

Dr. Adrienne Raglin (ARL)
Dr. P. Richard Hahn (ASU)
Dr. Liangjie Hong (LinkedIn)
Dr. Xiaoting Zhao (Etsy)
Dr. Adam Henderson (Etsy)
Dr. Paulo Shakarian (ASU)
Dr. Elham Shaabani (ASU)
Dr. Hamidreza Alvari (ASU)
Dr. Soumajyoti Sarkar (ASU)
Dr. Ericsson Marin (ASU)

DMML lab members

Funding

NSF
ARL
ONR

Q & A