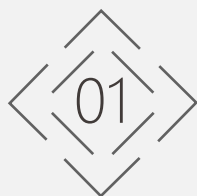


跨年龄人脸情感识别

Cross-age face emotion recognition

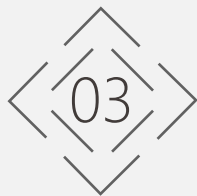
目录



项目内容



工作安排



目前的工作进展

第一部分

Part One

项目内容

Project Description

项目内容

跨年龄的人脸情感识别

人有幼儿期、少年期、青年期、中年期、老年期这几个时期，在每个时期，人情感的表达方式都有所不同，比如：在幼儿时期，人的表达情感更为激烈，但是到了中年之后，人表达情感则更为含蓄。



跨年龄的人脸情感识别

本次项目基于这种年龄所带来的情感表达差异，来研究跨年龄的人脸情感识别。现阶段拟采用的方法是先采用**GCN**对面部表情进行特征提取，接着对提取到的特征进行训练，然后再使用**DAL**相关对抗学习去除年龄和情感带来的依赖，以实现跨年龄人脸情感识别。



第二部分

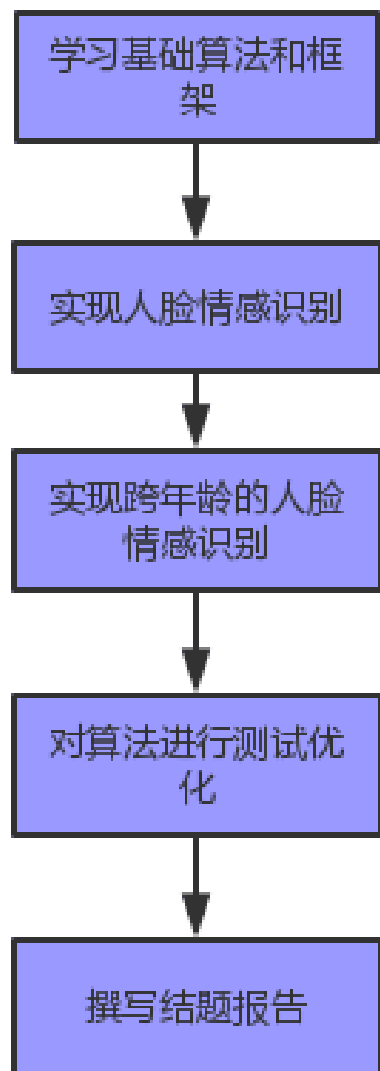
Part Two

工作安排

Working arrangements

工作安排

- 1.学习基础算法和框架
- 2.实现人脸情感识别
- 3.实现跨年龄的人脸情感识别
- 4.对算法进行测试优化
- 5.撰写结题报告





工作安排

现阶段工作安排：

- 1.边查找资料，边实现例子，边学习
- 2.实现普通人脸情感识别（12月中旬）
- 3.每周开展两次讨论



第三部分

Part Three

目前的工作进展

Current work progress



目前的工作进展

利用GCN提取图像的特征

- 1.在Linux上利用Anaconda3安装了Tensorflow、pytorch
- 2.找到了一篇还不错的例子

目前的工作进展

例子展示

GCN应用

本例中的训练, 评测和可视化的完整代码在下面的git仓库中, 后面还会陆续更新一些其他GNN算法

shenweichen/GraphNeuralNetwork

github.com



我们使用论文引用网络数据集Cora进行测试, Cora数据集包含2708个顶点, 5429条边, 每个顶点包含1433个特征, 共有7个类别。

按照论文的设置, 从每个类别中选取20个共140个顶点作为训练, 500个顶点作为验证集合, 1000个顶点作为测试集。DeepWalk从全体顶点集合中进行采样, 最后使用同样的140个顶点训练一个LR模型进行分类。

顶点分类任务结果

方法	准确率
DeepWalk	0.70
DeepWalk + feature	0.78
GCN(feature_less)	0.68

Implementation and experiments of graph neural networks, like gcn, graphsage, gat, etc.

gcn graphsage gnn gat graph-attention

7 commits

2 branches

0 packages

0 releases

Branch: master

New pull request



shenweichen Update README.md

data/cora

add GraphSAGE

gnn

Add GAT

.DS_Store

add gcn

.gitattributes

Initial commit

.gitignore

add GraphSAGE

LICENSE

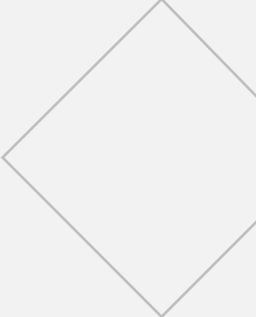
Initial commit

README.md

Update README.md

requirements.txt

add GraphSAGE



Cora数据集由机器学习论文组成。这些论文分为以下七个类别之一：

- 基于案例
- 遗传算法
- 神经网络
- 概率方法
- 补强_学习
- 规则学习
- 理论

选择论文的方式应使最终的语料库中的每篇论文都被另一篇论文引用或引用。整个语料库共有**2708**篇论文。

去除和停用停用词后，我们得到的词汇表大小为**1433**个唯一词。删除文档频率小于**10**的所有单词。

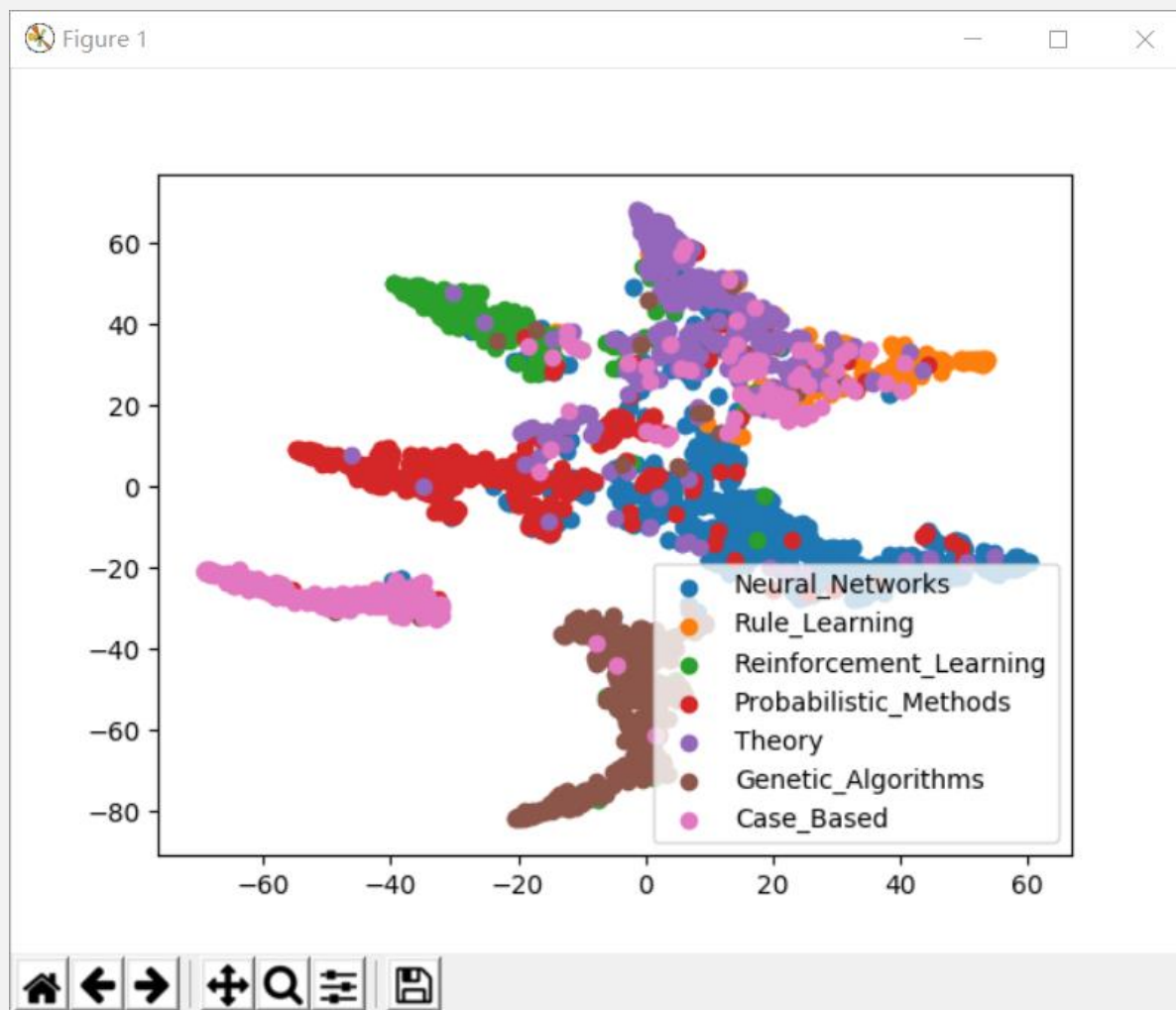
目前的工作进展

Cora数据集包含2708个顶点, 5429条边,每个顶点包含1433个特征, 共有7个类别。

按照论文的设置, 从每个类别中选取20个共140个顶点作为训练, 500个顶点作为验证集合, 1000个顶点作为测试集。DeepWalk从全体顶点集合中进行采样, 最后使用同样的140个顶点训练一个LR模型进行分类。

目前的工作进展

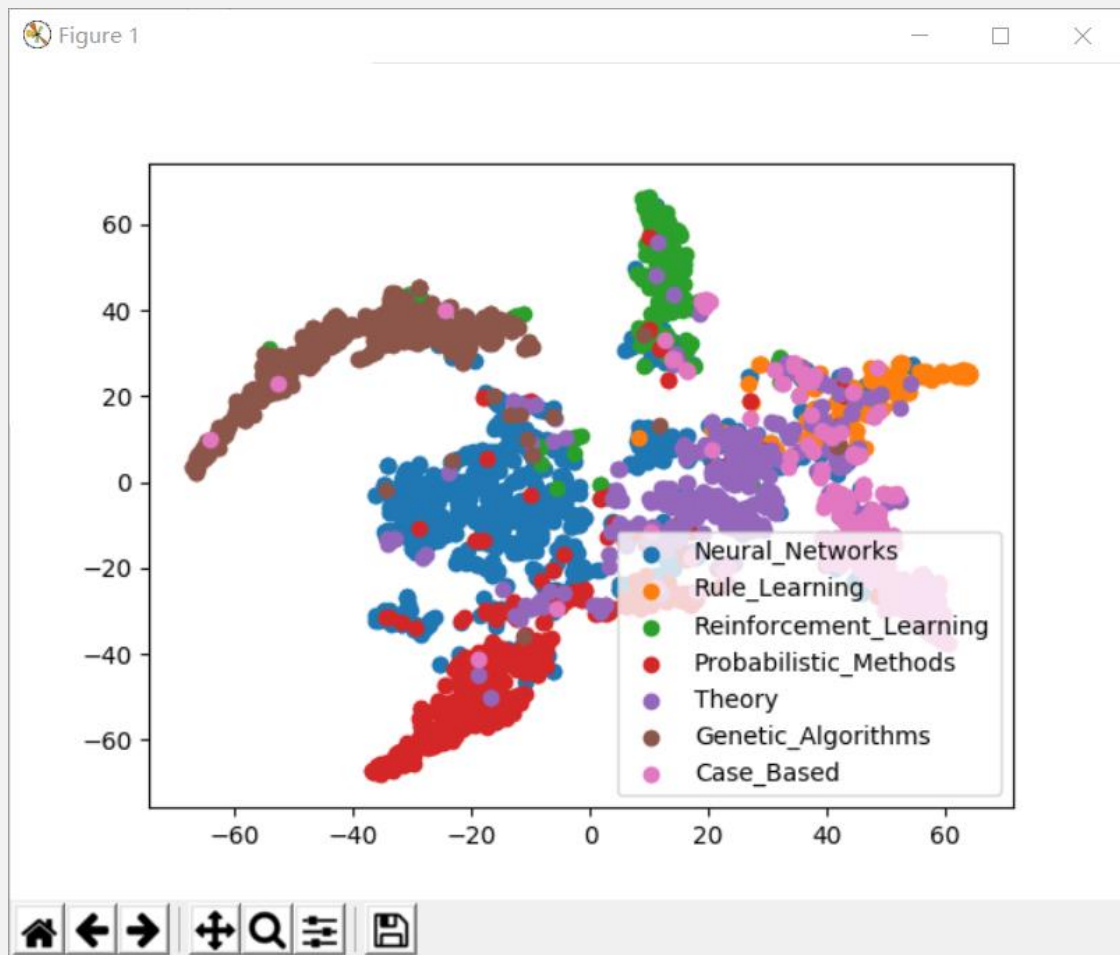
GCN的例子展示



```
Epoch 193/200
0s - loss: 0.9974 - weighted_categorical_crossentropy: 0.5773 -
0.9751 - val_weighted_acc: 0.7720
Epoch 194/200
0s - loss: 1.0504 - weighted_categorical_crossentropy: 0.6297 -
0.9824 - val_weighted_acc: 0.7720
Epoch 195/200
0s - loss: 1.0520 - weighted_categorical_crossentropy: 0.6308 -
0.9923 - val_weighted_acc: 0.7640
Epoch 196/200
0s - loss: 1.0336 - weighted_categorical_crossentropy: 0.6120 -
0.9983 - val_weighted_acc: 0.7640
Epoch 197/200
0s - loss: 1.1004 - weighted_categorical_crossentropy: 0.6782 -
0.9974 - val_weighted_acc: 0.7620
Epoch 198/200
0s - loss: 0.9864 - weighted_categorical_crossentropy: 0.5637 -
0.9893 - val_weighted_acc: 0.7620
Epoch 199/200
0s - loss: 1.0383 - weighted_categorical_crossentropy: 0.6150 -
0.9774 - val_weighted_acc: 0.7680
Epoch 200/200
0s - loss: 1.0067 - weighted_categorical_crossentropy: 0.5827 -
0.9657 - val_weighted_acc: 0.7680
2708/2708 [=====] - 0s
Done.
Test loss: 1.4220404624938965
Test weighted_loss: 1.0023752450942993
Test accuracy: 0.7840006351470947
```

目前的工作进展

GAT的例子展示



```
C:\Windows\System32\cmd.exe - python run_gat_cora.py
1.5462 - val_weighted_acc: 0.7520
Epoch 193/200
ts - loss: 1.7805 - weighted_categorical_crossentropy: 1.6251 - w
1.5427 - val_weighted_acc: 0.7600
Epoch 194/200
ts - loss: 1.6771 - weighted_categorical_crossentropy: 1.5213 - w
1.5413 - val_weighted_acc: 0.7560
Epoch 195/200
ts - loss: 1.7660 - weighted_categorical_crossentropy: 1.6096 - w
1.5406 - val_weighted_acc: 0.7620
Epoch 196/200
ts - loss: 1.6963 - weighted_categorical_crossentropy: 1.5394 - w
1.5411 - val_weighted_acc: 0.7620
Epoch 197/200
ts - loss: 1.8427 - weighted_categorical_crossentropy: 1.6852 - w
1.5413 - val_weighted_acc: 0.7660
Epoch 198/200
ts - loss: 1.7007 - weighted_categorical_crossentropy: 1.5428 - w
1.5420 - val_weighted_acc: 0.7640
Epoch 199/200
ts - loss: 1.7269 - weighted_categorical_crossentropy: 1.5685 - w
1.5435 - val_weighted_acc: 0.7660
Epoch 200/200
ts - loss: 1.6926 - weighted_categorical_crossentropy: 1.5337 - w
1.5445 - val_weighted_acc: 0.7580
2708/2708 [=====] - 2s
Done.
Test loss: 1.7004541158676147
Test weighted_loss: 1.543549656867981
Test accuracy: 0.801000714302063
```



演示完毕 谢谢欣赏