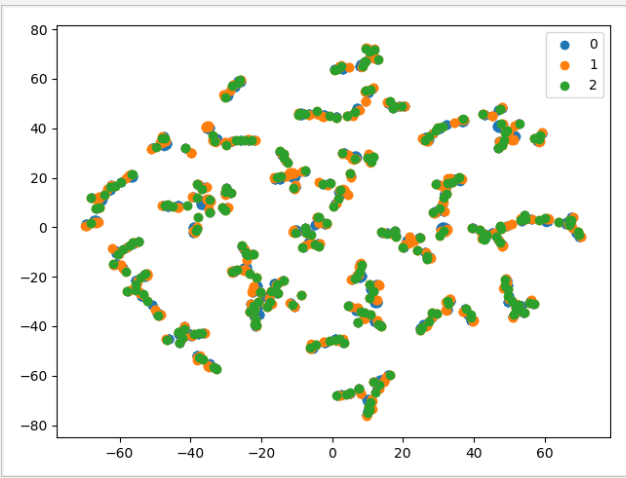
测试样例，**45456803 紧急联系人社区**

1. **Deepwalk 无向图**

参数 walk\_length=20, num\_walks=80

{'micro': 0.55863539445629, 'macro': 0.3777254562348335, 'samples': 0.55863539445629, 'weighted': 0.5116491793805881, 'acc': 0.55863539445629}

聚类效果



参数 walk\_length=40, num\_walks=80

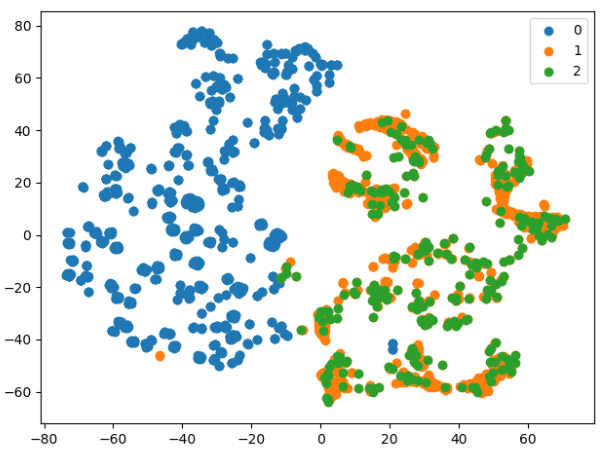
**{'micro': 0.5543710021321961, 'macro': 0.3841693833911733, 'samples': 0.5543710021321961, 'weighted': 0.5166485078326648, 'acc': 0.5543710021321961}**

参数 walk\_length=80, num\_walks=80

{'micro': 0.5671641791044776, 'macro': 0.3868299894555129, 'samples': 0.567164179104477*6, 'weight*ed': 0.5222637678659173, 'acc': 0.5671641791044776}

有向图 参数 walk\_length=40, num\_walks=80

{'micro': 0.8614072494669509, 'macro': 0.6068376068376069, 'samples': 0.8614072494669509, 'weighted': 0.81132797550708, 'acc': 0.8614072494669509}



对于人和电话的区分度非常高，主要没有区分个人电话与紧急联系人电话的能力

1. **Line**

**参数** batch\_size=1024, epochs=150, verbose=2

embedding\_size=128, order=**'second' 无向图**

**{'micro': 0.7782515991471216, 'macro': 0.5461988304093568, 'samples': 0.7782515991471215, 'weighted': 0.7300259834431083, 'acc': 0.7782515991471215}**

**参数** embedding\_size=300, order=**'second' 无向图**

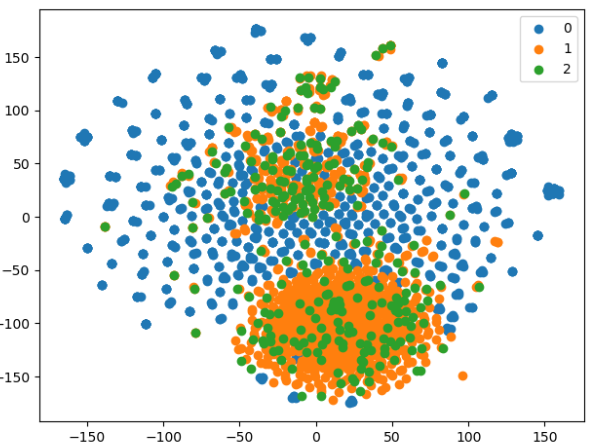
**{'micro': 0.7889125799573561, 'macro': 0.5573834053395369, 'samples': 0.7889125799573561, 'weighted': 0.7459046102630276, 'acc': 0.7889125799573561}参数** embedding\_size=400, order=**'second' 无向图**

**{'micro': 0.8038379530916845, 'macro': 0.5673016916035206, 'samples': 0.8038379530916845, 'weighted': 0.7584780097331941, 'acc': 0.8038379530916845}参数** embedding\_size=500, order=**'second' 无向图**

**{'micro': 0.8081023454157783, 'macro': 0.576820209603922, 'samples': 0.8081023454157783, 'weighted': 0.7726869876649297, 'acc': 0.8081023454157783}**

embedding\_size=600, order=**'second' 无向图**

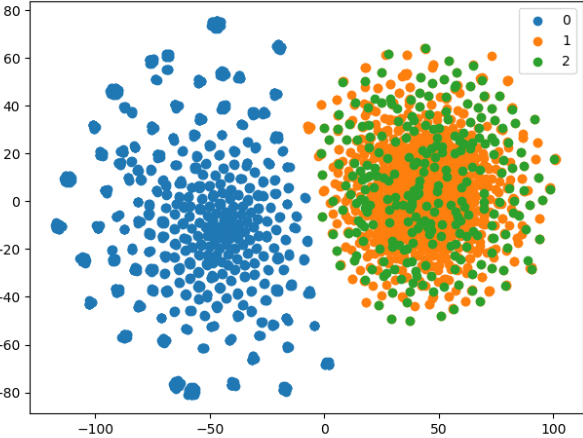
**{'micro': 0.8208955223880597, 'macro': 0.5950248321598297, 'samples': 0.8208955223880597, 'weighted': 0.7875315049604084, 'acc': 0.8208955223880597}**



**参数** embedding\_size=128, order=**'second' 有向图**

{'micro': 0.8848614072494669, 'macro': 0.6244131455399061, 'samples': 0.8848614072494669, 'weighted': 0.8345896273161356, 'acc': 0.8848614072494669}**参数** embedding\_size=300, order=**'second' 有向图**

{'micro': 0.8848614072494669, 'macro': 0.6244131455399061, 'samples': 0.8848614072494669, 'weighted': 0.8345896273161356, 'acc': 0.8848614072494669}



**对电话个用户是有完全的区分能力，对个人电话和紧急联系人电话区分能力弱**

1. **SDNE**

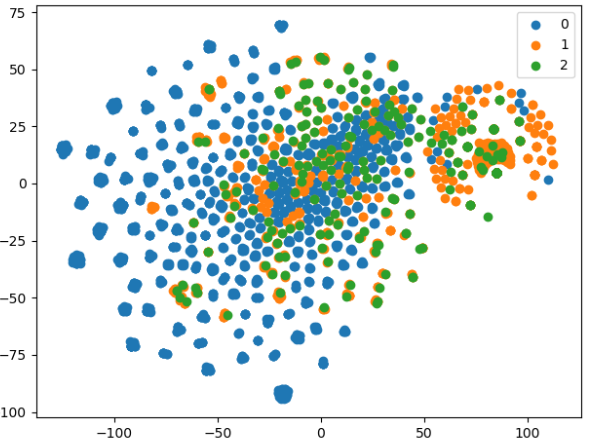
hidden\_size=[256, 128] 无向图

batch\_size=3000, epochs=150, verbose=2

**{'micro': 0.7057569296375267, 'macro': 0.4994633701039987, 'samples': 0.7057569296375267, 'weighted': 0.6673411954258115, 'acc': 0.7057569296375267}**

hidden\_size=[256, 256]

**{'micro': 0.7505330490405118, 'macro': 0.5307944307944309, 'samples': 0.7505330490405118, 'weighted': 0.7083375680390607, 'acc': 0.7505330490405118}**



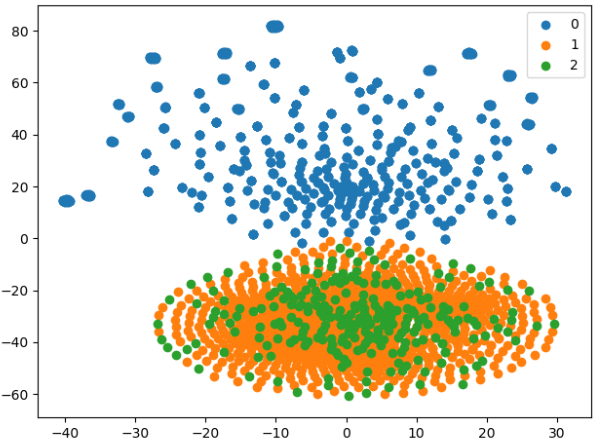
hidden\_size=[256, 128] 有向图

batch\_size=3000, epochs=150, verbose=2

**{'micro': 0.8805970149253731, 'macro': 0.6215909711974641, 'samples': 0.8805970149253731, 'weighted': 0.8308297770510807, 'acc': 0.8805970149253731}**

hidden\_size=[256, 256] 有向

**{'micro': 0.8848614072494669, 'macro': 0.6244131455399061, 'samples': 0.8848614072494669, 'weighted': 0.8345896273161356, 'acc': 0.8848614072494669}**



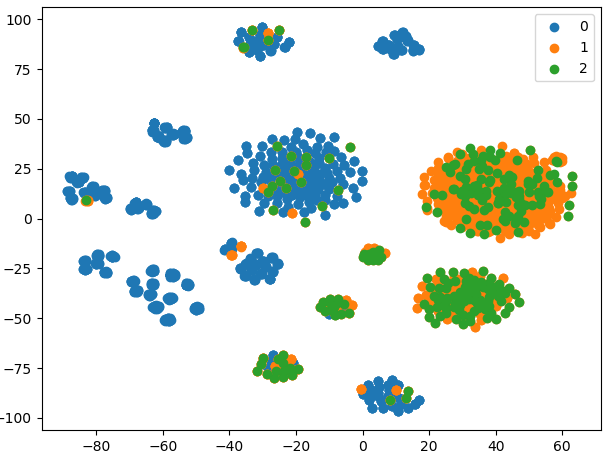
**二分类的情况下**

**{'micro': 0.9957356076759062, 'macro': 0.9957323287471791, 'samples': 0.9957356076759062, 'weighted': 0.9957351032253328, 'acc': 0.9957356076759062}**

**Struc2vec**

**参数 G, 10, 80, workers=1, verbose=40 embedding=128 无向图**

**{'micro': 0.8315565031982942, 'macro': 0.5846214000740487, 'samples': 0.8315565031982942, 'weighted': 0.7814344773372623, 'acc': 0.8315565031982942}**



效果比较明显，用户节点与电话节点基本是散开的，而且用户之间也有不同的区分，学习的较好

**参数 G, 10, 80, workers=1, verbose=40 embedding=3 无向图**

{'micro': 0.8059701492537313, 'macro': 0.5647093315463226, 'samples': 0.8059701492537313, 'weighted': 0.7545320331306634, 'acc': 0.8059701492537313}

虽然只使用了3维嵌入向量，任然可以达到不错的效果，embedding=300

Micro在0.84的水平，但是并不适合进入XGB模型，因为这些嵌入向量的每一维度都有关联，不能当做孤立的部分来看

**参数 G, 10, 80, workers=1, verbose=40 embedding=1 无向图**

{'micro': 0.7526652452025586, 'macro': 0.5297009373093808, 'samples': 0.7526652452025586, 'weighted': 0.7075487130078475, 'acc': 0.7526652452025586}

5 GCN

首先利用struct2vec训练的15维嵌入向量作为节点特征，进入GCN训练

NB\_EPOCH = 400

Test loss: 1.8855440616607666

Test weighted\_loss: 1.8755078315734863

Test accuracy: 0.18532820045948029

从嵌入向量只去一维作为节点特征入模

NB\_EPOCH = 200

Test loss: 0.8818385004997253

Test weighted\_loss: 0.877108633518219

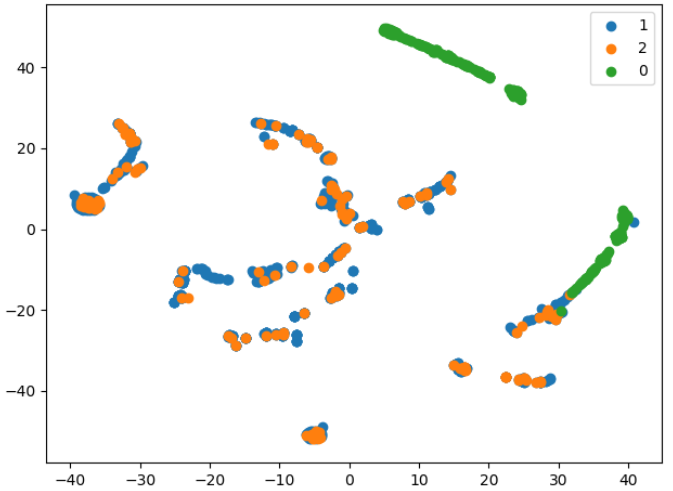
Test accuracy: 0.8378381729125977

NB\_EPOCH = 400

Test loss: 0.7959872484207153

Test weighted\_loss: 0.7894496917724609

Test accuracy: 0.8436296582221985



为了证明是GCN从结构中学到的结果，我们把维的特征都初始化为1

NB\_EPOCH = 400

Test loss: 0.7941292524337769

Test weighted\_loss: 0.7876652479171753

Test accuracy: 0.8436296582221985