数据科学HW3

刘家宁

November 2025

1 实验结果

本次作业的内容是关于对12个被试在62个通道，5个频率上的脑电信号利用DANN进行情绪分类，其中情绪分为积极，中性和消极。本次实验采用跨被试留一交叉验证，即令每个被试成为测试集，其他所有被试成为训练集进行12次单独的训练，最后根据准确率的平均值和方差来判定模型的好坏与泛化能力。首先给出本次实验的结果：

Model	acc_mean	acc_std
Unsupervised Domain Adaptation	0.7768	0.1240
Domain Generalization	0.6532	0.0962
Without Domain classifier	0.5927	0.1270

Table 1: 不同模型在留一交叉验证下的准确率均值和标准差

2 数据预处理

由于留一交叉验证本身就能体现模型的泛化能力，所以这里对于每次训练不划分CV集，只有训练集和测试集。

对于每次训练，保留一个被试作为测试集，其他所有被试成为训练集。

首先对于62×5个通道和频段分别计算mean和std，做z-score归一化。这里不选择对每个人分别进行归一化的原因是这样跨被试差异被消除，每个人都是均值为0方差为1的分布，模型无法学习不同被试之间的统计分布，且在训练集和测试集上使用的标准化公式不一致。

对于CNN，我们使用给定的二维矩阵来把原本的62个通道变为8×9个通道，并在部分无通道的位置补0。这样的好处是模型现在可以利用空间信息在二维上进行卷积。

对于数据增强，我们尝试了三个方式进行数据增强：

- **高斯噪声**：对数据进行0.01倍方差的微扰，这是考虑到标准化采用的是训练集的数据，但测试集的数据在标准化之后未必方差为1。
- **幅值缩放**：考虑到每个人情绪激烈程度不同，我们对幅值进行缩放，倍数在(0.95,1.05)之间。

- **随机dropout**：随机dropout一些通道，尝试泛化模型的学习能力。

3 模型参数

尽管在数据方面我们尽量做了一些增强，但是数据量实在还是太少，我们发现依然存在较强的过拟合问题，如下图所示，由于DANN本质上是一个对抗的网络，所以在过拟合之后在测试集上的准确率会有很大的波动，这是我们不希望看到的。

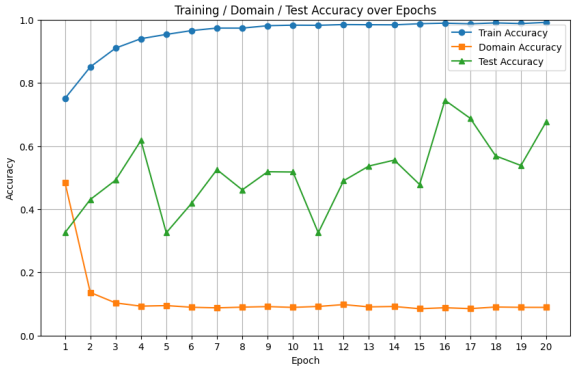

Figure 1: 第三个受试的模型在域泛化上的表现

为了解决这个问题，我们采用了一些其他的lambda scheduler。一般常见的lambda scheduler是：

$$\lambda = \frac{2}{1 + e^{-10p}} - 1$$

其中p为当前已经训练的epoch的百分比。为了解决这个问题，我们尝试了几种lambda scheduler，如线性的$\lambda = p$, 多项式的$\lambda = p^2$，还有保守的$\lambda = \frac{2}{1+e^{-5p}} - 1$，但是效果都不是很好，没有看出明显的改善。最后比较好的解决方案是保持scheduler不变，但是前两个epoch不对抗，同时降低学习率$lr = 1e - 4$，并给整个λ乘上0.3的缩放系数。

同时我们使用了早停策略。观察到大约5个epoch之后训练集准确率已经到达95%，我们在第5个epoch之后保留最优的结果作为最终的模型。

最终的模型结构如下：

```python
class FeatureExtractor(nn.Module):
    def __init__(self):
        super().__init__()

        self.conv = nn.Sequential(
            nn.Conv2d(5, 32, kernel_size=3, padding=1),  # (32,8,9)
            nn.BatchNorm2d(32),
            nn.ReLU(),

            nn.Conv2d(32, 64, kernel_size=3, padding=1), # (64,8,9)
            nn.BatchNorm2d(64),
            nn.ReLU(),

            nn.MaxPool2d(2),  # (64,4,4)

            nn.Conv2d(64, 128, kernel_size=3, padding=1), # (128,4,4)
            nn.BatchNorm2d(128),
            nn.ReLU(),

            nn.AdaptiveAvgPool2d((1, 1))  # (128,1,1)
        )

    def forward(self, x):
        x = self.conv(x)
        x = x.view(x.size(0), -1)  # (batch, 128)
        return x
```

Figure 2: 特征提取器/卷积层结构

```python
class LabelClassifier(nn.Module):
    def __init__(self, num_classes=3):
        super().__init__()
        self.fc = nn.Sequential(
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(64, num_classes)
        )

    def forward(self, x):
        return self.fc(x)
```

Figure 3: 标签分类器结构（全连接层+dropout）

```python
class DomainClassifier(nn.Module):
    def __init__(self, num_domains=12):
        super().__init__()
        self.fc = nn.Sequential(
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Linear(64, num_domains)
        )

    def forward(self, x, lambd):
        x = grad_reverse(x, lambd)
        return self.fc(x)
```

Figure 4: 域分类器（全连接层）

调整之后，我们的表现如下。可以看到整体的acc上升了不少，虽然波动仍然存在，但是整体是在一个较高的范围波动。

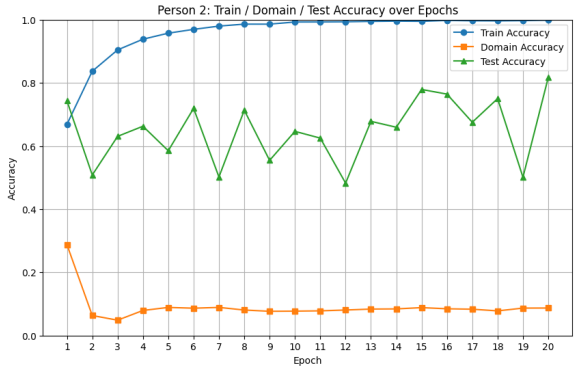

Figure 5: 第三个受试的在新模型上域泛化的表现

4 域适应 Unsupervised Domain Adaption

以下给出域适应留一交叉验证的结果:

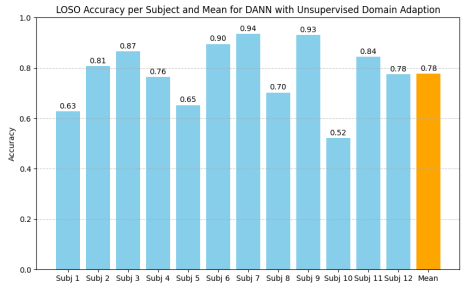

Figure 6: 域适应留一交叉验证结果

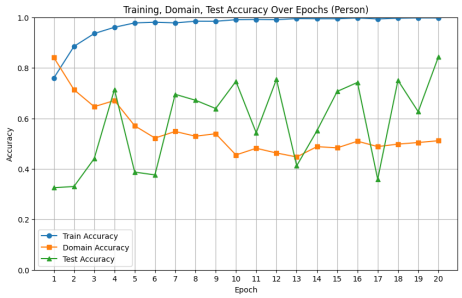

Figure 7: 受试11在域适应上的训练结果

在域适应中，我们的域判别器只需要判别是否是训练集的样本，其本质是一个二分类器。可以看到准确率达到了惊人的0.78，这是因为域适应可以看到测试集的无标签样本，并且适应出一个无法区分训练集和测试集的映射。

这里同时给出受试11在该模型上的训练结果，我们可以发现，最后域分类准确率收敛到了约0.5，不过这个模型仍然有较大的波动，偶尔会出现一些负面的迭代，但是总体的准确率都非常高。

5 域泛化 Domain Generalization

以下给出域泛化留一交叉验证的结果:

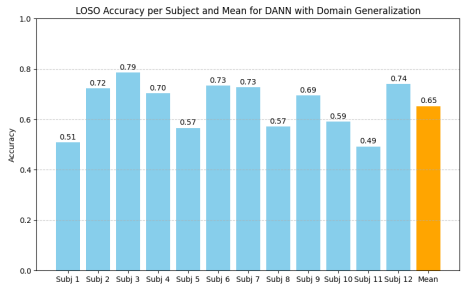

Figure 8: 域泛化留一交叉验证结果

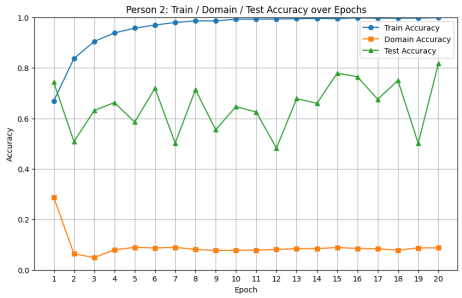

Figure 9: 受试3的在域泛化上的表现

域泛化和域适应的区别有两个：（1）没有测试集的信息；（2）每个人单独为一类，而不是分成训练集和测试集两类。

可以看到比上一次作业中的分类结果好好许多，准确率到了0.65，但是还是不如域适应。原本分类比较准确的人依然保持了0.7左右的准确率，但是部分原本表现的很差的人现在提高了0.1-0.2的准确率。

这里训练的时候需要注意要同时训练标签分类器和域分类器，不能分开来训练，否则结果会是更加波动的，甚至欠拟合的。我们每个训练的batch都包含两种分类，这样我们迭代会更加平滑。同时，我们需要注意，检验DANN跑起来的方式是域分类正确率结果接近随机分类（在该情况下约为0.09（1/11）），而最终我们的正确率多为0.08-0.09，符合这个条件，说明我们的DANN是正确的。

6 无域分类

本质上我们设计的DANN去掉域分类之后就只是一个CNN了。可以看到这里的结果是远差于加上域分类之后的结果的。

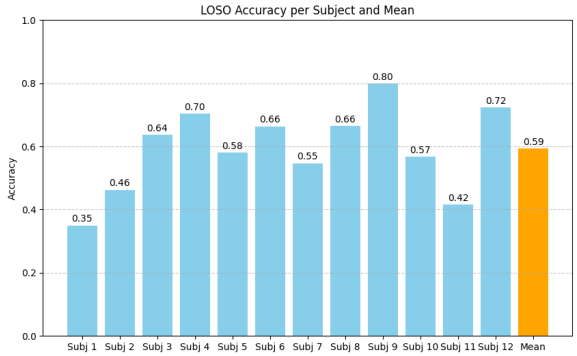

Figure 10: 去掉域分类之后的结果

7 TSNE可视化

我们统一以第三个被试为例。

7.1 域适应

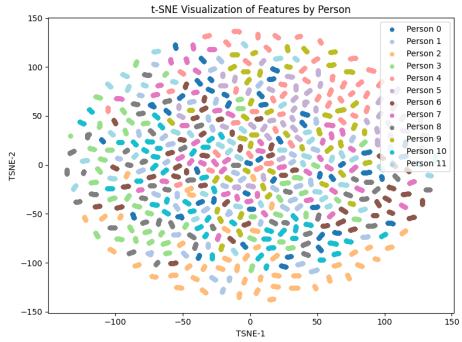

Figure 11: 域适应整体分类结果

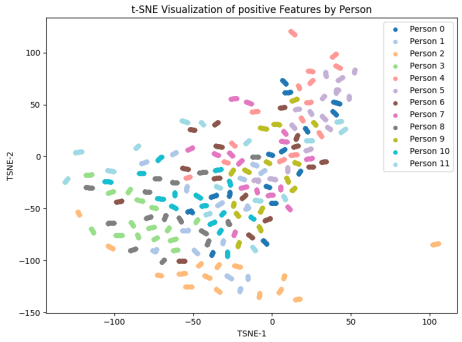

Figure 12: 域适应积极分类结果

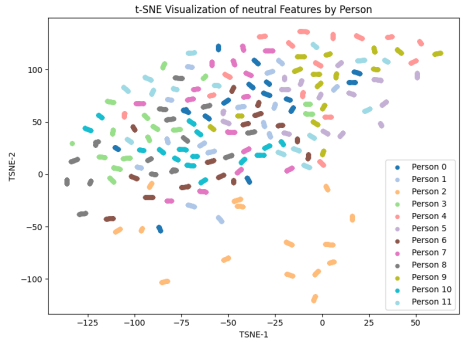

Figure 13: 域适应中性分类结果

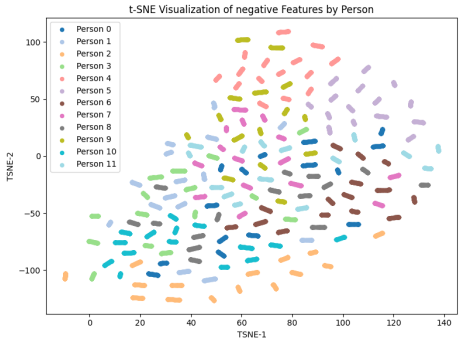

Figure 14: 域适应消极分类结果

由于tsne是无监督的，所以我们的降维结果可以表现出我们特征向量聚类是否成功。可以看到积极和消极基本上都是和其他人的情绪差不多，只有中性聚类的比较失败。

7.2 域泛化

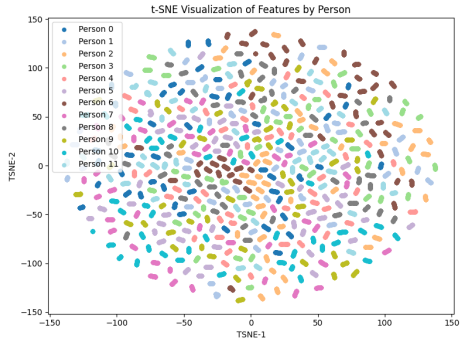

Figure 15: 域泛化整体分类结果

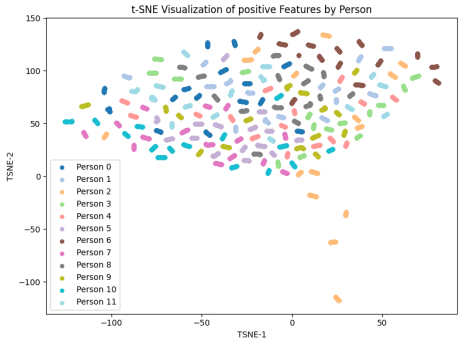

Figure 16: 域泛化积极分类结果

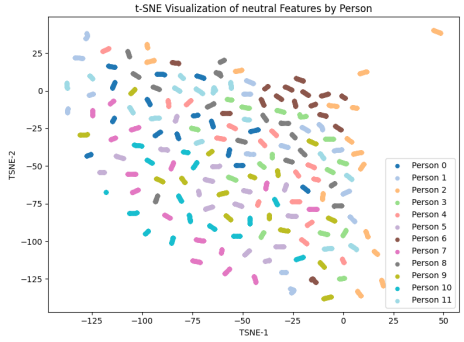

Figure 17: 域泛化中性分类结果

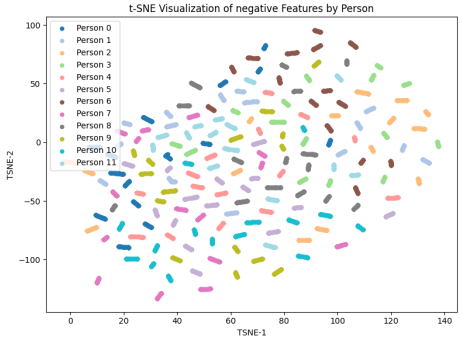

Figure 18: 域泛化消极分类结果

同理，我们发现域泛化中积极的聚类有点小问题，其他基本都是聚在一起的。

7.3 无域分类器

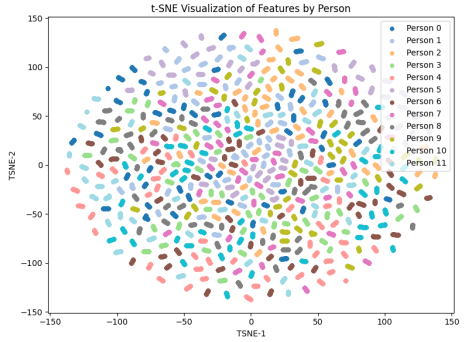

Figure 19: 无域分类器整体分类结果

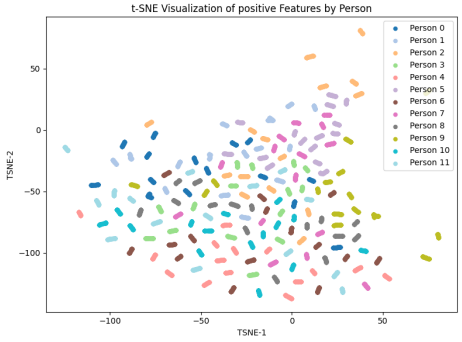

Figure 20: 无域分类器积极分类结果

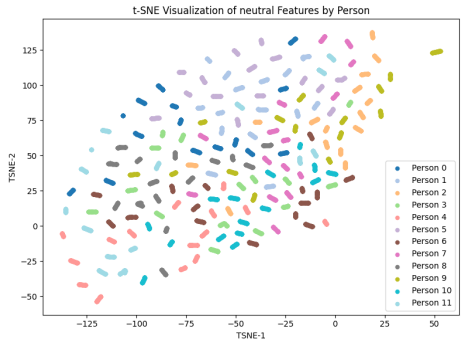

Figure 21: 无域分类器中性分类结果

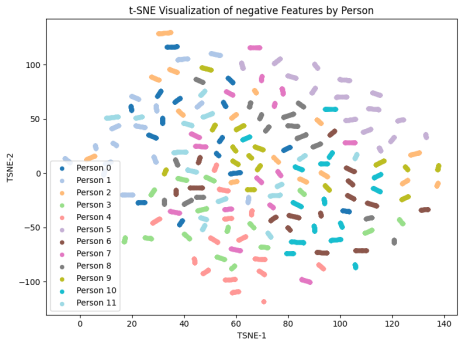

Figure 22: 无域分类器消极分类结果

我们可以发现，无域分类器聚出来的积极和消极的结果相对于前面的都要散一些。这说明我们的域训练是有效地去除了不同域之间的不同。

8 总结

DANN相对于没有域分类的版本准确率有明显的提升，但是缺点是会有很大的波动，因为DANN本质上是一个类似minmax的过程，是一个对抗的网路，所以在测试集上的表现的波动是正常的现象，这里我们采用增加数据、调整超参、早停等策略去尽量保留更优的解，最后相对于没有域分类的版本准确率上升了0.2。