Research Proposal

王梓奕 李一玄 刘俊豪 姚嘉浩 黄肖炜

背景

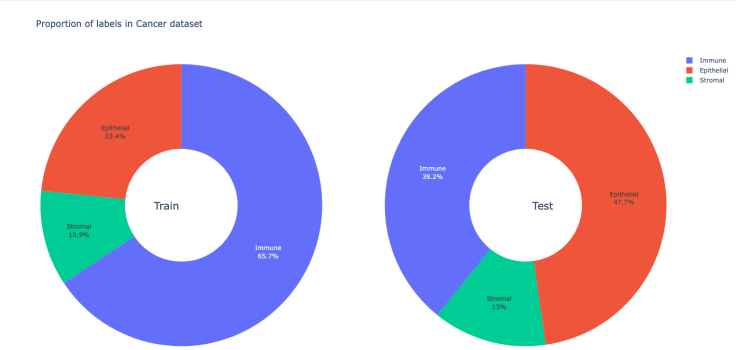
1. 协变量偏移
   在假设标记函数不变的前提下，特征分布的变化导致的训练集和测试集存在本质上区别的分布偏移，称为协变量偏移。
2. 标签偏移
   标签在训练集和测试集中出现的频率不同，例如，在训练集中，很少出现细胞类别为 q 的样本，但在测试集中，同时存在细胞类别为 p 和细胞类别 q 为的样本，其中不变的是不同基因的表现。
3. 概念偏移
   除上述两种偏移之外，还存在标签本身概念发生变化的偏移，例如，对三大类癌细胞的定义发生变化。

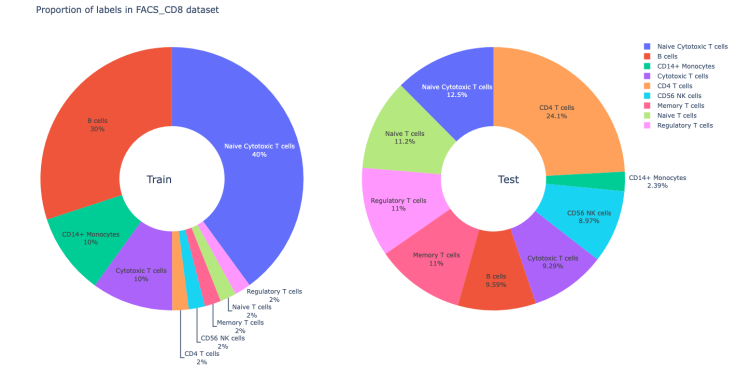
意义

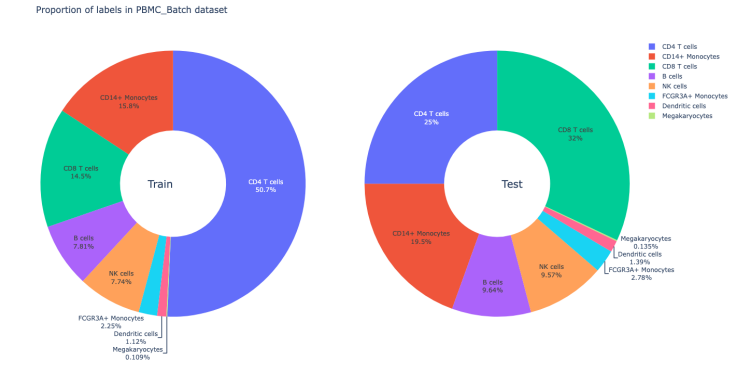
1. 提高模型在实际应用中的泛化能力和性能。
2. 提高模型在不同标签分布下的预测准确性。
3. 通过引入假设检验和超参数等方法，可以进一步解决不同类型的偏移问题，并提高模型的性能，改进预测方法。

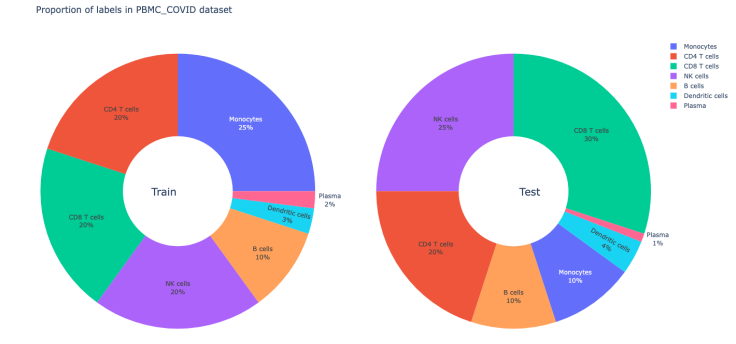
探索性分析

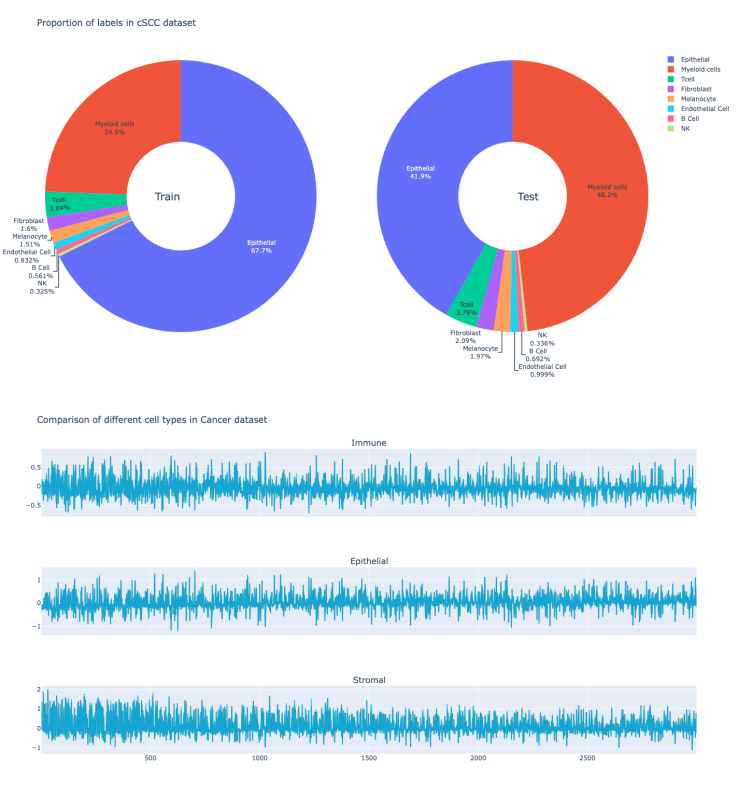
- 统计五组训练集和测试集中各类别细胞的频率，绘制饼图，可看出各数据集均涉及协变量偏移











- 统计五组训练集和测试集中各类别细胞各基因的均值，绘制折线图，深色代表训练集与测试集的均值相近，浅蓝色与红色则代表该基因下均值存在差异，结果发现在五组数据集中差异均不显著，给出`Cancer`数据集的折线图作为参考

结论

五组数据集主要问题在于较为严重的协变量偏移，因此针对该现象训练模型并拟合

思路

> *How to resist distribution shift*

神经网络

- 调用 **PyTorch** 的 API 建立模型

> ### *Architecture of Neural Network*
>
> > #### *Input and Embedding layers*
> >
> > > ***Linear*** *(*`in_features`*=n_features,* `out_features`*=n_features,* `bias`*=True)*
> >
> > > ***Linear*** *(*`in_features`*=n_features,* `out_features`*=300,* `bias`*=True)*
> >
> > #### *MLP*
> >
> > > ***Linear*** *(*`in_features`*=300,* `out_features`*=_inter_features,* `bias`*=True)*
> >
> > > ***BatchNorm1d*** *(*`batch_size`*=_inter_features,* `eps`*=1e-05,* `momentum`*=0.1,* `affine`*=True,* `track_running_stats`*=True)*
> > > ***ELU*** *(*`alpha`*=1.0)*
> > > ***Dropout*** *(*`p`*=0.5,* `inplace`*=False)*

> > > ***Linear*** *(`in_features`=_inter_features, `out_features`=_inter_features, `bias`=True)*
>
> > > ***BatchNorm1d*** *(`batch_size`=_inter_features, `eps`=1e-05, `momentum`=0.1, `affine`=True, `track_running_stats`=True)*
> > > ***ELU*** *(`alpha`=1.0)*
> > > ***Dropout*** *(`p`=0.5, `inplace`=False)*
>
> > > ***Linear*** *(`in_features`=_inter_features, `out_features`=n_labels, `bias`=True)*
>
> > > ***Softmax*** *(`dim`=1)*
>
> ### *Optimizer*
>
> > > ***Adam*** *(`lr`=LR, `betas`=(BETA1, BETA2), `eps`=EPS)*
>
> ### *LossFuntion*
>
> > > ***CrossEntropyLoss***
>
> ### *Learning-rate Scheduler*
>
> > > ***ReduceLROnPlateau*** *("min", `patience`=PATIENCE, `threshold`=THRESHOLD)*

- `_inter_features` = $\frac{2}{3}$ (`in_features` + `out_features`)
- `nEpoch` = 10

抽样

> > *For BGD @property@abstractmethod*

方案 1

使用一种概率分布 (默认为标准高斯分布) 为训练集中的所有样本生成一个概率 (权重)，基于这些权重抽取样本

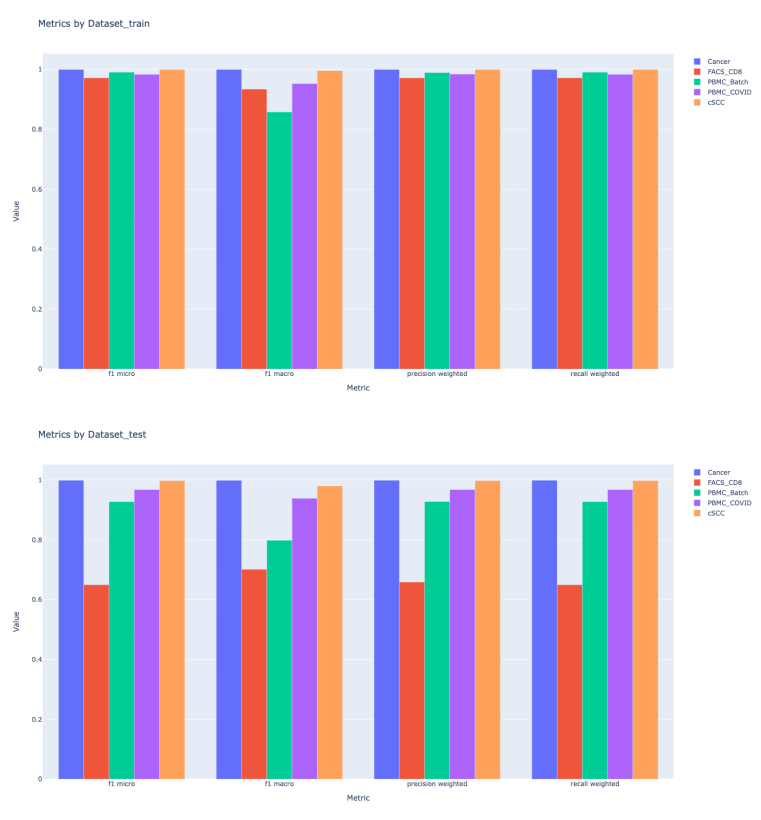
```
def sample(self, distribution: scipy.stats.rv_continuous = scipy.stats.uniform, *args
    ) -> tuple[torch.Tensor, torch.Tensor]:
    """Use the generated weights to sample the data

    Parameters
    ----------
    distribution : scipy.stats.rv_continuous, optional
        The distribution to generate each probability from, by default scipy.stats.uniform

    Returns
    -------
    tuple[torch.Tensor, torch.Tensor]
        X_Batch, y_Batch
    """
    self.weights = prosGenerator(distribution=distribution, size=self.size, *args)
    self.choices = np.random.choice(
```

```
        range(self.size), self.batch_size, False,
self.weights
    )
    return featureLabelSplit(self.data[self.choices])
```

结果



- 在 `FACS_CD8` 与 `PBMC_Batch` 上表现很差

方案 2

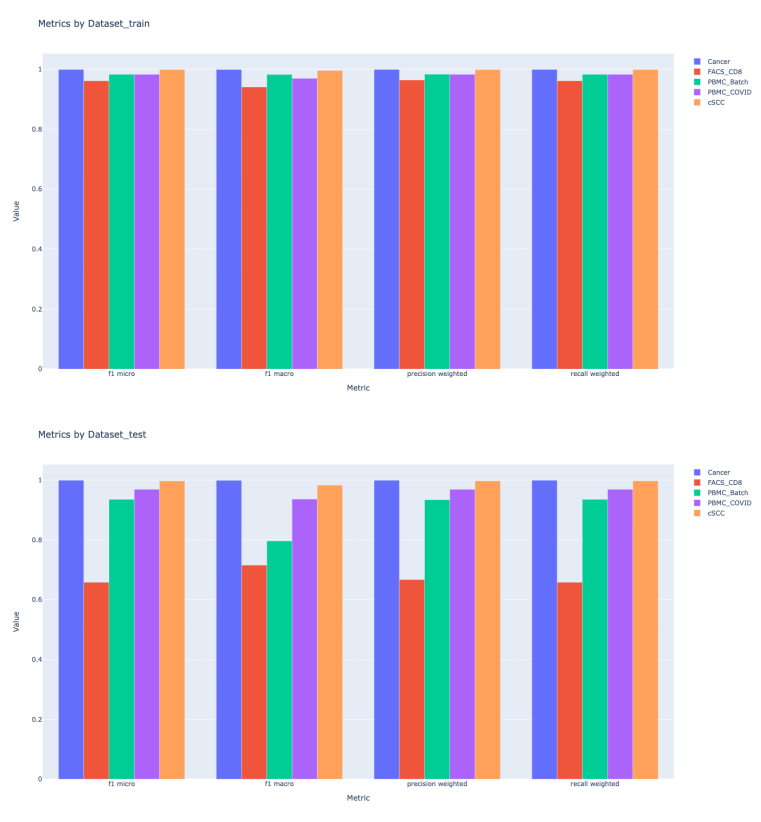
将要抽取的样本数随机分为 n 组，其中 n 为数据集中不同标签个数，每组中的样本数对应于各个标签中采用 Bootstrap 采样的样本数

```
  def sample(self) -> tuple[torch.Tensor, torch.Tensor]:
      """Utilize the generated counts to sample the data by
Bootstrap

      Returns
      -------
      tuple[torch.Tensor, torch.Tensor]
          X_Batch, y_Batch
      """
      nums = self.getNum  # A property returns the nums to be
sampled for each label
      for i in range(len(self.changeIndexes) - 1):
          self.choices += list(
              np.random.choice(
                  range(self.changeIndexes[i],
self.changeIndexes[i + 1]),
                  nums[i],
                  True,
              )
          )
```

```
18          return featureLabelSplit(self.data[self.choices])
```

结果



- 相比方案 1 有提升但在 `FACS_CD8` 与 `PBMC_Batch` 上表现依然很差

方案 3

利用训练集中各标签下数据的均值及标准差伪造样本并植入到训练集中

```
1   def sample(self) -> NotImplemented
2       return NotImplemented
```

结果

TODOS

1. 实现方案 3
2. 改善研究背景与研究意义
3. 编写函数删除所有 .html 文件

未来

1. 混合各个抽样方案
2. 对表现较好的数据集部署更简单的模型以节省成本