

实验一:基于 UCI 数据集的 SVM 与神经网络多模型构建

1. 实验目的与要求

本实验基于两类典型的 UCI 数据集,通过系统地改变模型参数,构建多个 SVM 与神经网络模型,对比不同配置下的泛化性能,并保存模型及其在训练集、验证集、测试集上的预测结果.具体目标如下:

- 1.熟悉对真实 UCI 数据集进行预处理、划分训练/验证/测试集的基本流程.
- 2.通过改变 SVM 与神经网络的超参数(核函数、正则化系数、网络深度、学习率等),构建共 20 个不同模型(学生数据集 10 个 SVM,NHANES 数据集 10 个 ANN),体会模型容量与泛化能力之间的关系.
- 3.将训练好的模型保存到本地磁盘,并分别保存其在训练集、验证集、测试集上的预测标签,为后续的模型融合实验(朴素贝叶斯和 DS 证据理论)提供基模型.
- 4.基于训练过程中生成的两个汇总结果文件 student_svm_results_with_val.csv 与 nhanes_ann_results_with_val.csv 做简单统计和可视化分析,对实验现象给出合理解释.

2. 数据集介绍

2.1 学生表现预测数据集

学生表现预测数据集(Predict Students' Dropout and Academic Success)来自 UCI 机器学习库.

主要特点如下:

样本量约 4424 条,每条记录对应一名学生.

特征数 36 个,包含入学成绩、学籍信息、人口学特征及社会经济背景等.

目标变量为学生在正常学制结束时的状态,属于三分类任务(退学 Dropout/在读 Enrolled/毕业 Graduate).

数据集中不含缺失值,原始数据已完成基本清洗.

在本实验中,我们从原始 CSV 文件 data.csv 中读取数据,提取所有数值型特征作为 SVM 的输入,以 Target 列作为分类标签.

2.2 NHANES 2013–2014 年龄预测子集

NHANES(National Health and Nutrition Examination Survey)由美国 CDC 组织实施,是一项大规模健康与营养调查.本实验使用其 2013–2014 年度的 Age Prediction 子集:

样本量约 6287 条,每条记录对应一名受访者.

选取 7 个与健康状况高度相关的特征:性别、体力活动情况、体重指数 BMI、空腹血糖、糖尿病状态、口服葡萄糖耐量、胰岛素水平等.

原 UCI 任务为二分类(senior/non-senior),本实验根据提供的数据文件

NHANES_age_prediction.csv,将连续变量 RIDAGEYR(实际年龄)作为回归目标,构建年龄预测模型.

3. 实验方法与实现

本实验仅涉及两个核心程序文件:svm.py(学生数据集+SVM)与 nn.py(NHANES 数据集+ANN).下面分别说明其关键实现.

3.1 学生表现数据集:SVM 多模型实验(svm.py)

3.1.1 数据预处理与集划分

对学生数据集的处理流程为:

用 pandas 读取 data.csv;

以 Target 作为标签 y,其余数值型字段作为特征 X;

使用 LabelEncoder 对三分类标签进行整数编码;

使用 StandardScaler 对特征做标准化;

按 80%/10%/10%的比例划分训练集、验证集与测试集,并在划分时使用 stratify 保持类别分布一致.

代码 1-1 数据预处理与划分(节选)

```
01 data = pd.read_csv('data.csv', sep=',')
02 y = data['Target'].values
03 X = data.select_dtypes(include=[np.number]).copy()
04 if 'Target' in X.columns:
05     X = X.drop(columns=['Target'])
06
07 label_encoder = LabelEncoder()
08 y_encoded = label_encoder.fit_transform(y)
09
10 scaler = StandardScaler()
11 X_scaled = scaler.fit_transform(X)
12
13 X_temp, X_test, y_temp, y_test = train_test_split(
14     X_scaled, y_encoded, test_size=0.1,
15     random_state=42, stratify=y_encoded
16 )
17 X_train, X_val, y_train, y_val = train_test_split(
18     X_temp, y_temp, test_size=0.1111,
19     random_state=42, stratify=y_temp
20 )
```

同时,程序创建两个目录 student_svm_models 与 student_svm_predictions,用于保存模型文件和各划分上的预测结果 CSV.

3.1.2 SVM 超参数配置与模型集合

根据实验要求,需要"通过改变参数生成多个模型".在学生数据集上,设计了 10 组 SVM 配置,主要改变核函数类型、正则化系数 C、核参数 gamma 等:

线性核:C $\in\{0.1, 1.0, 10.0\}$;

RBF 核:C $\in\{1.0, 10.0\}$,gamma 取 scale 或固定值 0.01;

多项式核:二次与三次多项式,多种 C 和 gamma 组合;

Sigmoid 核:作为对比模型.

代码 1-2 SVM 模型配置(节选)

```
01 student_svm_configs =
02     {'kernel': 'linear', 'C': 0.1},
03     {'kernel': 'linear', 'C': 1.0},
04     {'kernel': 'linear', 'C': 10.0},
05     {'kernel': 'rbf',   'C': 1.0, 'gamma': 'scale'},
06     {'kernel': 'rbf',   'C': 10.0, 'gamma': 'scale'},
07     {'kernel': 'rbf',   'C': 1.0, 'gamma': 0.01},
08     {'kernel': 'poly',  'C': 1.0, 'degree': 2, 'gamma': 'scale'},
09     {'kernel': 'poly',  'C': 1.0, 'degree': 3, 'gamma': 'scale'},
10     {'kernel': 'poly',  'C': 10.0, 'degree': 2, 'gamma': 'scale'},
```

```
11     {'kernel': 'sigmoid', 'C': 1.0, 'gamma': 'scale'},
12 ]
```

每一组配置对应一个 SVM 模型,总计 10 个模型.

3.1.3 训练、预测结果保存与汇总

在训练阶段,程序对每一组配置循环执行以下步骤:

- 1.在训练集上拟合 SVM 模型;
- 2.在训练/验证/测试集分别预测标签,并计算准确率;
- 3.将真实标签与预测标签通过 `LabelEncoder.inverse_transform` 还原为原始类别值,分别保存到 `student_svm_predictions` 目录下的 CSV 文件中;
- 4.将每个模型的配置及其三种划分上的准确率记录到 `svm_results` 列表.

最终,程序将 10 个模型的结果汇总成一个 DataFrame,按验证集准确率 `val_accuracy` 从高到低排序,保存为 `student_svm_results_with_val.csv`,并在终端打印.

代码 1-3 训练循环与结果汇总(节选)

```
01 def save_split_predictions(base_name, split_name, y_true, y_pred, encoder):
02     y_true_labels = encoder.inverse_transform(y_true)
03     y_pred_labels = encoder.inverse_transform(y_pred)
04     df = pd.DataFrame({
05         'y_true_encoded': y_true,
06         'y_pred_encoded': y_pred,
07         'y_true': y_true_labels,
08         'y_pred': y_pred_labels
09     })
10     df.to_csv(
11         os.path.join("student_svm_predictions",
12                     f'{base_name}_{split_name}_predictions.csv'),
13         index=False
14     )
15
16 for idx, cfg in enumerate(student_svm_configs):
17     model = SVC(**cfg)
18     model.fit(X_train, y_train)
19
20     y_train_pred = model.predict(X_train)
21     y_val_pred = model.predict(X_val)
22     y_test_pred = model.predict(X_test)
23
24     train_acc = accuracy_score(y_train, y_train_pred)
25     val_acc = accuracy_score(y_val, y_val_pred)
26     test_acc = accuracy_score(y_test, y_test_pred)
27
28     base_name = f"student_svm_model_{idx + 1}"
29     save_split_predictions(base_name, "train", y_train, y_train_pred, label_encoder)
30     save_split_predictions(base_name, "val", y_val, y_val_pred, label_encoder)
31     save_split_predictions(base_name, "test", y_test, y_test_pred, label_encoder)
32
33     model_path = os.path.join("student_svm_models", f'{base_name}.joblib')
34     joblib.dump(model, model_path)
35
36     svm_results.append({
37         'model_index': idx + 1,
38         'config': cfg,
```

```
39     'train_accuracy': train_acc,
40     'val_accuracy': val_acc,
41     'test_accuracy': test_acc,
42     'model_path': model_path
43   })
44
45 svm_results_df = pd.DataFrame(svm_results)
46 svm_results_df = svm_results_df.sort_values(by='val_accuracy', ascending=False)
47 svm_results_df.to_csv("student_svm_results_with_val.csv", index=False)
```

3.2 NHANES 年龄预测 ANN 多模型实验(nn.py)

3.2.1 特征选择与数据划分

在 NHANES 数据集上,实验使用以下 7 个特征作为输入:

RIAGENDR,PAQ605,BMXBMI,LBXGLU,DIQ010,LBXGLT,LBXIN.

标签为连续年龄 RIDAGEYR.

处理流程与学生数据集类似:

- 1.读取 NHANES_age_prediction.csv;
- 2.选取上述特征及年龄标签;
- 3.对特征做标准化;
- 4.按 80%/10%/10%划分训练、验证、测试集;
- 5.使用 TensorDataset 和 DataLoader 将数据封装成 PyTorch 批量输入格式.

代码 1-4 NHANES 数据预处理与划分(节选)

```
01 nhanes_df = pd.read_csv('NHANES_age_prediction.csv')
02 feature_cols = ['RIAGENDR', 'PAQ605', 'BMXBMI',
03                   'LBXGLU', 'DIQ010', 'LBXGLT', 'LBXIN']
04 X = nhanes_df[feature_cols].copy()
05 y = nhanes_df['RIDAGEYR'].values.astype(np.float32)
06
07 scaler = StandardScaler()
08 X_scaled = scaler.fit_transform(X)
09
10 X_temp, X_test, y_temp, y_test = train_test_split(
11     X_scaled, y, test_size=0.1, random_state=42
12 )
13 X_train, X_val, y_train, y_val = train_test_split(
14     X_temp, y_temp, test_size=0.1111, random_state=42
15 )
16
17 X_train_tensor = torch.tensor(X_train, dtype=torch.float32)
18 y_train_tensor = torch.tensor(y_train.reshape(-1, 1))
19 # 同理构建 X_val_tensor / X_test_tensor
```

程序同样创建 nhanes_ann_models 与 nhanes_ann_predictions 目录,以保存网络权重和预测结果.

3.2.2 神经网络结构与超参数设计

实验采用多层全连接网络 AgePredictionNet, 每层结构为 Linear+ReLU+Dropout,最后一层输出 1 维连续值,用于回归.

代码 1-5 网络结构定义(核心代码)

```
01 class AgePredictionNet(nn.Module):
```

```

02     def __init__(self, input_dim, hidden_dims, dropout_rate):
03         super(AgePredictionNet, self).__init__()
04         layers = []
05         prev_dim = input_dim
06         for h in hidden_dims:
07             layers.append(nn.Linear(prev_dim, h))
08             layers.append(nn.ReLU())
09             layers.append(nn.Dropout(dropout_rate))
10             prev_dim = h
11         layers.append(nn.Linear(prev_dim, 1)) # 回归输出
12         self.network = nn.Sequential(*layers)
13
14     def forward(self, x):
15         return self.network(x)

```

为了满足"通过改变参数生成多个模型"的要求,实验设计了10组不同的网络配置,主要在隐藏层维度、层数、学习率和Dropout上进行组合,例如:

- 简单单隐层小网络:[32];
- 深层网络:[64,32]、[128,64,32];
- 宽网络:[128]、[256,128];
- 大学习率/小学习率、不同 Dropout 比例等.

代码 1-6 ANN 配置列表(节选)

```

1 nhanes_ann_configs = [
2     {"name": "NHANES_NN_1_Simple", "hidden_dims": [32],           "lr": 0.001, "dropout": 0.1},
3     {"name": "NHANES_NN_2_Deep",   "hidden_dims": [64, 32],        "lr": 0.001, "dropout": 0.2},
4     {"name": "NHANES_NN_3_Wide",   "hidden_dims": [128],          "lr": 0.01,  "dropout": 0.3},
5     {"name": "NHANES_NN_4_DeepWide", "hidden_dims": [128, 64, 32], "lr": 0.0001, "dropout": 0.2},
6 ...
7     {"name": "NHANES_NN_10_VeryWide", "hidden_dims": [256, 128],    "lr": 0.001, "dropout": 0.3},
8 ]

```

3.2.3 训练策略、预测保存与结果汇总

损失函数:均方误差 MSELoss ;

优化器 Adam;

每个模型训练 200 个 epoch,每个 epoch 对训练集和验证集分别计算损失和验证集 RMSE;

在验证集 RMSE 最小处保存最佳权重;

利用最佳权重在训练/验证/测试集上分别评估 RMSE、MAE、 R^2 ,并保存预测结果;

将 10 个模型的指标整理成 `nhanes_ann_results_with_val.csv`,按验证集 RMSE 从小到大排序.

代码 1-7 训练与结果汇总(节选)

```

01 criterion_reg = nn.MSELoss()
02 num_epochs = 200
03
04 def rmse(y_true, y_pred):
05     return np.sqrt(mean_squared_error(y_true, y_pred))
06
07 def evaluate_regression(model, X_tensor, y_tensor, criterion):

```

```
08 model.eval()
09 with torch.no_grad():
10     outputs = model(X_tensor)
11     loss = criterion(outputs, y_tensor)
12     y_true = y_tensor.cpu().numpy().flatten()
13     y_pred = outputs.cpu().numpy().flatten()
14     return (y_true, y_pred,
15             loss.item(),
16             rmse(y_true, y_pred),
17             mean_absolute_error(y_true, y_pred),
18             r2_score(y_true, y_pred))
19
20 nhanes_ann_results = []
21
22 for idx, cfg in enumerate(nhanes_ann_configs):
23     model = AgePredictionNet(input_dim, cfg["hidden_dims"], cfg["dropout"])
24     optimizer = torch.optim.Adam(model.parameters(), lr=cfg["lr"])
25
26     best_val_rmse = float("inf")
27     best_state_dict = None
28
29     for epoch in range(num_epochs):
30         model.train()
31         for xb, yb in train_loader:
32             optimizer.zero_grad()
33             outputs = model(xb)
34             loss = criterion_reg(outputs, yb)
35             loss.backward()
36             optimizer.step()
37
38             _, _, _, val_rmse_epoch, _, _ = evaluate_regression(
39                 model, X_val_tensor, y_val_tensor, criterion_reg
40             )
41             if val_rmse_epoch < best_val_rmse:
42                 best_val_rmse = val_rmse_epoch
43                 best_state_dict = model.state_dict()
44
45     if best_state_dict is not None:
46         model.load_state_dict(best_state_dict)
47
48     # 评估并保存预测
49     y_train_true, y_train_pred, train_loss, train_rmse, train_mae, train_r2 = \
50         evaluate_regression(model, X_train_tensor, y_train_tensor, criterion_reg)
51     # 同理计算 y_val_true / y_test_true 等
52
53     base_name = f"nhanes_ann_model_{idx + 1}"
54     # 保存 train / val / test 预测到 nhanes_ann_predictions 目录
55     # 保存模型到 nhanes_ann_models 目录
56
57     nhanes_ann_results.append({
58         'model_index': idx + 1,
59         'name': cfg["name"],
60         'hidden_dims': cfg["hidden_dims"],
61         'dropout': cfg["dropout"],
62         'lr': cfg["lr"],
63         'train_rmse': train_rmse,
64         'val_rmse': val_rmse,
65         'test_rmse': test_rmse,
66         'train_mae': train_mae,
67         'val_mae': val_mae,
```

```
68     'test_mae': test_mae,
69     'train_r2': train_r2,
70     'val_r2': val_r2,
71     'test_r2': test_r2,
72     'model_path': model_path
73 }
74
75 nhanes_ann_results_df = pd.DataFrame(nhanes_ann_results)
76 nhanes_ann_results_df = nhanes_ann_results_df.sort_values(by='val_rmse', ascending=True)
77 nhanes_ann_results_df.to_csv("nhanes_ann_results_with_val.csv", index=False)
```

4. 实验结果与可视化分析

本节所有数值来自你目前运行得到的两个汇总 CSV 文件.

4.1 学生表现数据集:10 个 SVM 模型的比较

从 student_svm_results_with_val.csv 可以看到:

验证集准确率最高的模型为模型 1(线性核,C=0.1),其训练集准确率约为 0.776,验证集准确率约为 0.745,测试集准确率约为 0.779.

验证集表现较好的若干模型(模型 1、4、2、3)在测试集上也取得了相近的性能,说明在本划分下验证集能够较好地反映泛化能力.

部分 RBF 和多项式核模型(如模型 5)在训练集上准确率明显偏高(接近 0.92),但验证/测试准确率下降,存在一定过拟合倾向.

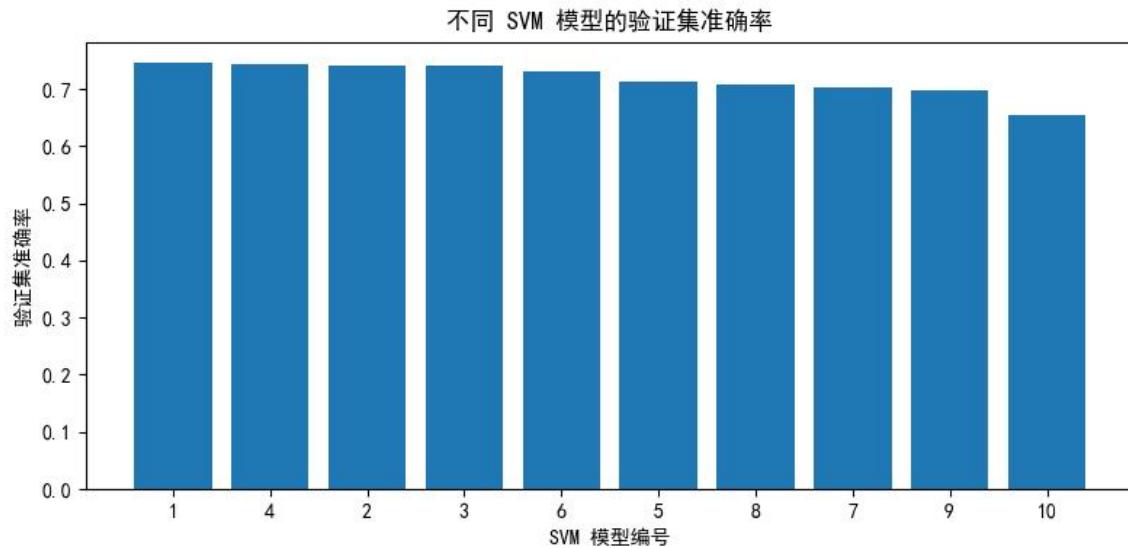


图 1-1 不同 SVM 模型的验证集准确率

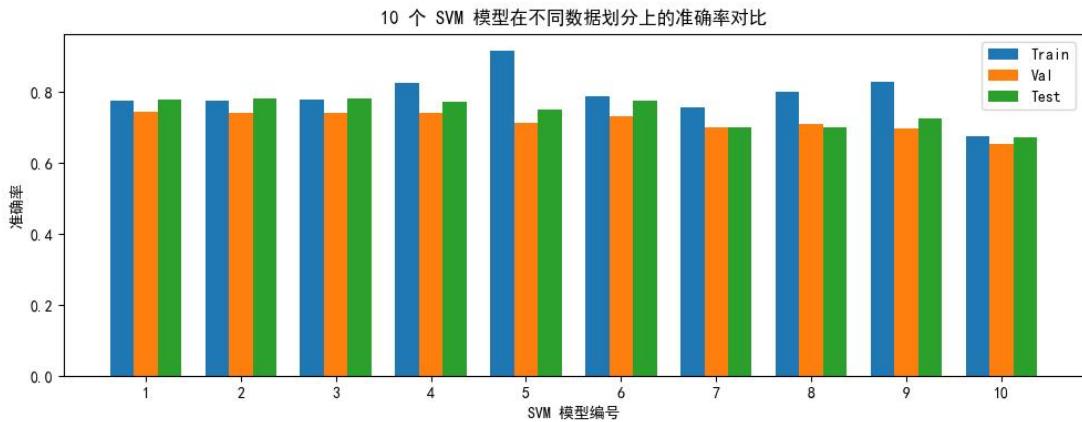


图 1-2 Train/Val/Test 准确率对比

通过图 2 可以看到:部分 RBF、多项式核模型的训练准确率远高于验证/测试准确率,而线性核模型在三种划分上的表现较为接近,泛化更稳定.

4.2 NHANES 年龄预测:10 个 ANN 模型的比较

在 nhanes_ann_results_with_val.csv 中,按照验证集 RMSE 排序后的结果显示:
 验证集 RMSE 最小的模型为模型 10(hidden_dims=[256,128],lr=0.001,dropout=0.3):
 训练 RMSE≈15.70,验证 RMSE≈15.32,测试 RMSE≈15.85;
 相应的 R^2 指标在验证集上约为 0.384,在测试集上约为 0.359,说明模型能够解释大约 35%–38% 的年龄方差.
 随着网络规模减小或学习率设置不当(例如模型 5,学习率 0.0001;模型 9,窄网络 [16,8]),验证和测试 RMSE 明显增大, R^2 降低,体现了欠拟合或训练不充分的问题.
 一些较大的网络(如模型 8:非常深的结构)在训练集和验证集上 RMSE 均较小,但并未显著优于模型 10,说明在目前数据规模下过度加深网络带来的收益有限.

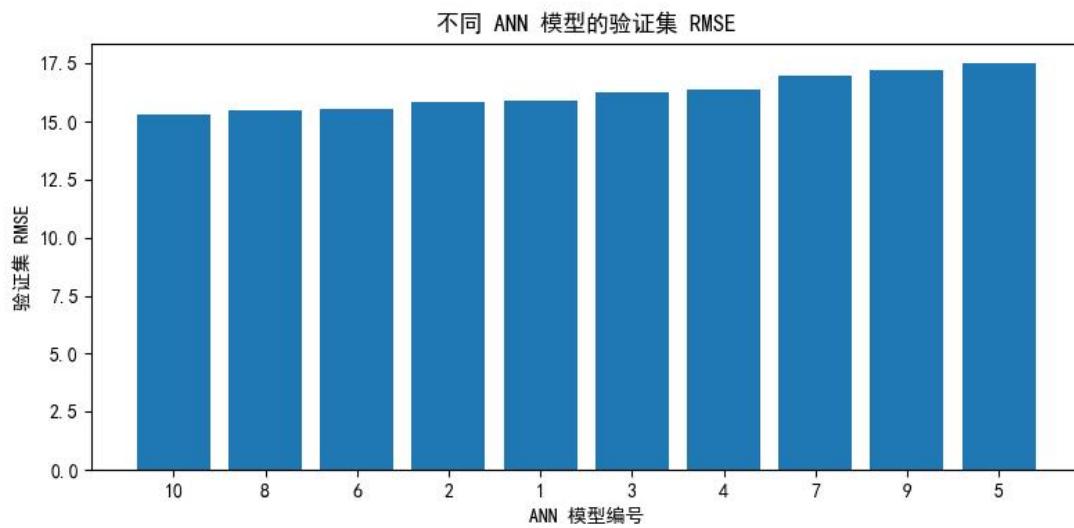


图 1-3 不同 ANN 模型的验证集 RMSE

为了更直观地分析网络复杂度与泛化性能之间的关系,可以将每个模型的"规模"简单定义为隐藏层神经元总数(hidden_dims 之和),绘制"模型规模 vs 验证 RMSE"的散点图.

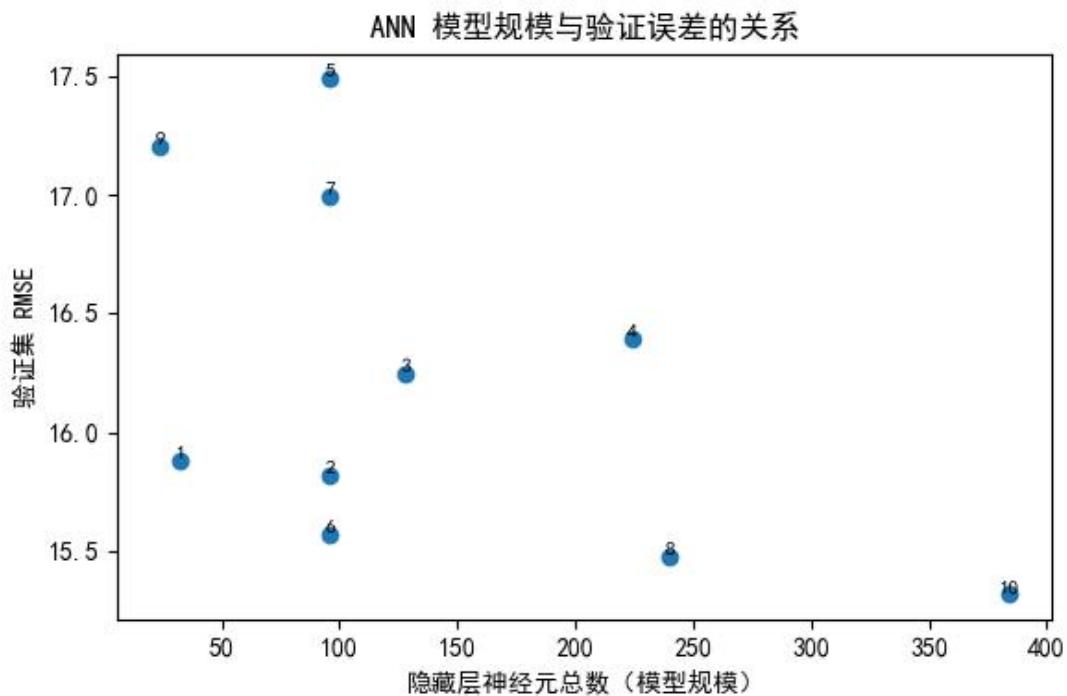


图 1-4 模型规模与验证 RMSE 的关系

从图 4 可以看到,当模型规模从较小值增加到中等规模(例如模型 2、6、8、10)时,验证 RMSE 有明显下降;但继续扩展规模并不能显著改善性能,甚至在某些组合下可能导致验证误差略有升高,这说明在当前样本量下,过大的网络并不必要.

5. 实验小结

本实验基于两个 UCI 数据集,共构建了 20 个不同的模型(10 个 SVM,10 个 ANN),满足了"通过修改参数生成多个模型并保存预测结果"的课程要求.对学生表现数据集,线性核 SVM(C 较小)在验证和测试集上取得了最稳定、最优的分类性能,而部分 RBF、多项式核模型在训练集上过拟合明显.对 NHANES 年龄预测数据集,多层全连接网络能够捕捉健康指标与年龄之间的非线性关系.适中规模、合理 Dropout 与学习率配置的网络(如模型 10)在 RMSE 与 R^2 指标上表现最好.通过对汇总 CSV 的可视化分析,可以直观地观察到超参数变化对模型性能的影响,为后续基于这些基模型进行朴素贝叶斯融合和 DS 证据理论融合提供了依据.

实验二 基于朴素贝叶斯的二层模型融合

1. 实验目的与思路

实验一分别在两个数据集上构建了 10 个基模型:学生数据集为 10 个 SVM 分类器,NHANES 年龄数据集为 10 个神经网络回归模型.实验二在此基础上,采用朴素贝叶斯(Naive Bayes)作为二层融合器,将"多个基模型的预测结果"视为新的输入特征,学习如何综合这些预测,从而得到最终的融合输出.实验目标如下:

- 1.利用实验一输出的多模型预测结果构造元特征(meta-features),训练朴素贝叶斯融合模型;
- 2.分别在学生分类任务与 NHANES 年龄任务上评估 train/val/test 性能;
- 3.输出融合后的预测表格与汇总结果,并保存融合模型(joblib)以便复现与后续对比(实验三).

2 数据与文件说明

2.1 输入来源

学生数据集:实验一中 10 个 SVM 在 train/val/test 上的预测输出(每个模型一份预测表).

NHANES 数据集:实验一中 10 个 ANN 在 train/val/test 上的预测输出(每个模型一份预测表).

实验二不再使用原始特征重新训练模型,而是把"基模型的输出"作为二层学习器的输入.

2.2 本实验输出文件(你已生成并上传)

学生数据集(NB 融合输出):

student_nb_train_predictions.csv
student_nb_val_predictions.csv
student_nb_test_predictions.csv
student_naive_bayes_results.csv
student_naive_bayes_model.joblib

NHANES 数据集(NB 融合输出):
nhanes_nb_train_predictions.csv
nhanes_nb_val_predictions.csv
nhanes_nb_test_predictions.csv
nhanes_naive_bayes_results.csv
nhanes_naive_bayes_model.joblib

3. 方法原理与实现思路

3.1 二层融合的基本思想

实验二采用一种简化的 stacking 思路:

第一层(基模型):实验一训练得到的 10 个 SVM/10 个 ANN;

第二层(融合器):朴素贝叶斯模型,只学习"基模型输出→最终真实标签/数值"的映射.

因此,融合器的输入不再是原始特征 X,而是一个由基模型输出的新特征矩阵 Z.代码 2-1 构建融合器的输入特征(节选)

```

01 def _load_student_split(split: str):
02     """
03     加载学生数据集某个划分上 10 个 SVM 模型的预测.
04     返回:
05     X_split: (n_samples, 10) 特征矩阵(每列是一个模型的 y_pred_encoded)
06     y_true_encoded: (n_samples,)
07     y_true: (n_samples,) 字符串标签
08     """
09     features = []
10     y_true_encoded = None
11     y_true = None
12
13     for i in range(1, N_MODELS + 1):
14         file_name = f"student_svm_model_{i}_{split}_predictions.csv"
15         path = os.path.join(SVM_DIR, file_name)
16
17         df = pd.read_csv(path)
18         required_cols = {"y_true_encoded", "y_pred_encoded", "y_true"}
19         if not required_cols.issubset(df.columns):
20             raise ValueError(f"文件 {path} 缺少必要列 {required_cols}")
21
22         if y_true_encoded is None:
23             y_true_encoded = df["y_true_encoded"].values
24             y_true = df["y_true"].values
25         else:
26             if len(df) != len(y_true_encoded):
27                 raise ValueError("各 SVM 预测文件样本数不一致")
28
29         features.append(df["y_pred_encoded"].values)
30
31     X_split = np.vstack(features).T
32     return X_split, y_true_encoded, y_true

```

3.2 学生数据集:以 10 个 SVM 预测标签作为离散特征 对每个样本,收集 10 个 SVM 的预测标签编码,形成:

$$Z = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{10}]$$

其中 \hat{y}_i 为第 i 个 SVM 的预测类别编码.二层融合器学习:

$$P(Y|Z) \propto P(Y) \prod_{j=1}^{10} P(Z_j | Y)$$

并输出 $\text{argmax}_Y P(Y|Z)$.

代码 2-2 模型训练

```

01 def run_student_naive_bayes_fusion():
02     """学生表现实验:使用 10 个 SVM 预测作为输入训练朴素贝叶斯融合器."""
03     _check_dir(SVM_DIR, "svm.py (学生表现 SVM)")
04     os.makedirs(STUDENT_FUSION_DIR, exist_ok=True)
05
06     print("===== 学生数据集:朴素贝叶斯融合 10 个 SVM =====")
07
08     # 1) 加载三个划分
09     X_train, y_train_enc, y_train = _load_student_split("train")
10     X_val, y_val_enc, y_val = _load_student_split("val")

```

```

11 X_test, y_test_enc, y_test = _load_student_split("test")
12
13 # 2) 训练朴素贝叶斯
14 nb = GaussianNB()
15 nb.fit(X_train, y_train_enc)
16
17 # 3) 在 train / val / test 上评估
18 results = []
19 for split, X, y_enc in [
20     ("train", X_train, y_train_enc),
21     ("val", X_val, y_val_enc),
22     ("test", X_test, y_test_enc),
23 ]:
24     y_pred_enc = nb.predict(X)
25     acc = accuracy_score(y_enc, y_pred_enc)
26     results.append({"split": split, "accuracy": acc})
27     print(f"{split.capitalize()} accuracy: {acc:.4f}")
28
29 # 4) 保存预测与模型
30 label_map = _build_label_map(y_train_enc, y_train)
31 train_pred_path = _save_student_predictions(
32     STUDENT_FUSION_DIR, "train",
33     y_train_enc, y_train, nb.predict(X_train), label_map
34 )
35 val_pred_path = _save_student_predictions(
36     STUDENT_FUSION_DIR, "val",
37     y_val_enc, y_val, nb.predict(X_val), label_map
38 )
39 test_pred_path = _save_student_predictions(
40     STUDENT_FUSION_DIR, "test",
41     y_test_enc, y_test, nb.predict(X_test), label_map
42 )
43
44 results_df = pd.DataFrame(results)
45 results_path = os.path.join(STUDENT_FUSION_DIR,
46                             "student_naive_bayes_results.csv")
47 results_df.to_csv(results_path, index=False)
48
49 model_path = os.path.join(STUDENT_FUSION_DIR,
50                           "student_naive_bayes_model.joblib")
51 joblib.dump({
52     "naive_bayes_model": nb,
53     "label_map": label_map,
54     "description": "Naive Bayes fusion over 10 SVM models (student dataset)."
55 }, model_path)

```

3.3 NHANES 年龄数据集:将回归任务离散化后做分类融合

NHANES 的基模型输出为连续年龄预测值.实验二的处理是:将真实年龄 y 四舍五

入为整数 y_{int} ,把"年龄整数"视为类别;以 10 个 ANN 的连续预测值构成特征向量 Z ;

用 GaussianNB 学习 $P(y_{int}|Z)$,输出预测整数年龄 y_{int} .

评估时同时给出:

`class_accuracy`:整数年龄完全一致的比例(严格指标);

MAE/RMSE:把 y_{int} 作为年龄估计值计算误差(更符合回归质量).

代码 2-3 模型训练

```
01 def _save_nhanes_predictions(output_dir, split,
02     y_true, y_true_int, y_pred_int):
03     df = pd.DataFrame({
04         "y_true": y_true,
05         "y_true_int": y_true_int,
06         "y_pred_int": y_pred_int,
07         "abs_error": np.abs(y_true - y_pred_int)
08     })
09     out_path = os.path.join(output_dir, f"nhanes_nb_{split}_predictions.csv")
10     df.to_csv(out_path, index=False)
11     return out_path
12
13
14 def run_nhanes_naive_bayes_fusion():
15     """
16     NHANES 年龄实验:使用 10 个 ANN 的连续预测作为输入,
17     以整数年龄为类别训练朴素贝叶斯融合器.
18     """
19     check_dir(ANN_DIR, "nn.py (NHANES ANN)")
20     os.makedirs(NHANES_FUSION_DIR, exist_ok=True)
21
22     X_train, y_train = _load_nhanes_split("train")
23     X_val, y_val = _load_nhanes_split("val")
24     X_test, y_test = _load_nhanes_split("test")
25
26     # 将真实年龄取整,作为类别标签
27     y_train_int = np.rint(y_train).astype(int)
28     y_val_int = np.rint(y_val).astype(int)
29     y_test_int = np.rint(y_test).astype(int)
30
31     # 训练朴素贝叶斯
32     nb = GaussianNB()
33     nb.fit(X_train, y_train_int)
34
35     # 在三个划分上评估
36     results = []
37     for split, X, y_int, y_true in [
38         ("train", X_train, y_train_int, y_train),
39         ("val", X_val, y_val_int, y_val),
40         ("test", X_test, y_test_int, y_test),
41     ]:
42         y_pred_int = nb.predict(X)
43         acc = accuracy_score(y_int, y_pred_int)      # 整数年龄准确率
44         mae = mean_absolute_error(y_true, y_pred_int) # 直接把整数当连续年龄
45         rmse = np.sqrt(mean_squared_error(y_true, y_pred_int))
46
47         results.append({
48             "split": split,
49             "class_accuracy": acc,
50             "mae": mae,
51             "rmse": rmse
52         })
53
54     _save_nhanes_predictions(
55         NHANES_FUSION_DIR, split,
56         y_true, y_int, y_pred_int
```

```

57     )
58
59     results_df = pd.DataFrame(results)
60     results_df.to_csv(
61         os.path.join(NHANES_FUSION_DIR, "nhanes_naive_bayes_results.csv"),
62         index=False
63     )
64
65     joblib.dump(
66         "naive_bayes_model": nb,
67         "description":
68             "Naive Bayes fusion over 10 ANN models (NHANES age dataset; "
69             "age treated as discrete)."
70     }, os.path.join(NHANES_FUSION_DIR, "nhanes_naive_bayes_model.joblib"))

```

4. 实验结果与可视化分析

4.1 学生表现数据集:融合效果分析

4.1.1 与最优单个 SVM 的整体性能比较

student_naive_bayes_results.csv 中给出的朴素贝叶斯融合准确率为:

训练集:0.8485 验证集:0.7201 测试集:0.7585

与实验一中基于验证集性能最优的 SVM 模型(线性核,C=0.1)的指标相比:

最优 SVM:train_accuracy=0.7761,

val_accuracy=0.7449,

test_accuracy=0.7788

朴素贝叶斯融合:在训练集上准确率更高,但在验证集和测试集上略低于最优 SVM.说明在此数据集上,简单地用朴素贝叶斯对 10 个 SVM 预测再学习,并没有带来明显的泛化性能提升,反而有轻微退化,可能原因包括:

10 个 SVM 之间的错误模式高度相关,朴素贝叶斯的条件独立假设不成立;

线性核 SVM 本身在该任务上已经有较强表现,融合空间有限.

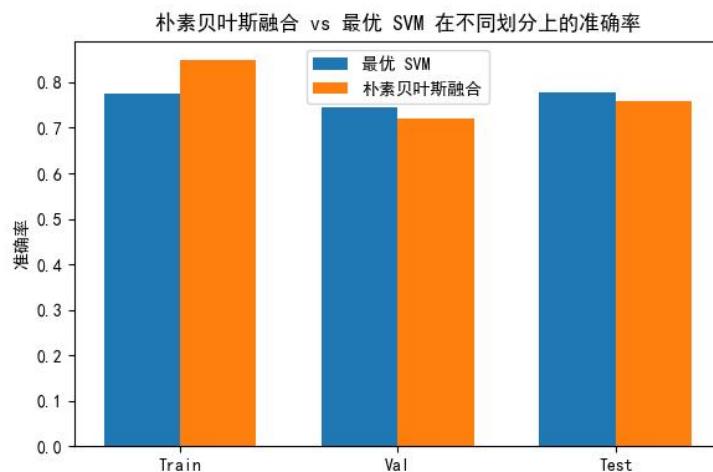


图 2-1 朴素贝叶斯融合 vs 最优 SVM 准确率比较

4.1.2 测试集混淆矩阵分析

student_nb_test_predictions.csv 中给出了朴素贝叶斯融合器在测试集上的真实标签和预测标签.基于这份表,我们可以计算 3×3 混淆矩阵,观察对不同类别的区分能力.从计算得到的混淆矩阵可以看到(以 Dropout/Enrolled/ Graduate 为顺序):大部分 Graduate 样本被正确分类,说明融合器对"顺利毕业"这一主流类别有较好

识别能力;Dropout 与 Enrolled 之间存在一定混淆,部分 Enrolled 被误判为 Graduate,表明模型对处于"在读"状态的学生区分能力有限.

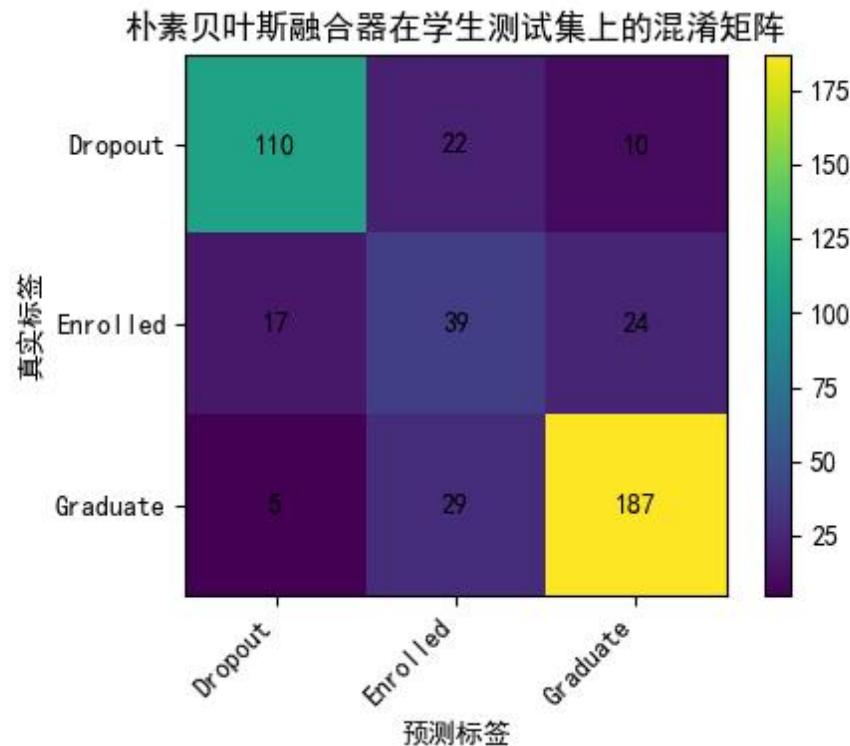


图 2-2 学生测试集混淆矩阵

4.2 NHANES 年龄预测数据集:融合效果分析

4.2.1 与最优单个神经网络的 MAE/RMSE 比较

nhanes_naive_bayes_results.csv 中给出的朴素贝叶斯融合结果为:

训练集:类别准确率约 0.0505(整数年龄完全相同的比例约 5%);MAE≈15.81,RMSE≈20.93;

验证集:类别准确率约 0.0219;MAE ≈ 15.53,RMSE ≈ 20.52;

测试集:类别准确率约 0.0307;MAE ≈ 15.55,RMSE ≈ 20.24.

与实验一中验证集 RMSE 最小的 ANN 模型

(hidden_dims=[256,128],lr=0.001,dropout=0.3) 相比:

最优 ANN:验证集 MAE ≈ 12.35,RMSE ≈ 15.32;测试集 MAE≈12.76,RMSE≈15.85;

朴素贝叶斯融合:验证集与测试集的 MAE 和 RMSE 都明显大于最优单模型,说明把年龄离散化后用朴素贝叶斯建模并不适合这个问题.

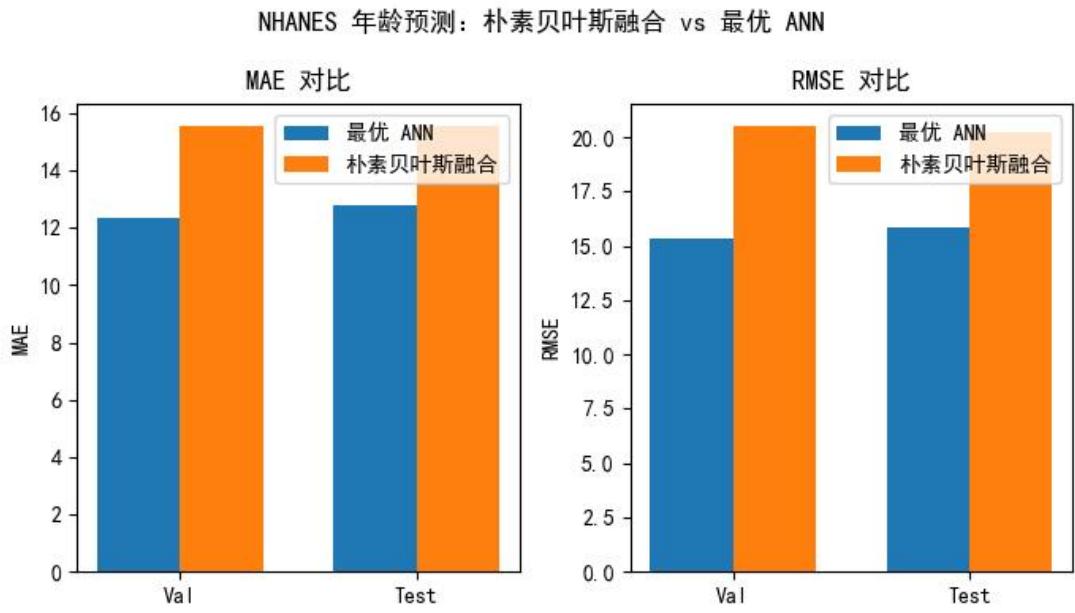


图 2-3 朴素贝叶斯融合 vs 最优 ANN 的 MAE/RMSE 对比

4.2.2 年龄预测误差分布

`nhanes_nb_test_predictions.csv` 提供了每个测试样本的真实年龄 y_{true} 、取整后的真实年龄 y_{true_int} 、朴素贝叶斯预测的整数年龄 y_{pred_int} 以及绝对误差 abs_error 。

对 abs_error 作描述性统计和直方图,可以发现:

误差的中位数约在 13 岁左右;四分之一的样本误差小于 5 岁,但也有一部分样本误差超过 20 岁甚至更高;从散点图(真实年龄 vs 预测年龄)来看,预测值整体围绕 40 岁附近波动,对年龄较大的样本(>60 岁)和较小的样本(<20 岁)偏差较大.这些现象说明:在当前设置下,朴素贝叶斯更像是对"平均年龄"的粗略回归,难以捕捉年龄的细粒度变化.

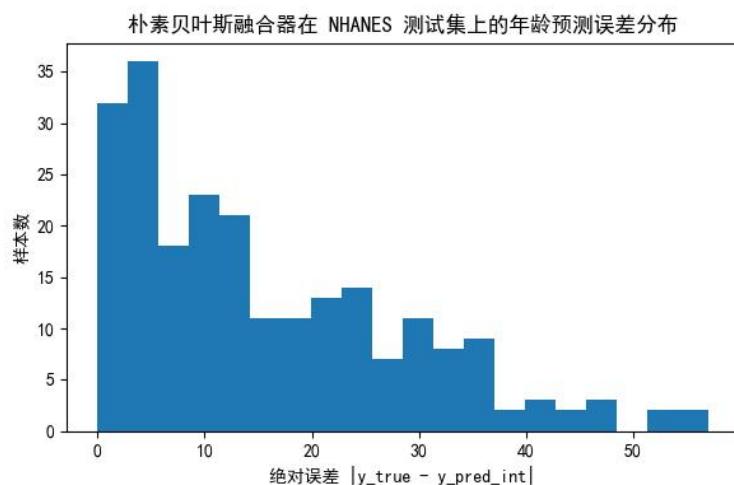


图 2-4:绝对误差直方图

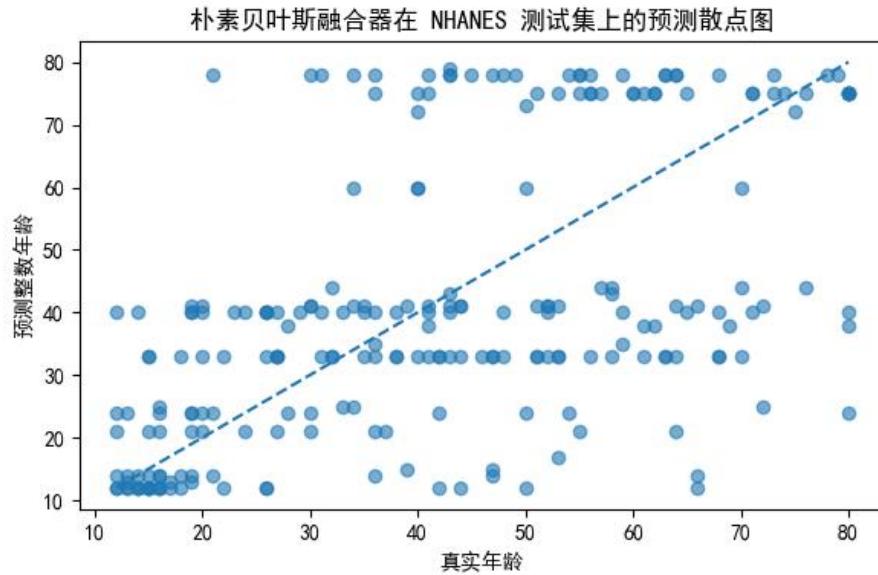


图 2-5:真实年龄 vs 预测年龄散点图

5. 实验小结

本实验在实验一训练好的 10 个 SVM 和 10 个 ANN 基础上,使用朴素贝叶斯构建了二层融合模型,所有融合逻辑集中在 `naive_bayes_fusion.py` 中完成,并生成了 8 份预测表、2 份汇总结果表以及 2 个融合模型文件.在学生表现数据集上,朴素贝叶斯融合器的训练集准确率高于任一单 SVM,但在验证集和测试集上并未超过最优 SVM 模型,说明简单的朴素贝叶斯融合在该场景下缺乏明显优势.在 NHANES 年龄预测数据集上,将年龄离散化后使用朴素贝叶斯进行二次建模,得到的 MAE 和 RMSE 明显劣于实验一中的最优神经网络,整数年龄的"精确命中"比例也较低,说明这种离散化+朴素贝叶斯的策略不适合作为该回归任务的最终方案.尽管融合效果未必优于最好的单模型,但实验二完整演示了利用基模型预测结果构造元特征、训练二层融合器、保存融合预测与评估指标的完整流程,为后续尝试更复杂的融合方法(例如加权投票、线性堆叠、树模型堆叠等)提供了基础.

实验三 基于 D-S 证据理论的多模型融合

1. 实验目的与要求

实验一已分别在学生表现数据集与 NHANES 年龄数据集上训练得到 10 个基模型(SVM/ANN 并生成 train/val/test 预测结果;实验二使用朴素贝叶斯对多模型预测进行二层融合.本实验进一步采用 D-S(Dempster-Shafer)证据理论实现融合,重点关注在"多模型预测存在冲突或不确定性"情况下的融合稳定性.

本实验目标如下:

1. 基于实验一输出的 10 个基模型预测结果,构建证据并进行 D-S 组合;
2. 分别在学生分类任务与 NHANES 年龄任务上给出 train/val/test 性能;
3. 与实验二朴素贝叶斯融合、以及实验一最优单模型进行对比分析;
4. 输出融合后的预测表格与汇总结果 CSV,并保存融合所需的 BPA(基本信任分配)对象(joblib)

2. 数据与文件说明

2.1 输入来源

学生表现数据集:实验一训练得到的 10 个 SVM 模型在 train/val/test 上的预测文件(作为证据来源).

NHANES 年龄数据集:实验一训练得到的 10 个 ANN 模型在 train/val/test 上的预测文件(作为证据来源).

注意:实验三不再使用原始特征训练新模型,而是直接利用"基模型输出"构造证据进行融合.

2.2 本实验输出文件

学生数据集(D-S 融合输出):

student_ds_train_predictions.csv
student_ds_val_predictions.csv
student_ds_test_predictions.csv
student_ds_results.csv
student_ds_bpa.joblib

NHANES 数据集(D-S 融合输出):

nhanes_ds_train_predictions.csv
nhanes_ds_val_predictions.csv
nhanes_ds_test_predictions.csv
nhanes_ds_results.csv
nhanes_ds_bpa.joblib

3. 方法原理与实现思路

3.1 证据表示与 BPA 构造

D-S 证据理论以识别框架 Θ 表示所有可能命题集合(分类任务中即所有类别).对每个基模型 m 以及每个样本 x ,需要给出其对各类别的"证据强度",用基本信任分配(Basic Probability Assignment,BPA)表示为 $m_m(\cdot)$.

本实验采用"由训练集统计得到的条件分布"来构造 BPA:对每个模型的预测标签 j,在训练集中统计真实标签落在各类 k 的频率,从而得到

$$m_m(k) = P(Y=c_k | \hat{Y}_m=j)$$

为避免某些类别计数为 0 导致融合时质量分配坍缩,构造过程中引入 Laplace 平滑.该策略的含义是:若某模型在训练数据上"预测为 j 时往往对应真实为 k",则当它在新样本上预测为 j 时,对 k 的证据强度更高.

代码 3-1

```

01 # ALPHA = 1.0 # Laplace 平滑
02 for pred_code in classes:
03     j_idx = code_to_idx[pred_code]
04     mask = (y_pred_enc == pred_code)
05     if not np.any(mask):
06         probs = prior.copy()
07     else:
08         counts = np.zeros(K, dtype=float)
09         for t in y_true_enc[mask]:
10             counts[code_to_idx[t]] += 1
11         probs = (counts + alpha) / (counts.sum() + alpha * K)
12     mat[j_idx, :] = probs # P(true | pred)

```

3.2 Dempster 组合规则与冲突处理

当同一样本存在多条证据(来自多个基模型)时,需要将其 BPA 进行组合.Dempster 组合规则的核心思想是:对一致证据进行强化,对冲突证据进行折扣并重新归一化.在仅对单点假设singleton)分配质量的实现设定下,组合结果可以写成逐元素乘积并归一化:

$$m_{DS}(k) = \frac{\prod_{m=1}^M m_m(k)}{\sum_{k'} \prod_{m=1}^M m_m(k')}$$

为避免多模型连乘出现数值下溢,本实验在实现中采用 log-domain 累加后指数化的方式计算乘积,并通过归一化得到最终质量分配.

最终预测输出采用最大信任度准则

$$\hat{y} = \operatorname{argmax}_k m_{DS}(k).$$

代码 3-2

```

1 def _ds_combine_masses(mass_list):
2     log_mass = None
3     for m in mass_list:
4         m = np.clip(np.asarray(m, dtype=float), EPS, 1.0)
5         log_mass = np.log(m) if log_mass is None else log_mass + np.log(m)
6         log_mass = np.max(log_mass)
7     mass = np.exp(log_mass)

```

```
8     mass /= mass.sum()
9     return mass
```

3.3 融合流程与评估指标设计

本实验在学生数据集与 NHANES 数据集上均按统一流程执行:

1. 读取基模型预测结果: 分别加载 10 个基模型在 train/val/test 上的预测文件;
2. 构建证据(BPA): 在训练集上统计得到每个模型的 BPA 矩阵, 并保存为 *_ds_bpa.joblib;
3. 进行 D-S 融合推断: 在三个数据划分上对每个样本执行证据组合, 得到融合后的预测;
4. 输出与评估:

学生数据集, 输出融合预测标签, 评估指标为准确率(accuracy);

NHANES 数据集, 将年龄离散为整数类别进行融合, 评估 class_accuracy, 并将预测整数作为年龄估计值计算 MAE 与 RMSE.

4. 实验结果与分析

4.1 学生数据集:D-S 融合性能

student_ds_results.csv 给出的准确率为:

Train accuracy=0.828434

Val accuracy=0.753950

Test accuracy=0.774266

与实验二朴素贝叶斯融合、实验一最优点 SVM(按验证集选取)对比如下:

方法	Train Acc	Val Acc	Test Acc
最优 SVM(实验一)	0.776145	0.744921	0.778781
朴素贝叶斯融合(实验二)	0.848502	0.720090	0.758465
D-S 证据融合(实验三)	0.828434	0.753950	0.774266

从结果可见:

- (1) D-S 融合在验证集上较最优点 SVM 提升 0.0090(约 0.90 个百分点), 说明证据融合在验证集上能够更稳定地利用多模型一致性信息;
- (2) 测试集上 D-S 融合略低于最优点 SVM(差 0.0045), 但明显优于朴素贝叶斯融合(提升 0.0158), 表明在该分类任务中, D-S 融合较 NB 融合具有更好的泛化表现;
- (3) 与朴素贝叶斯融合相比, D-S 融合训练集准确率更低(差 0.0201), 但验证/测试更优, 体现其在一定程度上降低了"仅在训练集拟合更好但泛化不足"的风险.

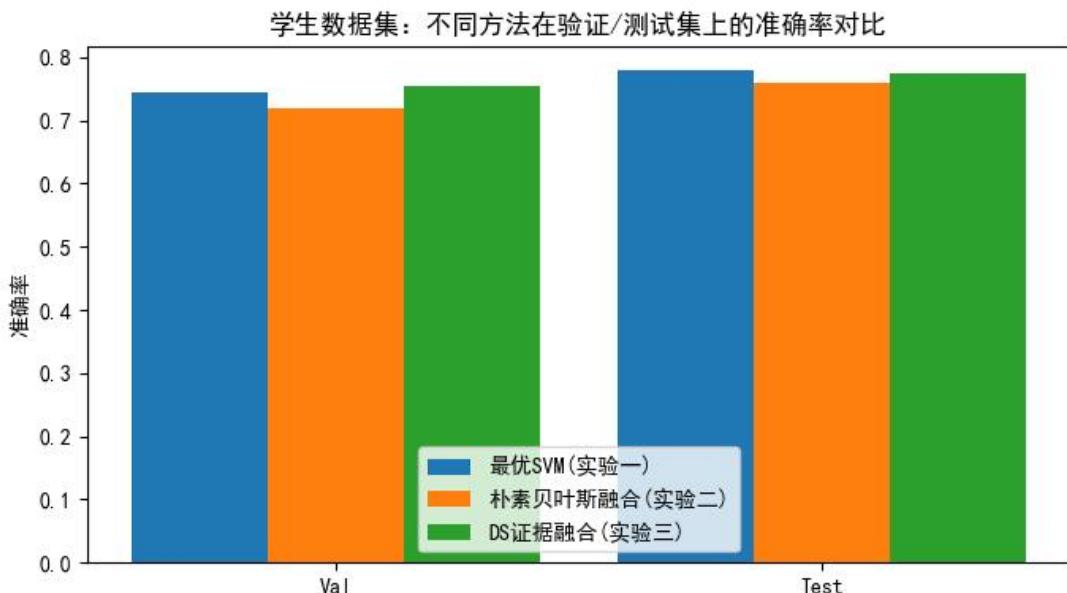


图 3-1 学生数据集:方法准确率对比

如图 3-1 所示(在验证集与测试集的柱状对比图中),D-S 融合在验证集上优势更明显,测试集与最优单模型接近,整体表现更均衡.

4.2 学生数据集:类别层面分析

为进一步分析 D-S 融合在不同类别上的表现,基于 student_ds_test_predictions.csv 计算测试集混淆矩阵(类别顺序为 Dropout/Enrolled/Graduate),结果如下:

真实\预测	Dropout	Enrolled	Graduate
Dropout	111	15	16
Enrolled	17	26	37
Graduate	7	8	206

由此可得到测试集类别分布为:Graduate 221,Dropout 142,Enrolled 80(类别不均衡较明显).进一步计算 Precision/Recall/F1:

Dropout:Precision 0.822,Recall 0.782,F1 0.801

Enrolled:Precision 0.531,Recall 0.325,F1 0.403

Graduate:Precision 0.795,Recall 0.932,F1 0.858

可以看到,D-S 融合对 Graduate 的召回率最高(0.932),说明其对"毕业"这一类别识别更充分;但 Enrolled 的召回率较低(0.325),主要错误来源为将 Enrolled 预测为 Graduate(37/80),这说明在多模型证据组合时,"倾向 Graduate 的证据"在乘积强化后更占优势,从而压制 Enrolled 的判别空间.由于 Enrolled 本身样本量较少且边界更模糊,该现象在集成模型中较常见.

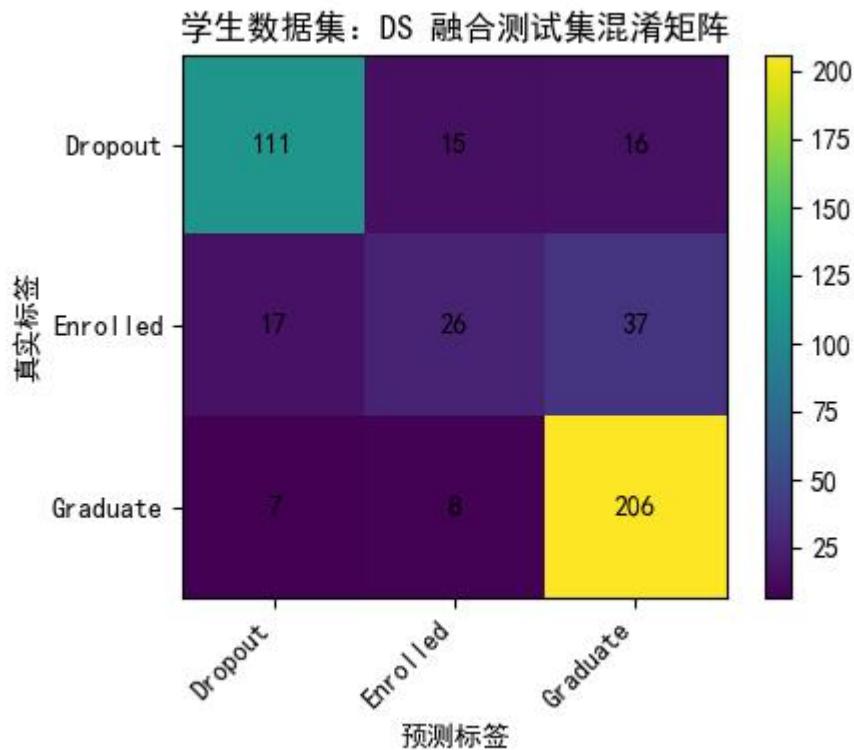


图 3-2 学生数据集:D-S 融合混淆矩阵

如图 3-2 所示,混淆矩阵的颜色分布能够直观呈现 Enrolled→Graduate 的主要混淆路径,为后续改进(例如引入类别均衡权重或对证据进行校准)提供方向.

4.3 NHANES 年龄数据集:性能表现与误差结构诊断

NHANES 年龄数据集采用 80%/10%/10% 划分(训练集 1822,验证集 228,测试集 228),年龄范围为 12–80.nhanes_ds_results.csv 给出的 D-S 融合结果如下:

Train:class_acc 0.205269,MAE 19.870472,RMSE 27.219097

Val:class_acc 0.074561,MAE 20.557018,RMSE 26.590709

Test:class_acc 0.039474,MAE 21.736842,RMSE 28.150598

与实验二朴素贝叶斯融合、实验一最优 ANN(按验证 RMSE 最小)对比如下:

方法	Val MAE	Val RMSE	Test MAE	Test RMSE
最优 ANN(实验一)	12.350339	15.317771	12.763282	15.845288
朴素贝叶斯融合(实验二)	15.526316	20.522132	15.552632	20.236973
D-S 证据融合(实验三)	20.557018	26.590709	21.736842	28.150598

可以得到以下结论:

- (1)在 MAE/RMSE 这类回归误差指标上,D-S 融合显著劣于最优 ANN 与 NB 融合,说明"回归离散化+证据组合"的策略在该数据集上并不适配;
- (2)尽管 D-S 的整数命中率 class_accuracy 在测试集上略高于 NB(0.0395 vs 0.0307),但该指标对年龄预测极为苛刻(必须整数完全一致),对模型优劣的区分能力有限;综合判断应以 MAE/RMSE 为主.

(3) 进一步对测试集误差进行统计,D-S 融合的绝对误差均值为 21.74,中位数为 18,上四分位数达到 36;而 NB 融合绝对误差均值为 15.55,中位数为 13,上四分位数为 23.25.这表明 D-S 融合存在更明显的"长尾大误差"现象.

为解释该现象,本实验对 D-S 融合的预测值分布进行了诊断:在测试集 228 个样本中,D-S 融合将 110 个样本预测为 80 岁(占 48.25%),而真实年龄为 80 的样本仅 8 个(占 3.51%).这一结果表明预测出现明显的"上界塌缩".结合实现细节可知:NHANES 融合中将基模型输出四舍五入后,并对预测年龄进行了[12,80]的裁剪;当多个基模型输出偏大时,裁剪会将大量预测压到 80 岁,证据组合的乘积强化进一步放大了该偏置,从而导致 MAE/RMSE 明显上升.

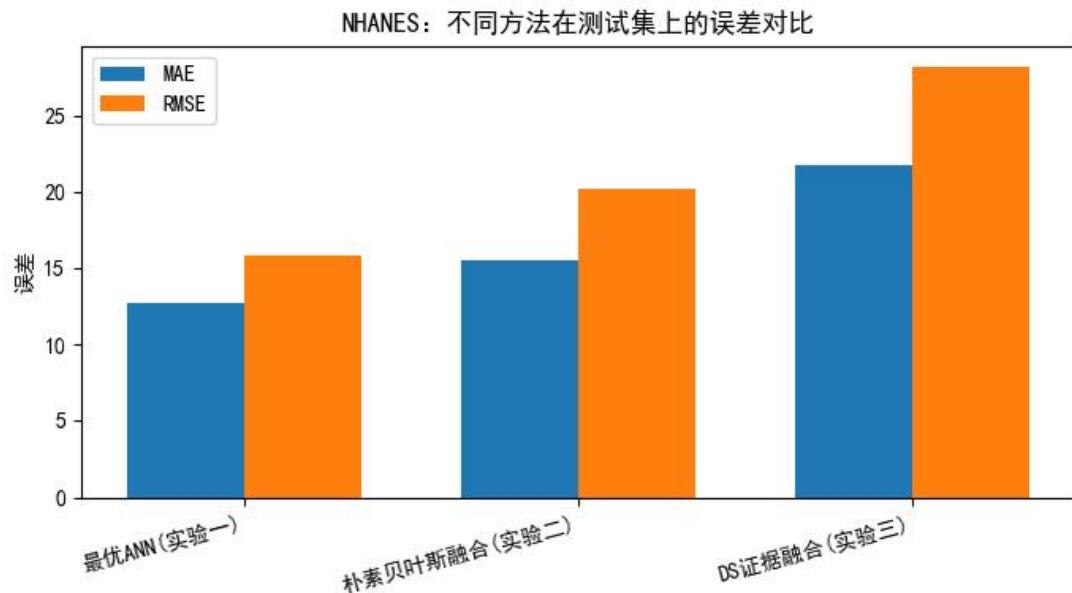


图 3-3 NHANES:MAE/RMSE 方法对比

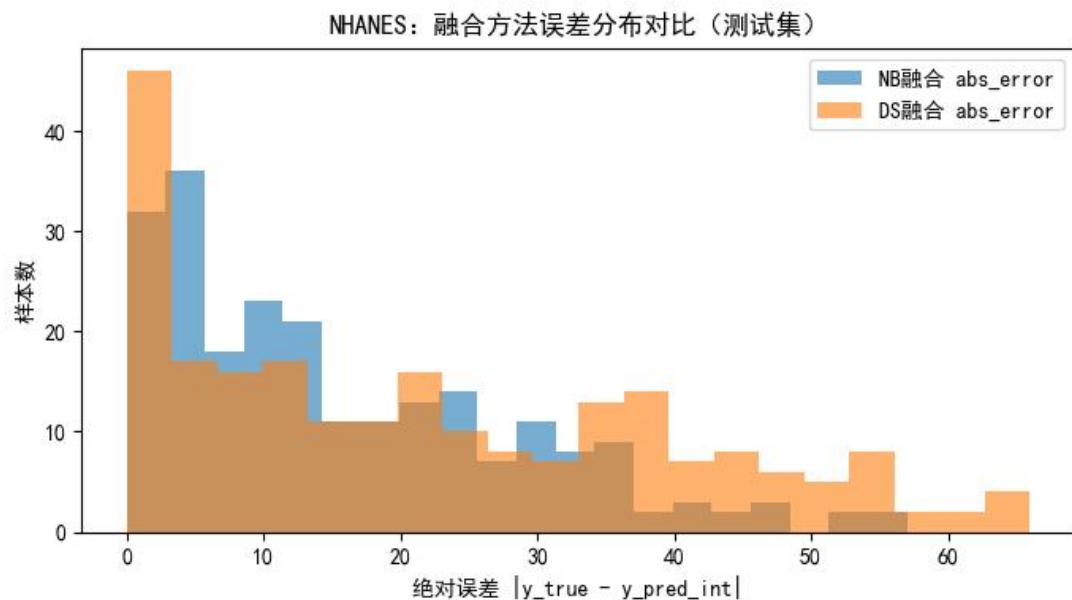


图 3-4 NHANES:误差分布(DS vs NB)

如图 3-3 所示(测试集 MAE/RMSE 对比),D-S 融合在误差上明显处于劣势;如图 3-4

所示(误差直方图对比),D-S 的误差分布右尾更长,与上述"预测塌缩"诊断一致.

5. 实验小结

本实验实现了基于 D-S 证据理论的多模型融合流程:训练集估计 BPA(预测→真实条件分布)→Dempster 组合(归一化乘积)→输出 train/val/test 融合预测与结果汇总,并保存 BPA joblib 以便复现实验.在学生分类任务上,D-S 融合在验证集上优于最优点 SVM,测试集略低于最优点 SVM 但明显优于朴素贝叶斯融合,表现出较好的稳定性.在 NHANES 年龄任务上,由于回归离散化与年龄取值空间较大导致证据稀疏,D-S 融合的 MAE/RMSE 显著劣于最优点 ANN 与朴素贝叶斯融合,说明该实现更适合类别空间较小、证据更"密集"的分类场景.综合三次实验可见:融合方法并非一定超越最优点模型,其效果与任务类型、证据质量与建模假设密切相关;对回归任务更建议采用连续输出的融合策略(如加权平均、线性堆叠或基于校准的回归融合).

基于模糊数学增强 DS 证据理论的多模型信息融合方法研究——

面向学生表现分类与 NHANES 年龄回归的统一框架

摘要: 在"多基模型输出层融合"场景中,Dempster-Shafer(DS)证据理论能够以基本概率分配(BPA)显式描述不确定性与无知,从而为集成学习提供可解释的融合机制.然而,在证据冲突较大、证据源可靠性差异显著以及连续回归输出需离散化处理等条件下,经典 DS 融合易出现过度自信、对样本难度不敏感和量化误差累积等问题.本文提出一种以模糊数学为核心的 DS 改进框架:通过模糊隶属函数构造"软 BPA"(允许将质量分配到集合/区间以表达犹豫),利用模糊推理系统实现模型级与样本级的动态折扣(discounting),并引入冲突再分配策略以抑制高冲突下的归一化放大效应.进一步地,本文结合两类典型数据集给出可落地的建模细化方案:对学生表现数据(3 类不平衡分类)强调"相邻类别的模糊集合质量"与"少数类的谨慎决策";对 NHANES 年龄预测(连续回归、年龄范围约 12–80)强调"连续输出的模糊区间证据表达"与"年龄段异质性可靠性建模".该方案不依赖改动基模型训练流程,仅替换融合层即可验证,具备良好的工程兼容性与扩展性.

关键词: 信息融合;证据理论;模糊数学;分类融合;回归融合

1.引言

在多模型集成中,常见融合策略(投票、均值、加权平均、Stacking)通常将模型输出视为"可直接比较的概率或数值",但对"模型不确定、犹豫或缺乏信息"的表达能力有限.DS 证据理论的优势在于: 允许将质量分配给集合命题(例如"属于 A 或 B")以及全集 Θ (表示无知),从而在理论层面对不确定性建模更自然.然而,在你现有实验设置中,融合器输入为 10 个基模型的输出(学生数据集为 10 个 SVM 的类别预测;NHANES 为 10 个 ANN 的年龄预测),此类"输出层融合"具有两个现实特征:

1. 证据源间冲突普遍: 模型结构/超参数差异导致预测分歧不可避免;
2. 证据源可靠性并非恒定: 同一模型在不同类别、不同年龄段或不同样本难度下性能差异明显.

经典DS若采用"单点假设+归一化合成",在冲突较大时容易产生"归一化放大",导致少数一致证据被过度强化,形成不合理的确定性结论.另一方面,回归任务若将连续输出硬离散(例如取整),会带来量化误差与信息损失.为此,引入模糊数学以增强 DS 的"软表达"与"自适应可靠性",是与当前实验框架高度匹配的一条改进路线.

2.数据集与任务特性分析

2.1 学生表现数据集: 三分类且类别不平衡

学生数据集样本量为 4424, 特征维度为 37, 目标变量为三类: Graduate/Dropout/Enrolled. 从总体分布看,类别显著不平衡: Graduate 约占一半,Enrolled 为最少类(约 18%),Dropout 居中(约 32%).

这种结构会对融合产生两个直接影响:

少数类(Enrolled)更易被多数类"淹没":若融合规则对冲突不敏感,最终更倾向输出多数类;

混淆结构具有方向性:例如"预测 Enrolled 但真实为 Dropout/Graduate"的代价与"预测 Graduate 但真实为 Enrolled"的代价可能不对称,融合机制应当具备更细粒度的犹豫表达与可靠性调节能力.

2.2 NHANES 年龄预测: 连续回归且年龄段异质性明显

NHANES 数据集样本量约 2278,年龄(目标 RIDAGEYR)范围约 12–80,均值约 41.8,标准差约 20.该数据天然具有"连续输出+年龄段异质性"的特征:

回归输出的连续性很关键:若融合阶段对预测值取整,会引入最少 0.5 年量级的量化误差,并削弱"预测接近但不相等"所蕴含的信息;

不同年龄段样本密度与可预测性差异:例如 Senior 子集样本相对更少、指标分布更复杂,模型误差可能随年龄段变化,可靠性应当可自适应调节.

3. 基线 DS 融合的实现形态与局限

在现有 DS 融合范式中,可概括为三步:

①证据构造: 对每个模型 i ,基于训练集统计"预测标签 \hat{y} 条件下真实标签 y 的分布",形成一个从 \hat{y} 到 y 的映射矩阵(可理解为行归一化的混淆统计).回归任务则将年龄预测与真实年龄取整后做同样统计.

②证据合成: 对每个样本收集 10 个模型给出的质量向量,采用 Dempster 规则在"仅单点假设"的退化情形下进行合成(等价于逐元素乘积后归一化).

③决策输出: 取合成质量最大的类别/年龄作为预测结果.

该方案工程上简洁,但存在三类关键不足,且与两套数据集的特性高度相关:

不足 A 单点假设导致"犹豫信息"缺失:学生三分类任务中,许多样本本质处于类别边界(例如 Enrolled 与 Dropout 的行为模式可能接近).若证据仅允许把质量分给单一类别,就无法表达"更像 A,但也可能是 B"的结构化不确定性.

不足 B 高冲突下的归一化放大:当 10 个模型产生明显分歧时,乘积归一化会压制少数类与非一致证据,使最终输出偏向"恰好一致的一方",容易过度自信.对不平衡分类任务,这会进一步加剧少数类被压制的问题.

不足 C 连续回归被硬离散化:NHANES 年龄预测若在融合阶段取整,模型之间"35.4 vs 35.6"的细微差异会被粗暴抹平,且"接近正确值"的证据无法以更高权重体现.

4. 模糊数学增强 DS 的核心思想: 把"证据"做软、把"可靠性"做活、把"冲突"做稳

4.1 模糊 BPA: 将质量分配到集合/区间以表达"犹豫"

4.1.1 面向学生三分类: 引入二元集合焦元(focal elements)

令识别框架 $\Theta = \{D, E, G\}$ 分别代表 Dropout、Enrolled、Graduate.经典 DS 常仅使用 $\text{singleton}\{D\}, \{E\}, \{G\}$.模糊增强的关键是显式引入二元集合焦元:

$$\mathcal{F} = \{\{D\}, \{E\}, \{G\}, \{D, E\}, \{E, G\}, \{D, G\}, \Theta\}$$

其含义是：当模型证据不够强时，不必硬押单一类，而可把部分质量分配给“二者之一”的集合命题。

现有证据构造本质上可得到：

$$q_i(k | \hat{y}) \approx P(y = k | \hat{y}, \text{modi})$$

对于样本 x ，模型 i 给出预测 \hat{y} 。可将“正确类概率”主要分配给 singleton，将“混淆部分”分配给二元集合命题。一个具有可解释性的构造方式为：singleton 质量：

$$m_i(\{\hat{y}\} | x) = (1 - \eta_i(x)) \cdot q_i(\hat{y} | \hat{y})$$

对任意 $k \neq \hat{y}$ ，二元集合质量：

$$m_i(\{\hat{y}, k\} | x) = (1 - \eta_i(x)) \cdot q_i(k | \hat{y})$$

无知质量：

$$m_i(\Theta | x) = \eta_i(x)$$

其中 $\eta_i(x)$ 为样本级不确定度

该构造带来一个直接好处：当某模型在训练统计上表现出“预测 Enrolled 时常混淆为 Dropout”，它会自然给 $\{E, D\}$ 分配更多质量，从而把“可解释的犹豫结构”带入合成阶段。这对学生数据集尤其关键，因为 Enrolled 为少数类且更容易被误判，集合焦元可抑制“一步到位的错误确定”。

4.1.2 面向 NHANES 回归：用模糊区间证据替代“取整单点”

回归任务建议将论域视为整数年龄集合或年龄区间集合。为兼顾计算与表达，可采用“区间焦元”：以预测 $\hat{y}^{(i)}$ 为中心定义三角或高斯隶属函数 $\mu_i(t)$ ，表示“年龄为 t 的相对可信度；将质量分配给若干个相邻区间（例如 $\{[a, a+2], [a-2, a]\}$ ）或邻域整数集合（例如 $\{[\hat{y}-1, \dots, [\hat{y}]+1]\}$ ）。

例如，用三角隶属函数表达“预测值附近更可信”：

$$\mu_i(t) = \max\left(0, 1 - \frac{|t - \hat{y}^{(i)}|}{d_i}\right)$$

其中 d_i 表示模型 i 的“模糊半宽”，可与其验证集 MAE/RMSE 相关；误差越大， d_i 越大。随后将 $\mu_i(t)$ 归一化得到对邻域年龄的软质量分配，并保留一定无知质

量 $m_i(\Theta)$ 表达"该样本上模型不稳定". 该方式相对"取整→单点假设"的优势是明确的：它保留了连续信息,且能把"接近正确"的预测以更高强度进入融合,理论上更有利降低 MAE/RMSE.

4.2 模糊折扣：让可靠性既考虑"模型差异",也考虑"样本难度"

经典 DS 折扣用常数 α_i 表示模型 i 的可靠性：

$$m'_i(A) = \alpha_i m_i(A) (A \neq \Theta), \quad m'_i(\Theta) = 1 - \alpha_i + \alpha_i m_i(\Theta)$$

其问题在于 α_i 通常是固定的,无法反映"同一模型在不同样本上可靠性不同".

4.2.1 结合你的实验可用信息：构造可靠性输入变量

在不额外改动基模型训练代码的前提下,你依然拥有足够的"融合层可用信息"来做动态可靠性建模:

模型级稳定性(来自验证集表现)

学生数据集：验证集 Accuracy、Macro-F1(建议引入 Macro-F1 以照顾少数类 Enrolled);

NHANES：验证集 MAE/RMSE.

样本级一致性(来自 10 模型输出的分歧程度)

分类：一致性可用"投票集中度"表示,例如多数类票数占比;

回归：一致性可用 10 个预测的标准差 $\sigma(x)$ 或 IQR 表示, σ 越大表示样本越难或模型越不稳定.

类别/年龄段敏感性(来自数据结构)

学生数据集：Enrolled 是少数类,若模型对 Enrolled 的验证集召回率低,则对"预测为 Enrolled"的证据应更谨慎(更高无知质量或更低折扣);

NHANES：Senior(高年龄段)样本相对少,若在高年龄段误差更大,可对 $t > 60$ 的样本提高模糊半宽或降低折扣.

4.2.2 模糊推理系统(FIS)示例：将"性能+一致性"映射到 $\alpha_i(x)$

可建立一个简洁而可解释的 Mamdani 模糊系统：

输入 1：模型级性能 P_i (分类用 Accuracy/Macro-F1; 回归用 RMSE 的反向指标);

模糊集：Low/Medium/High

输入 2：样本级一致性 $C(x)$ (分类用投票集中度; 回归用 $-\sigma(x)$ 或 $\sigma(x)$);

模糊集：Low/High

输出：动态折扣 $\alpha_i(x)$; 模糊集：Small/Medium/Large

典型规则：

R1： IF P_i is High AND $C(x)$ is High THEN $\alpha_i(x)$ is Large

R2： IF P_i is Medium AND $C(x)$ is High THEN $\alpha_i(x)$ is Medium

R3: IF P_i is Low OR $C(x)$ is Low THEN $\alpha_i(x)$ is Small

机制对两套数据集都能产生"符合直觉的行为": 当模型总体可靠且样本上各模型一致时, 强化证据; 当模型不够强或样本分歧大时, 将更多质量转移到无知 Θ , 避免在高冲突样本上做过度自信判断.

5.面向两套数据集的具体融合设计建议

5.1 学生三分类: 针对"不平衡+易混淆"的模糊 DS 设计

1. 焦元设计: 使用 singleton+二元集合+ Θ , 尤其强调以刻画 Enrolled 与其它类的犹豫区间.

2. 不确定度的设计: 由条件分布熵 $H(q_i(\cdot | \hat{y}))$ 驱动, 熵越大表示"预测该类时真实分布越分散", 应提高 η ; 加入少数类谨慎项, 当 $\hat{y} = E$ 且模型在验证集上对 E 的召回较低时, 进一步提高 η .

3. 冲突策略: 优先采用"冲突并入无知", 提升对高分歧样本的稳健性; 在决策阶段使用 Pignistic 概率(下注概率)输出最终类别, 避免仅用最大质量导致的硬切换.

4. 预期改善点(报告可讨论): Macro-F1 提升(尤其 Enrolled 的召回率/精确率更平衡); 高冲突样本子集上的错误率下降; 融合结果的解释性增强: 可通过 $m_i(\Theta)$ 或 $\{E, D\}$ 的质量观察模型犹豫来源.

5.2 NHANES 年龄回归: 针对"连续性+年龄段差异"的模糊 DS 设计

1. 避免取整: 以预测值为中心构造模糊隶属函数, 将质量分配到邻域年龄集合/区间;

2. 模糊半宽 d_i 的自适应: 由模型 i 在验证集上的 MAE/RMSE 决定(误差更大→更宽的模糊区间), 并可对 Senior 年龄段设置更大的 d 以反映更高不确定性;

3. 一致性驱动的折扣: 当 10 个模型预测方差 $\sigma(x)$ 较大时, 降低各模型折扣或提高 $m_i(\Theta)$, 避免冲突合成导致"离散化后错误集中";

4. 决策输出: 推荐输出期望年龄或 BetP 后的加权平均, 而非"取最大质量的单点年龄", 以获得更平滑的 MAE/RMSE 表现.

5. 预期改善点: 相比"取整 DS", 量化误差显著降低; 在模型分歧较大样本上, 输出更稳定、极端误差减少; 年龄段分层评估中, 高年龄段(Senior)误差更可控.

6.实验方案撰写模板

6.1 对照组设置

基线 A: 简单融合(分类多数投票; 回归均值/中位数)

基线 B: 经典 DS(单点 BPA+Dempster 合成)

改进：模糊 DS(模糊 BPA+模糊折扣+冲突再分配)

6.2 指标与分析维度

学生数据集：Accuracy、Macro-F1、各类 Recall/Precision、混淆矩阵;额外建议报告"少数类 Enrolled 的 F1"

NHANES：MAE、RMSE、R²,并按年龄段(如 12–30、31–60、61–80)分层统计误差

6.3 消融实验

仅模糊 BPA(无折扣、无冲突再分配)

模糊 BPA+模糊折扣

模糊 BPA+模糊折扣+冲突再分配(完整方案)

7.结论

本文面向"10 个基模型输出层融合"的实验范式,提出一套模糊数学增强 DS 证据理论的统一框架.该框架以模糊 BPA 解决单点证据表达不足与回归离散化误差问题,以模糊折扣实现模型级与样本级可靠性的动态建模,并通过冲突再分配抑制高冲突下的归一化放大效应.结合学生三分类不平衡任务与 NHANES 连续年龄回归任务分析表明,该方法在理论上能够同时提升融合的稳健性、可解释性与对任务结构的适配性,为后续实现与实证验证提供了清晰路径.

多模态医学图像融合(MMIF)研究进展：从融合流程与层级 到方法体系与评价指标

摘要：随着医疗数字化基础设施与多模态采集能力的完善,临床决策正从单一影像征象判读演进为对影像、临床文本、实验室指标及组学信息等海量异构数据的综合建模.多源医学数据在特征分布、噪声模式与时间尺度上差异显著,易产生语义鸿沟、模态缺失与未对齐等问题,使得简单拼接难以获得稳定增益.本文围绕多模态医学图像融合(MMIF)在医学影像分析与精准预后中的作用,系统梳理融合流程(预处理、配准、融合与评价)及像素级、特征级、决策级三类融合层级,概述CT、MRI、PET/SPECT等主流成像模态的互补信息与典型融合场景.进一步从方法学视角综述传统融合策略(空间域、频域/多尺度变换及梯度域)与深度学习融合框架(以CNN、GAN为代表并扩展至AE、CSC、RNN、RBM等)的建模思想与适用性,讨论公开数据集(如OASIS、TCIA、ADNI等)与常用客观评价指标体系MI、SSIM、PSNR、EN、SF、AG等).最后,结合肝脏CT等临床应用需求,归纳当前在配准依赖、噪声鲁棒性、可解释性、数据规模与客观评价标准等方面瓶颈,并指出传统方法与深度学习方法协同、面向真实临床分布与任务驱动的融合评估将成为重要发展方向.

关键词:多源信息融合;多模态医学图像融合;CT;深度学习;图像质量评价指标

1.引言

随着医疗数字化基础设施的完善,现代临床决策过程已演变为对海量异构数据的综合处理过程.在肿瘤学、神经病学及心血管疾病等复杂领域,医生的诊断依据不再局限于单一的影像学征象,而是结合了患者的基因组学背景、病理学微观特征、电子健康档案(EHR)中的既往病史以及放射学报告中的专家描述.这种多源数据的汇聚,理论上为构建患者的"数字孪生(Digital Twin)"提供了可能,从而实现对疾病进展的精准预测.医学图像是临床诊断的重要依据,广泛应用于治疗规划、手术导航中.临床诊断的准确性需要同时提取多幅不同模态的医学图像信息来保证.因此,图像融合技术受到广泛关注.图像融合是计算机视觉和图像处理领域的一个热门课题,包括多聚焦图像融合、多曝光图像融合和多模态图像融合等.而多模态医学图像融合(multimodal medical image fusion,MMIF)是多模态图像融合中的代表性融合场景.通过将不同模态医学图像中的互补信息整合到一幅图像中,帮助放射科或肿瘤医生等加快诊断过程,提高决策能力,降低储存成本.同时,由于医疗保健行业的巨大进步,医学成像传感器也飞速发展,加大了医疗系统对图像融合的需求.

然而,数据量的增加并不直接等同于信息的增益.多源医学数据面临着严峻的"异构性(Heterogeneity)"挑战:放射影像(如肝脏CT)是高维、稠密的像素矩阵,反映了解剖结构与组织密度的空间分布;临床文本(Text)是离散、稀疏的符号序列,蕴含了医生的高度概括性语义与逻辑推理;而实验室指标则是时序性的数值流.不同模态数据在特征空间分布、噪声模式及时间尺度上的巨大差异,形成了难以逾越的"语义鸿沟(Semantic Gap)".此外,临床环境中的数据往往是非完备的,模态缺失(Missing Modality)与未对齐(Misalignment)现象普遍存在,这使得传统简单的特征

拼接方法(Concatenation)难以奏效,甚至可能因噪声引入而导致性能退化.

为了解决上述挑战,多源信息融合技术应运而生.其核心目标是通过算法模型,在特征提取、表示学习及决策输出等不同层级上,实现多模态数据的有机整合,从而获得优于任一单模态的预测性能.图像融合过程由预处理、图像配准、图像融合和性能评价4部分组成.在预处理阶段,识别出图像中的噪声和伪影并完全去除,获得高质量的医学图像.然后,选取一幅参考图像,对剩余图像进行几何变换,使其与参考图像同步,获得配准后的待融合图像,这是与图像融合直接相关的步骤,它纠正了输入图像之间的偏差,补偿了原始信号重建、平移、旋转和缩放过程中造成的变化,从根本上保证了图像融合的精度.在融合过程中,图像融合等级按照作用的层次分为像素级、特征级和决策级,如图1所示.像素级融合对图像像素进行综合分析,能够保持尽可能多的现场数据;特征级融合对图像特征进行综合分析,可以压缩信息使其具有良好的实时性;决策级融合对图像进行特征提取和特征分类,通过大量的决策系统对分类后的图像特征进行融合.

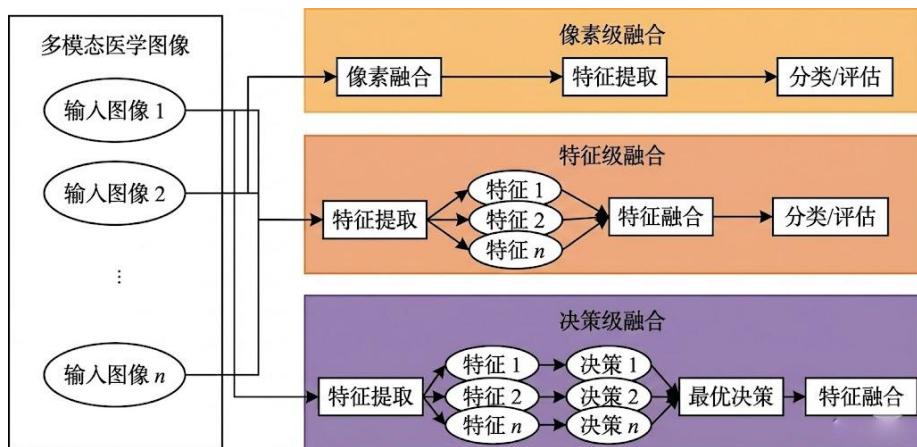


图1 图像融合等级

融合性能评价分为主观评价和客观评价.主观评价基于人眼视觉系统.在医疗领域中,通过观察图像的失真情况、空间细节、颜色和亮度等,或结合主观的标准分计算来判断融合质量是简单而可靠的.客观评价则通过一系列与人类视觉系统高度一致的指标来定量评估融合算法的性能.

2. 医学图像模态

医学成像的研究目的在于借助各种科学技术可视化人体内部的结构和组织,为疾病诊断提供重要信息.不同的成像技术可以从不同方面互补地反映患者的信息,而图像融合技术可以将它们结合起来.

医学图像是由能量和人体组织相互作用形成的.在医学领域中,成像技术主要分为利用电磁能成像和利用声能成像.利用声能成像是指利用超声波在不同介质中的传播速度不同来达到实时成像的效果,直接提供动态图像.而MMIF技术主要针对由电磁能成像技术形成的静态图像.如X射线计算机断层扫描成像(computed tomography,CT)、单光子发射型计算机断层成像(single photon emission computed tomography,SPECT)、正电子发射型断层成像(positron emission tomography,PET)和磁共振成像(magnetic resonance imaging,MRI)等.图2显示了

基于电磁能成像技术各电磁波频谱上的可形成的医学图像.

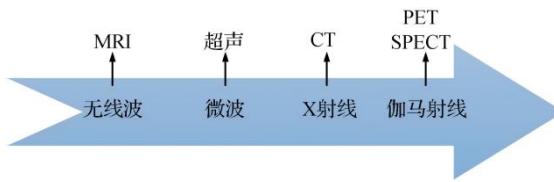


图 2 各电磁波频谱上的可形成的医学图像

2.1 计算机扫描成像

1895 年,W.K.Roentgen 发现了 X 射线,人类历史上第一次实现用科学技术观察人体内部结构.X 射线将 3 维目标投影在 2 维的检测平面上,但投影方向上信息相互重叠,成像的分辨率低,仅能区别密度差别很大的脏器.20 世纪 60 年代,计算机断层扫描技术(CT)逐渐发展起来,并从根本上解决了上述问题.CT 利用 X 射线对人体进行平移扫描,获取多个方向上的投影数据,然后用精确的数学公式重建出人体内部的剖面结构.而数字图像处理技术将感兴趣区域的细微灰度差变换为屏幕上人眼可分辨的灰度差,大幅提高了区分密度差异小的不同组织的能力,其灰度仅与组织的强度有关,与细胞的活性无关.CT 图像的应用范围几乎涵盖了全身各个系统,特别是对于中枢神经系统、头颈部和呼吸系统病变的检出和诊断具有明显优势.对于心血管系统、生殖系统和骨骼肌肉系统病变也具有较高的诊断价值.但是,CT 检查对某些病变的检出具有一定难度.例如,CT 对骨骼肌肉系统中的骨骼敏感,但对软骨、关节盘和韧带等软组织的病变显示仍十分困难.

2.2 磁共振成像

磁共振成像(magnetic resonance imaging,MRI)利用与原子核共振波长相同的电磁波激发人体内部的原子核,使之处于受激的状态. 原子核恢复到平衡态时会产生辐射,并携带核周围生化环境相关信息,它不仅可以显示解剖学形态的图像,还可以指示病理特征.据美国健康研究所(National Institute of Health,NIH)统计,现在全世界每年有 6 千多万人次接受 MRI 检查.MRI 可以对人体所有器官成像,尤其是对大脑组织和脊髓的成像最为精细,在研究和临幊上极有价值.如同 CT 图像,MRI 图像也是数字化的模拟灰度图像,亦具有窗技术显示和能够进行各种图像后处理的特点.然而,MRI 图像上的灰度并非表示组织和病变的密度,而是代表它们的信号强度,反应弛豫时间的长短.MRI 检查对于癌症的诊断、治疗和病程跟踪非常有效,可以为手术和放射治疗精确定位肿瘤的位置、边界以及对周边组织浸润的情况.但是在临床应用中,MRI 图像容易产生不同类型的伪影且有时不能完全消除,给图像解释带来困难,对某些系统疾病的检出和诊断的贡献有限,如 MRI 对于呼吸系统中大多数疾病诊断价值不高.

2.3 核医学成像

医学上放射性核素成像是将放射性核素标记在药物上,通过口服或注射方式引入人体内,检测体外该核素发射出来的能量形成图像.由于各脏器对同位素标记物的选择性吸收、正常组织与病变组织的吸收差异、代谢差异以及病变对标记物在体内循环产生影响等因素,使不同生理、病理的图像形成差异,故可以据此诊断疾病.临床应用的核素成像系统有两种,单光子发射型计算机断层成像(SPECT)和正电子发射型断层成像(PET),二者都用来测定人体各部位的生物化学和代谢过程.PET 图像的成像质量更高,但其示踪剂较少,应用范围受到限制,而 SPECT 的

示踪剂更易制作,可根据不同部位、不同症状选取合适的示踪剂进行成像.由于 PET 和 SPECT 图像的分辨率比较低,人们更强调其功能性研究.

2.4 融合模态

医学影像检查技术发展迅速,除了 CT、MRI 和核素显像,还有 X 射线、超声等常规成像技术.不同成像技术有各自优势和限度以及明确的范围.以脑部医学图像为例,CT 获取的图像可以提供丰富的解剖细节,能够清晰分辨脑内颅骨、脑实质、脑脊液和非病理性钙化区域;MRI 能够显示丰富的生理和生化信息,包括脑内神经、脑血管以及软组织;而 PET/SPECT 图像能够反映正常组织和病变组织对标记物的代谢情况以及脑部的血流信号.在实际的临床诊断中,对于某一系统疾病、某一类疾病通常需要综合应用几种成像技术才能满足诊断的需要.

图 3 是几种不同病例的多模态影像.图 3(a)显示了生活中常见的一种脑变性疾病阿尔茨海默症,其早期症状表现为短期记忆障碍,随着时间推移逐渐出现语言障碍、生活无法自理等症状.针对该疾病的首选检查方法为 MRI,其主要的影像依据是以海马为显著区域的脑萎缩,而导致脑萎缩的原因众多,如结合 PET 图像,则可明显观察到海马区域的血流信号减少、代谢降低等,从而确诊阿尔茨海默症.图 3(b)显示了脑膜瘤的医学影像.CT 可以扫出颅内的高密度区域(多为肿瘤区)以及其边界,肿瘤密度是否均匀等.而 MRI 图像可显示出肿瘤内是否存在条状流空血管、肿瘤内部信号是否均匀,亦可显示由脑膜瘤所致的骨改变.当脑膜瘤发生在功能区时,可有不同程度的神经性功能障碍,需结合 PET 或 SPECT 图像联合诊断.图 3(c)为 IV 级星形细胞肿瘤患者的医学影像,MRI 图像可体现其占位效应,瘤内坏死或出血以及微血管的密度和通透性,有助于评估肿瘤的病理分级.SPECT 图像可标记其葡萄糖代谢情况,可对其进行良、恶性肿瘤的鉴定.

基于实际的临床需求和应用价值,可以将主流的融合模态归纳为灰度图像的融合以及灰度与伪彩图像的融合两种.其中,灰度图像的融合主要涉及 CT 和 MRI 图像,灰度与伪彩图像的融合主要涉及 CT、MRI、PET、SPECT 4 种模态.融合示例如图 4 所示.

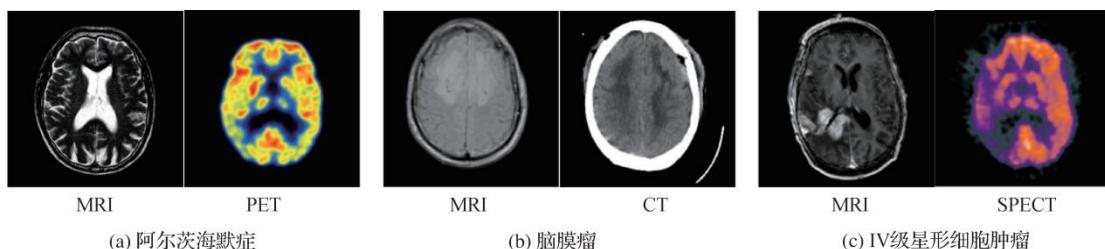


图 3 几种不同病例的多模态影像

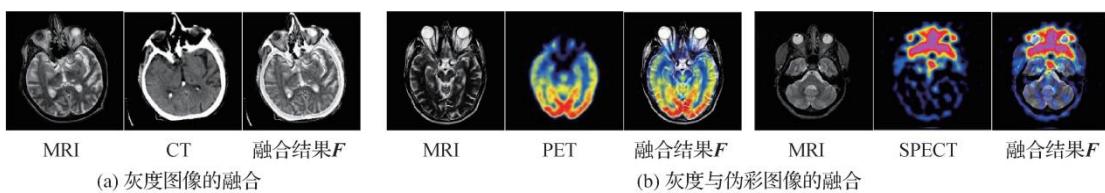


图 4 多模态医学图像融合示意图

3.融合方法

3.1 传统融合方法

传统 MMIF 方法着重处理源图像中的噪声、纹理、梯度、细节和颜色等信息,针对不同需求设计分解或重构策略和融合规则.这些方法按照对像素的处理方式和作用域的不同可分为基于空间域的融合方法、基于频率域的融合方法和基于梯度域的融合方法 3 类.

3.1.1 基于空间域的 MMIF 方法

空间域技术利用基础的像素级策略,对图像中的像元值起作用,得到的图像表现出更少的空间失真和较低的信噪比.基于空间域的方法主要包括简单最小/最大值、独立分量分析(independent component analysis,ICA)、主成分分析(principal component analysis,PCA)、加权平均、简单平均、模糊逻辑(fuzzy logic,FL)和云模型(cloud model,CM)等.其中,简单最小/最大值、简单平均方法相对原始,计算复杂度低,可以快速实现图像融合,然而融合效果并不理想.ICA 为了区分待融合图像中的有用信息和无用信息,在 ICA 域对源图像进行分解,并将其独立分量系数进行融合.而 PCA 则按照信息量对各个成分进行排序,将彼此不相关的各主成分分量进行筛选,丢弃信息量少的分量.模糊逻辑通常应用在融合规则的构造中,利用模糊隶属函数对图像内部的不确定性进行描述,从而对图像系数进行分配.云模型具有考虑随机性和模糊性的优点,利用逆向云发生器自适应地生成点云模型,计算其隶属度,找到合适的云模型构建融合规则.从多组数据中融合信息以提取一组最具特征的数据是融合任务的重点.而 ICA 提供一个简单的生成模型,对分解后的各个分量独立性进行度量,使多个模态充分交互以估计所有模态的潜在特征.Akhonda 等人讨论了 ICA 在多集融合中的两个重要扩展,即联合 ICA(joint independent component analysis,JICA) 以及多集典型相关分析和联合 ICA 技术,两种方法都假设相同的混合矩阵,强调跨多个数据集通用的组件,提出一个通用的框架,使用 ICA 的不相交子空间分析,不仅识别和提取多个数据集的共同成分,还可以提取不同的成分.该方法的一个关键组成部分是在后续分析之前识别这些子空间并分离它们,这有助于建立更好的模型匹配,并在算法和顺序选择方面提供了灵活性.Faragallah 等人提出一种基于主成分分析和奇异值分解(singular value decomposition,SVD)的多模态 MMIF 方法.该算法基于 PCA 与 SVD 的结合,将 CT 和 MRI 两种成像模式的所有相关信息整合在一起作为融合 CT 和 MRI 图像的预处理手段,可以减少融合过程的处理时间和对内存的需求,且图像质量与其他算法相同.Gao 等人采用基于图形的视觉显著性算法(graph-based visual saliency,GBVS)计算两个配准源图像的视觉显著性,在非下采样剪切波变换(non-subsampled shearlet transform,NSST)域内对源图像进行分解,得到低频和高频子带.对于低频子带,以局部能量和 GBVS 图为输入,利用模糊逻辑系统分别得到融合后低频子带的权值.此外,利用粒子群算法优化模糊逻辑系统的隶属度函数,使其更好地适应于医学图像和特征提取.Liu 等人提出一种基于改进和修正的拉普拉斯矩阵的像素级融合方法,利用局部拉普拉斯滤波对解剖图像和功能图像进行处理,然后通过最大化局部能量对子图像进行融合,来克服阻塞效应和伪影.Li 等人提出一种基于自适应云模型(adaptive cloud model,ACM)的 MMIF 方法,利用 ACM 融合经过局部拉普拉斯金字塔分解后的近似图像,使用拟合曲线通过捕获拟合曲线的谷点来表示图像细节信息.

3.1.2 基于频域的 MMIF 方法

在频域领域中,原始的手段是通过计算傅里叶变换将输入的图像从空域转换为频域,然后对转换后的图像应用融合算法,再进行傅里叶逆变换,得到最终的融

合图像.常见的变换域中的融合算法有基于金字塔、基于小波和基于多尺度几何变换的融合算法.

金字塔变换是最原始的图像分解手段之一,分解过程由连续滤波和下采样组成,产生一组类似金字塔结构的图像,过程如图 5 所示.首先将源图像分解为塔形结构的子图,随着分解层数的增加,子图尺寸逐渐减小,再将每一层的分解数据分别融合,最后重构成融合图像.在 MMIF 算法中,最流行的金字塔分解方法包括拉普拉斯金字塔(Laplacian pyramids,LP)、梯度金字塔(gradient pyramids,GP)和形态金字塔(morphological pyramids,MP)等.这些方法在面对有噪声干扰的图像时,容易产生块效应.形态金字塔边缘呈现的效率不高,而梯度金字塔会加入不需要的伪影(Lewis 等,2007).基于小波变换的融合算法成功克服了这些缺点.

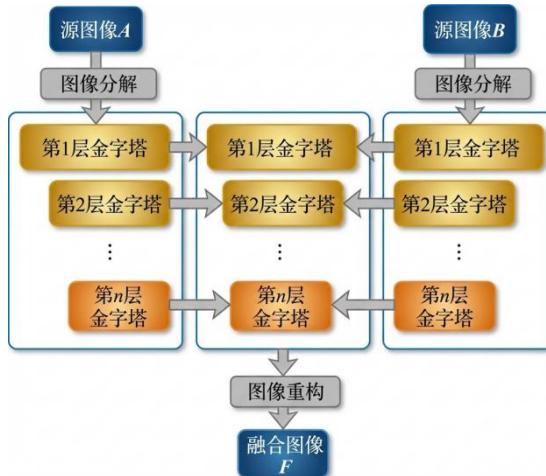


图 5 基于金字塔的图像融合方法

在 20 世纪 90 年代中期,出现了第 1 种基于小波的图像融合方法,并且证明其性能优于金字塔变换,其过程如图 6 所示.处理的基本步骤如下:1)对已配准的源图像进行小波分解,相当于使用一组高低通滤波器进行滤波,分离出高频信息和低频信息;2)对每层分解得到的高频和低频信息依据得到的信息特点采取不同的融合策略,在各自的变换域进行特征信息抽取,分别进行融合;3)采用步骤 1)小波变换的重构算法对处理后的小波系数进行反变换重建图像,即可得到融合图像.

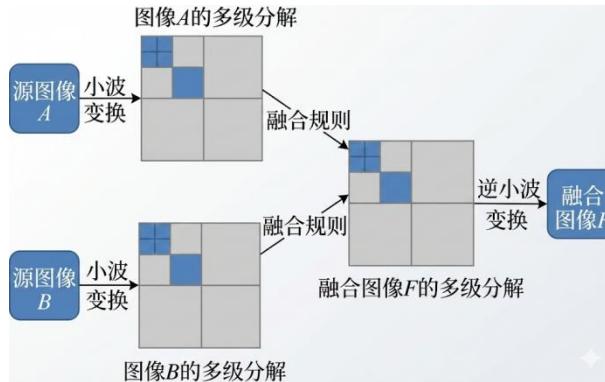


图 6 基于小波的图像融合方法图示

小波域通过将图像进行分解来保存图像信息,其系数对应源图像中不同的图像特征(Pajares 和 de la Cruz,2004).由于小波分解的近似图像包含了图像的绝大部分能量,且小波系数的均值基本为零,因此,基于小波变换的 MMIF 算法可以保留

源图像的基本亮度和色调.Shabanzade 和 Ghassemian 为了解决小波变换在表示方向性特征时缺乏位移不变性的缺点,提出了使用平稳小波变换(stationary wavelet transform,SWT)进行 MRI 与 PET 融合的算法框架,将非下采样轮廓波变换与 SWT 结合以发挥 SWT 的优势.Daniel 利用同态小波(homomorphic wavelet,HW)对源图像进行多级分解,并提出了针对解剖图像和功能图像的最佳比例系数.Prakash 等人利用双正交小波的线性相位特性来完成 MMIF 中的图像分解与重构.Ashwanth 和 Swamy 利用散小波变换(discrete wavelet transform,DWT)和 SWT 对源图像进行多级分解,针对分解后的子带图像分别采用基于边缘和基于能量的融合规则进行特征融合. 在 MMIF 中,小波变换的有方向性得到充分利用,且图像的频率信息利用率很高,但是小波变换本身不具有方向选择性和平移不变性,因此通常与其他频域变换方法结合来克服其局限性.而多尺度几何变换是通过定义特定的小波函数规则或在小波变换内进行特殊处理来克服这些限制.Bhateja 等人提出小波变换和几何变换 域中的脊波域(ridgelet domain, RD)具有一定的互补性.小波变换在边缘特征提取上能力较弱,而脊波却能很好地捕捉边缘信息.张鑫和陈伟斌提出基于曲线波变换(curvelet transform,CVT)的区域方差加权和条件加权融合算法并应用于 CT 和 MRI 图像融合.Mathiyalagan 提出一种基于曲线波变换的融合算法,对低通子带图像 进行小波变换,对高通子带图像进行脊波变换,利用子带图像的最大局部能量融合 PET 和 MRI.Do 和 Vetterli 提出轮廓波变换(contourlet transform,CRT).CRT 是利用拉普拉斯塔形分解和方向滤波器组实现的多分辨的、局域的、方向的图像表示方法.基于 CRT 的 MMIF 算法可以保持灰度图像的局部亮度,减少融合图像的失真.而 CRT 的平移可变性和方向混叠等缺陷促成了剪切波变换(shearlet transform,ST)的出现.为了解决 ST 出现的伪吉布斯现象,Easley 等人提出非下采样剪切波变换(non-subsampled shearlet transform,NSST),利用卷积代替下采样.目前,非下采样剪切波已经成为热门的图像分解和重构工具之一.

3.2 深度学习融合方法

深度学习在计算机视觉领域取得了巨大成就,在图像融合、语义分割和图像分类等视觉任务中都有卓越表现.DL 应用于图像融合的动机有两点,一是为了改进传统方法中多尺度和空间变换能力的局限性,DL 可以提供新的特征表示方法;二是传统方法的融合策略设计面临发展的瓶颈期,DL 能够更有效地映射输入与输出之间的相关性.深度学习领域内的融合方法主要包括卷积神经网络(CNN)、生成对抗网络(GAN)、卷积稀疏编码(CSC)、自动编码器(AE)、循环神经网络(RNN)和受限玻尔兹曼机(RBM)等.其中,以基于 CNN 和 GAN 的 MMIF 方法为主.

CNN 可以有效处理输入图像中邻域内的空间和结构信息.CNN 由一系列的卷积层、池化层和全连接层组成.卷积层和池化层可以提取源图像中的特征,全连接层完成从特征到最终输出的映射.图像融合在 CNN 中被视为一个分类问题,对应其特征 提取、特征选择和输出预测的过程,融合任务则面对的是图像变换、活动水平测量和融合规则的设计.

Liu 等人首次将用于多聚焦图像融合的 CNN 扩展到 MMIF 中,其中,图像变换利用了频域的拉普拉斯金字塔方法进行多尺度的分解,图像的活动水平测量利用 CNN 生成的权重图来计算.不同于其他数据类型的丰富储备,医学图像常常面临着小样本的限制,而 CNN 可以从小样本的医学图像数据集中进行学习,并且不容易产生过拟合的模型.Hermessi 等人提出了一种基于多通道 CNN 的 MRI 和 CT 图像的融合方法,先采用非下采样剪切波对图像进行高频和低频的子带分解,然后利用多通道 CNN 完成对高频子带图像的特征提取.Xia 等人提出一种深度堆叠

的 CNN 融合方法,该方法去除子采样层以获得与输入图像尺寸相同的输出大小,虽然用整个数据集作为输入所训练的网络可能忽略了模态之间的局部相似性,但该文献表明算法的融合能力受训练数据量的影响可以通过增加训练数据量来防止过拟合.Lahoud 和 Süsstrunk 提出一种基于 CNN 的零学习快速融合算法,不需要对特定模态的数据集进行预先训练,对各种模态的输入图像都能够提供有效融合.该算法通过充分利用已经训练好的网络来检测图像中的显著区域,并提取描述这些区域的深度特征图.通过比较这些特征映射,生成融合权重来合并源图像.Wang 等人在一个预先训练好的 CNN 模型下,采用对比金字塔分解源图像.Xu 和 Ma 提出一种无监督增强融合模型,通过一个转换网络完成灰度图像到伪彩图像的映射,利用编码解码器提取出具有独特信息的通道作为融合网络的信息约束.

与 CNN 不同,GAN 网络通过对抗性学习机制对医学图像中的显著性信息进行建模.GAN 是具有两个多层网络的生成模型,第 1 个网络是用来生成伪数据的生成器,第 2 个网络是用来将图像进行真实数据和伪数据分类的判别器. 基于反向传播的训练模式提高了 GAN 区分真实数据和生成数据的能力,尽管 GAN 在 MMIF 中应用不如 CNN 广泛,但具有深度研究的潜力.Tang 等人通过 GAN 融合了绿色荧光蛋白和相位对比图两种图像,利用生成器与鉴别器之间的对抗博弈,提取 GFP 图像的功能信息,同时提取 PC 图像的结构信息,并且提高融合图像与源图像之间的整体相似性.Xu 等人使用有密集连接的卷积层替换生成器中的 U-Net,因为密集连接可以加强层之间特征图的传递,使特征图的利用过程更加高效. 移除池化层后,没有了大步长的卷积核下采样造成的模糊,网络输出的融合图像更加清晰,并且判别器的输入不再是图像梯度,而是图像本身. 对于生成器中不同分辨率源图像的输入,不再对低分辨率源图像进行上采样,而是使用反卷积层来学习从低分辨率到高分辨率的映射.Zhao 等人构建了一个基于密集块和编码解码器的生成模块以及鉴别模块组成的 GAN. 并且在生成器的构造中,灵活设计了特征融合规则,扩大了算法的应用范围.

卷积稀疏编码起源于反卷积网络,该技术的主要目标是在稀疏性约束下实现图像的卷积分解. 输入图像的多阶段特征表示是通过发展这种分解的层次结构来从反卷积网络中学习的. 然后, 利用这些多重分解层次对输入图像进行分层重构.Liu 等人通过融合 PET/MRI 的感兴趣区域(region of interest,ROI) 来研究阿尔茨海默病的进展.Shi 等人通过使用堆叠去噪和稀疏自动编码器(denoising and sparse auto-encoder,DSAE)融合 MRI 特征来判断阿尔茨海默症的发展程度.Islam 等人比较了通过融合 MRI/CT 学习的 SEA(stacked auto-encoder)的分类准确性,并基于单一模式,取得了更好的性能. 循环神经网络(recurrent neural network,RNN)的主要应用领域是语音识别和文本分析.Chen 等人融合了 CNN 和 RNN 结构,利用 RNN 从 MRI 中提取特征,根据上下文信息,利用全卷积网络推断病变的概率. 受限玻尔兹曼机(restricted Boltzmann machine,RBM)是玻尔兹曼机(Boltzmann machine,BM)的一种变体,RBM 是由对称连接的可见层和隐藏层组成概率性、生成性、随机性和双向图形模型,动机是从隐藏层的向后传递中生成输入,并估计原始输入的概率分布.Suk 等人使用深度玻尔兹曼机(deep Boltzmann machine,DBM)进行深度特征融合,从 PET 和 MRI 中提取层次特征以提供阿尔茨海默症的计算机诊断辅助,但由于融合概念的动机不足,在这一背景下开展的融合工作较少.

4. 相关数据集及评价指标

4.1 多模态医学图像数据集

多模态医学图像融合(MMIF)研究主要依赖公开的标准数据集以确保配准精度和对比基准.常用的数据集包括：OASIS 系列(专注于脑部 MRI,涵盖阿尔茨海默症不同阶段的纵向数据及 PET 成像);TCIA(大规模癌症影像库,包含 CT、MRI、PET 等多种模态及基因、病理等综合数据);AANLIB(哈佛全脑图谱,专注于脑部病变如中风、肿瘤的 MRI/CT/PET 图像,配准质量高);ADNI(阿尔茨海默症神经影像及其生物标志物数据库,旨在跟踪疾病进展);以及 JIC 数据库(专注于细胞生物学,提供荧光显微镜 GFP 与相位差 PC 图像,用于细胞结构与功能融合研究).

表 1 多模态数据库比较

数据集名称	年份	核心影像模态	人体器官	涵盖疾病/研究对象	数据格式
OASIS	2007–2019	MRI(T1/T2), PET	大脑	阿尔茨海默症;正常老化;精神错乱	Analyze;NIfTI;JSON
TCIA	2011–至今	CT, MRI, PET, SPECT, US	全身(肺、脑、前列腺等)	20+种癌症(如肺癌、胶质瘤)	DICOM
AANLIB	1995–至今	MRI, CT, PET, SPECT	大脑	脑卒中;脑肿瘤;感染性脑病	GIF(网页展示为主)
ADNI	2004–至今	MRI, CT, PET	大脑	阿尔茨海默症;轻度认知障碍(MCI)	DICOM;NIfTI
JIC	—	显微成像(GFP, PC)	细胞(植物/生物)	细胞结构(线粒体、细胞核)	TIFF;PNG 等

注：—表示无固定格式

4.2 图像质量评价指标

无论使用何种图像融合方法,均需经过图像分解、融合规则、图像重构等过程.执行以上任务时,对图像的特征提取、压缩、传输和重建过程容易引起伪影、失真、过度增强等问题.因此,需要通过特定的图像质量评估(image quality assessment,IQA)方法来评判图像融合技术的优劣,评估方法可分为定性评估和定量评估两大类.

4.2.1 定性分析

在医学图像融合中,所谓定性分析是指通过人的视觉感受和视觉分析来评估融合图像的质量.这种主观的评估方法不使用任何模型技术,单纯依靠评估者的视觉感受和所发表的意见对融合图像质量进行点评.考虑到图像大小、空间细节、颜色等参数,此方法在许多实际融合中并不适用,因为这将导致价格昂贵、耗时、操作不便等问题.

4.2.2 定量评价

此分析法基于算法模型,采用一些评价指标,从客观上定量考核融合图像的质量,这些评价指标根据参考图像的适用性可进一步分为参考指标和非参考指标.参考指标包括互信息(mutual information,MI)、结构相似指数测量(structural similarity index measure,SSIM)、特征互信息(feature mutual information,FMI)、通用图像质量指数(universal image quality index,UIQI)、峰值信噪比(peak signal to noise

ratio,PSNR).非参考指标包括熵(entropy,EN)、标准偏差(standard deviation,SD)、空间频率(spatial frequency,SF)、平均梯度(average gradient,AG).

在上述参考指标中,MI 用于衡量融合图像与原始图像之间的相似程度,可以量化图像融合的质量,帮助优化图像融合算法,以实现更好的融合效果;SSIM 则旨在测量两幅图像之间的结构相似性,特别是在评估图像结构信息的保留与重建时极具优势;FMI 结合了互信息和特征表示的特点,提供一种更细粒度的图像相似性度量方式来反映图像融合后的特征保留程度;至于 UIQI,它结合了亮度、对比度和结构信息,旨在提供一个全面的图像质量度量对图像融合的效果进行全面的评估,以达到更好的融合效果;此外,PSNR 主要基于图像的像素值比较,而对图像的亮度、对比度和结构等特征以及特定类型的失真并不敏感,因此在评价图像融合质量时,还应结合其他指标来进行综合考量,以获得更全面的评价结果.

在前述非参考指标中,EN 反映了图像中的信息量,常被用来衡量图像的复杂度或信息量;SD 则常被用来评估图像的对比度或灰度分布的分散程度;此外,SF 分析提供一种直观的方式来理解图像中的空间结构变化,但某些情况下一些融合方法可能会导致 SF 的变化,会引入不自然或不希望的视觉效果,这就需要结合其他评价指标进行综合考量;AG 常用于衡量图像的清晰度或边缘的锐利程度,通过比较融合前后图像的 AG 值,可以了解融合是否导致了图像细节的丢失或边缘的模糊,从而评估融合的效果.

5 当前挑战与研究趋势

根据以上内容可得出,无论是传统方法还是深度学习方法都能够在一定程度上取得优越的融合表现.随着算法的不断优化和医学图像数据集的不断丰富,出现了越来越多融合效果好且模型鲁棒性高的医学图像融合方法.但是,在医学图像融合领域仍存在一些问题.本文通过总结上述研究工作,提出在 MMIF 领域中一些尚未解决的挑战和未来研究趋势.

5.1 现有挑战

现有 MMIF 算法主要面临以下问题:1)有限的算法创新没有推动 MMIF 产生质的飞跃,在助力图像融合系统、图像融合硬件设备发展过程中,算法创新性和性能提升带来的推动力远远不够.2)现有 MMIF 算法高度依赖图像配准,配准的精确程度直接影响融合的结果.3)像素强度异常、特征缺失、传感器误差、空间误差和图像间的变异性导致的特征处理和提取算法中的主要问题仍是医学图像融合中的一个开放问题.4)在近几年的 MMIF 算法中,对设备在采集图像过程中造成的噪声影响没有过多研究,多数 MMIF 算法未验证噪声鲁棒性.5)传统的 MMIF 算法中仍然存在计算量大、自适应性差和高度依赖人工设计融合策略的问题.6)基于深度学习的 MMIF 算法对数据集的量级和模态类别多样性的需求远大于目前可获取的内容,而网络的构建、损失函数的设计以及实验数据的设置同样缺乏合理的可解释性.7)图像融合结果的评价高度主观,针对其客观指标没有统一的标准,且目前没有针对医学图像融合结果评价的客观指标.

5.2 研究趋势

深度学习方法具有非常好的特征提取能力,能够完成绝大多数图像处理任务,但在没有 ground truth 的图像融合领域,传统方法仍然占据着一席之地.传统方法基于图像像素操作进行空间变换或系数变换.近几年,越来越多的学者将两者结合起来,充分利用两者的优势,其共同的研究趋势在于:拓展多种部位多种病例的医学图像、提出适合医学图像融合的客观指标以及拓展图像融合的研究范围.

6 结语

本研究对国内外医学图像融合方法相关文献进行归纳,将医学图像融合技术分为传统方法和深度学习方法两类。在传统医学图像融合方法中,基于空间域和频域的融合算法是近年的研究热点。空间域技术利用底层像素级策略作用于图像中的图像元素值,融合过程相对简单,算法复杂度低,通常计算量较少,在降低融合图像的光谱失真方面具有较好的性能。缺点是融合结果在清晰度和对比度上并不优越,时常导致空间分辨率较低。在频域,原始手段是通过计算傅里叶变换将输入图像从空域转换到频域,然后对转换后的图像应用融合算法,再进行傅里叶逆变换得到最终的融合图像。这类方法通常采用多级分解来增强融合图像的细节保持能力。输出的融合结果具有较高的空间分辨率和高质量的光谱成分。然而,这种算法也依赖于细粒度的融合规则设计。基于深度学习的方法主要是 CNN 和 GAN 网络,主要优点是不再依赖于细粒度的融合规则设计,减少了人工在融合过程中的参与,更强的特征提取能力使融合结果可以保留更多的源图像信息。然后,对现有多种医学图像数据库和融合质量评价指标进行全面概述。包括 5 个开源和免费访问的医学图像数据库 OASIS、TCIA、AANLIB、ANDI 和 JIC。常用的融合图像评价指标可以归纳为 4 类,即基于信息论的指标、基于图像特征的指标、基于图像结构相似性的指标和基于人类视觉感知的指标。此外,本研究进行大量实验比较基于深度学习的图像融合方法与传统医学图像融合方法的性能。通过对定性和定量结果的分析,对医学图像融合技术的现状、重点难点进行讨论,并指出未来发展前景。

参考文献

1. Akhonda MABS, Gabrielson B, Bhinge S, Calhoun VD and Adali T. 2021. Disjoint subspaces for common and distinct component analysis: application to the fusion of multi-task fMRI data. *Journal of Neuroscience Methods*, 358: #109214 [DOI: 10.1016/j.jneumeth.2021.109214]
2. Ashwanth B and Swamy K V. 2020. Medical image fusion using transform techniques // Proceedings of the 5th International Conference on Devices, Circuits and Systems. Coimbatore, India: IEEE: 303-306 [DOI: 10.1109/ICDCS48716.2020.9243604]
3. Azam MA, Khan K B, Salahuddin S, Rehman E, Khan SA, Khan M A, Kadry S and Gandomi A H. 2022. A review on multimodal medical image fusion: compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics. *Computers in Biology and Medicine*, 144: #105253 [DOI: 10.1016/j.combiomed.2022.105253]
4. Daniel E. 2018. Optimum wavelet-based homomorphic medical image fusion using hybrid genetic-grey wolf optimization algorithm. *IEEE Sensors Journal*, 18(16): 6804-6811 [DOI: 10.1109/JSEN.2018.2822712]
5. Du J, Li W S and Tan H L. 2020b. Three-layer image representation by an enhanced illumination-based image fusion method. *IEEE Journal of Biomedical and Health Informatics*, 24(4): 1169-1179 [DOI: 10.1109/JBHI.2019.2930978]
6. Faragallah OS, Muhammed A N, Taha T S and Geweid GGN. 2021. PCA based SVD fusion for MRI and CT medical images. *Journal of Intelligent and Fuzzy Systems*, 41(2): 4021-4033 [DOI: 10.3233/JIFS-202884]
7. Gao Y, Ma S W, Liu JJ, Liu Y Y and Zhang X X. 2021. Fusion of medical images based on salient features extraction by PSO optimized fuzzy logic in NSST domain. *Biomedical Signal Processing and Control*, 69: #102852 [DOI: 10.1016/j.bspc.2021.102852]

8. Hermessi H, Mourali O and Zagrouba E. 2018. Convolutional neural network-based multimodal image fusion via similarity learning in the shearlet domain. *Neural Computing and Applications*, 30(7): 2029-2045 [DOI: 10.1007/s00521-018-3441-1]
9. Li WS, Chao FF, Wang GF, Fu J and Peng X X. 2022a. Medical image fusion based on local Laplacian decomposition and iterative joint filter. *International Journal of Imaging Systems and Technology*, 32(5): 1631-1647 [DOI: 10.1002/ima.22714]
10. Li XX, Guo X P, Han PF, Wang X, Li HG and Luo T. 2020. Laplacian redecomposition for multimodal medical image fusion. *IEEE Transactions on Instrumentation and Measurement*, 69(9): 6880-6890 [DOI: 10.1109/TIM.2020.2975405]
11. Prakash O, Park CM, Khare A, Jeon M and Gwak J. 2019. Multiscale fusion of multimodal medical images using lifting scheme based biorthogonal wavelet transform. *Optik*, 182: 995-1014 [DOI: 10.1016/j.ijleo.2018.12.028]
12. Xu H and Ma JY. 2021. EMFusion: an unsupervised enhanced medical image fusion network. *Information Fusion*, 76: 177-186 [DOI: 10.1016/j.inffus.2021.06.001]
13. Xu H, Ma JY, Jiang JJ, Guo XJ and Ling HB. 2022. U2Fusion: a unified unsupervised image fusion network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 502-518 [DOI: 10.1109/TPAMI.2020.3012548]
14. Zhang H, Xu H, Tian X, Jiang JJ and Ma JY. 2021. Image fusion meets deep learning: a survey and perspective. *Information Fusion*, 76: 323-336 [DOI: 0.1016/j.inffus.2021.06.008]\
15. Zhang JJ, Zhou T, Lu HL and Wang HQ. 2016. Research progress of multi-model medical image fusion at feature level. *Journal of Biomedical Engineering*, 33(2): 394-399 (张俊杰, 周涛, 陆惠玲, 王惠群. 2016. 特征级多模态医学图像融合技术的研究与进展. 生物医学工程学杂志, 33(2): 394-399) [DOI: 10.7507/1001-5515.20160067]
16. Zhang X and Chen W B. 2014. Medical image fusion based on weighted Contourlet transformation coefficients. *Journal of Image and Graphics*, 19(1): 133-140 (张鑫, 陈伟斌. 2014. Contourlet 变换系数加权的医学图像融合. 中国图象图形学报, 19(1): 133-140) [DOI: 10.11834/jig.20140117]
17. Zhou P, Xi RH, Song LL and Wu X D. 2006. Realization of medical image fusion based on wavelet transform. *Journal of Image and Graphics*, 11(11): 1720-1723 (周朋, 奚日辉, 宋玲玲, 吴小丹. 2006. 基于小波变换的医学图像融合技术的实现. 中国图象图形学报, 11(11): 1720-1723) [DOI: 10.11834/jig.2006011301]
18. Zhou T, Liu S, Dong Y L, Huo BQ and Ma Z J. 2021. Research on pixel-level image fusion based on multi-scale transformation: progress, application and challenges. *Journal of Image and Graphics*, 26(9): 2094-2110 (周涛, 刘珊, 董雅丽, 霍兵强, 马宗军. 2021. 多尺度变换像素级医学图像融合: 研究进展、应用和挑战. 中国图象图形学报, 26(9): 2094-2110) [DOI: 10.11834/jig.200803]