智能制造大数据技术实践
风电功率预测建模方法

一、实验目的

1. 掌握风电场数据清洗和预处理方法；
2. 掌握 DBSCAN 聚类和 K-means 聚类方法；
3. 掌握最小二乘法、支持向量回归和 BP 神经网络预测建模方法。

二、实验仪器设备

1. PC
2. Matlab 或 Python

三、实验原理

风电功率预测技术：对未来一段时间内风电场所能输出的功率大小进行预测，以便安排调度计划。

$$P=\frac{1}{2}C_p\rho Sv^3$$

其中，C_p 称为功率系数，ρ 为空气密度，S 为风机叶片扫过的横截面积，v 为风速。在上述影响风电功率大小的因素中，风机的功率系数 C_p、风机叶片扫过的横截面积 S 可视为常量。空气密度大小又受到温度、湿度、压强等因素的影响，但相同地形地貌下的空气密度可近似为常量，因此主要考虑风速对风电功率的影响。根据上式可知，输出功率大小与风速的三次方成正比。

数据清洗和预处理：由于传感器精度、电磁干扰、信息处理、错误存储或通信故障等，以及目前电力系统对风电的消纳能力有限，风电场强制弃风已成为常态，这就使得原始记录数据中会存在大量的异常数据，用风速-功率（*v*-*p*）散点图表示时，这些异常数据严重破坏了风速和功率所应有的整体分布规律和对应关系，所以对风电场数据进行数据清洗和预处理是必要过程。

Density-Based Spatial Clustering of Applications with Noise（DBSCAN）：是一个比较有代表性的基于密度的聚类算法。它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。

BP(Back Propagation)神经网络：BP 神经网络是一种按误差反向传播(简称误差反传)训

练的多层前馈网络，其算法称为 BP 算法，它的基本思想是梯度下降法，利用梯度搜索技术，以期使网络的实际输出值和期望输出值的误差均方差为最小。

四、实验任务与步骤

根据如图 4-1 所示框图进行风电功率预测。

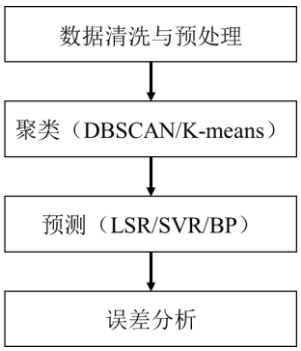

图 4-1　风电功率预测流程框图

4.1 数据清洗与预处理

1. 缺失值清洗：对风电场数据中为 NAN（空值）的数据值进行删除，并进行数据填充。

2. 异常值清洗：对数据进行可视化，删除明显的离群点；进行简单的数据分析，判断属性的取值是否超过合理范围。

3. 格式内容清洗：对风电场数据中不该存在的字符，属性与内容不符的格式问题进行清洗。

4. 逻辑错误清洗：首先若数据中的重复数据与实际情况不符，对数据进行去重处理；之后对风电数据中的不合理值进行去除，例如低风速高功率点。

5. 非需求数据清洗：若风电场数据中含有实际操作中不需要的数据，则进行清洗。

6. 数据归一化：消除所有变量量纲和取值范围差异。

4.2 数据聚类

1. DBSCAN 聚类算法

DBSCAN 算法需要首先确定两个参数：
（1）Epsilon：在一个点周围邻近区域的半径；
（2）MinPts：邻近区域内至少包含点的个数。
根据以上两个参数，结合 epsilon-neighborhood 的特征，可以把样本中的点分成三类：
（1）核心点：在半径 Eps 内含有超过 MinPts 数目的点。
（2）边界点：在半径 Eps 内点的数量小于 MinPts,但是落在核心点的邻域内的点。
（3）噪音点：既不是核心点也不是边界点的点。

DBSCAN 算法流程：

（1）DBSCAN算法通过检查数据集中每点的Eps邻域来搜索簇，如果点p的Eps邻域包含的点多于MinPts个，则创建一个以p为核心对象的簇；

（2）然后，DBSCAN迭代地聚集从这些核心对象直接密度可达的对象，这个过程可能涉及一些密度可达簇的合并；

（3）当没有新的点添加到任何簇时，该过程结束。

2. K-means 聚类算法

K-means 算法流程：

（1）从集合 D 中随机取 k 个元素，作为 k 个簇的各自的中心。

（2）分别计算剩下的元素到 k 个簇中心的相异度，将这些元素分别划归到相异度最低的簇。

（3）根据聚类结果，重新计算 k 个簇各自的中心，计算方法是取簇中所有元素各自维度的算术平均数。

（4）将集合 D 中全部元素按照新的中心重新聚类。

（5）重复第 4 步，直到聚类结果不再变化。

4.3 预测建模（BP 神经网络）

BP 神经网络设计问题：

（1）网络层数：具有至少一个S型隐含层加上一个线性输入层的网络，能够逼近任何有理函数。增加层数可以进一步的降低误差，提高精度，但同时也使网络复杂化。

（2）隐含层神经元：网络训练精度的提高，可以通过采用一个隐含层，而增加其神经元数的方法来获得。

（3）初始权值选取：BP神经网络是非线性优化算法，初始值设置不当，可能陷入局部极小。

（4）学习速率：决定每一次循环训练中所产生的权值变化量。为了减少寻找学习速率的训练次数以及训练时间，比较合适的方法是采用变化的自适应学习速率，使网络的训练在不同的阶段设置不同大小的学习速率。

4.4 误差分析

分析实测功率与预测功率之间的误差特性，主要为统计关系。以 MAE、RMSE 作为风电功率预测的误差指标，如下式：

$$MAE=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i-y_i\right| \tag{1}$$

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i-y_i\right)^2} \tag{2}$$

其中，$\hat{y}_i$ 为预测功率，y_i 为实际功率。

五、实验注意事项

1. 完成风电功率预测的每个步骤时，注意重新保存数据；
2. 保证代码美观并及时注释，保持良好的代码编写习惯。

六、实验报告

1. 根据图 4-1 的步骤来进行风电功率预测，根据风电功率数据特征，选取合适的数据预处理方法，完成预测建模，实现风电功率预测，对预测结果进行误差分析。
2. 绘制每个模块的流程图。
3. 完成各个步骤时，对数据进行可视化显示。

七、代码参考

Matlab

DBSCAN 聚类核心代码：

```
[IDX, isnoise]=DBSCAN(samples, epsilon, MinPts);
```

K-means 聚类核心代码：

```
[cluster_label, sse]=kmeans(samples, K);
```

BP 神经网络核心代码：

```
[error, BPoutput]=BP(train_input, train_output, test_input, test_output);
```

Python

DBSCAN 聚类核心代码：

```
from sklearn.cluster import DBSCAN
y_pred = DBSCAN(eps=3, min_samples=2). fit_predict (cluster_data)
```

K-means 聚类核心代码：

```
from sklearn.cluster import Kmeans
y_pred = Kmeans(n_clusters=2).fit_predict(cluster_data)
```

BP 神经网络核心代码：

```
from sklearn.neural_network import MLPClassifier
BP = MLPClassifier(solver='sgd', activation='relu', max_iter=500, alpha=1e-3, hidden_layer_sizes = (32,32), random_state=1)
BP.fit(train_samples, train_labels)
predict_labels = BP.predict(train_samples)
```