

# 半监督学习

## 1. 模型的一些通用方法：

- `get_params([deep])` : 返回模型的参数。
  - `deep` : 如果为 `True` , 则可以返回模型参数的子对象。
- `set_params(**params)` : 设置模型的参数。
  - `params` : 待设置的关键字参数。
- `fit(X,y)` : 训练模型。
  - `X` : 训练集样本集合。通常是一个 `numpy array` , 每行代表一个样本 , 每列代表一个特征。
  - `y` : 训练样本的标签集合。它与 `X` 的每一行相对应。其中未标记样本的标记为 `-1` 。
- `predict(x)` : 利用模型执行预测。返回一个预测结果序列。
  - `X` : 测试集样本集合。通常是一个 `numpy array` , 每行代表一个样本 , 每列代表一个特征。
- `predict_proba(x)` : 利用模型执行预测。返回每个样本在每个类别上的概率分布。
  - `X` : 测试集样本集合。通常是一个 `numpy array` , 每行代表一个样本 , 每列代表一个特征。
- `score(X,y[,sample_weight])` : 对模型进行评估 , 返回模型的准确率评估结果。
  - `X` : 验证集样本集合。通常是一个 `numpy array` , 每行代表一个样本 , 每列代表一个特征。
  - `y` : 验证集样本的标签集合。它与 `X` 的每一行相对应。
  - `sample_weight` : 每个样本的权重。它与 `X` 的每一行相对应。

## 2. 模型的一些通用参数：

- `n_jobs` : 一个正数 , 指定任务并行时指定的 `CPU` 数量。
 

如果为 `-1` 则使用所有可用的 `CPU` 。
- `max_iter` : 一个整数 , 指定最大迭代次数。
 

如果为 `None` 则为默认值 ( 不同 `solver` 的默认值不同 ) 。
- `tol` : 一个浮点数 , 指定了算法收敛的阈值。

# 一、标签传播算法

## 1. `scikit-learn` 有两个类实现了标签传播算法：

- `LabelPropagation` : 迭代过程 :
  - 执行标签传播 :  $\mathbf{F}^{<t+1>} = \mathbf{P}\mathbf{F}^{<t>}$  。
  - 重置  $\mathbf{F}$  中的标签样本标记 :  $\mathbf{F}_l^{<t+1>} = \mathbf{Y}_l$  , 其中  $\mathbf{F}_l$  表示  $\mathbf{F}$  的前  $l$  行。
- `LabelSpreading` : 迭代过程 :
  - $\mathbf{F}^{<t+1>} = \alpha \mathbf{S}\mathbf{F}^{<t>} + (1 - \alpha)\mathbf{Y}$

## 1.1 LabelPropagation

### 1. `LabelPropagation` 是 `scikit-learn` 提供的 `LabelPropagation` 算法模型 , 其原型为 :

```
class sklearn.semi_supervised.LabelPropagation(kernel='rbf', gamma=20,
n_neighbors=7, alpha=1, max_iter=30, tol=0.001)
```

- `kernel` : 一个字符串，指定距离函数（用于计算边的权重）。可以为下列的值：
  - `'rbf'` : 距离函数为： $\exp(-\gamma|x - y|^2)$ ,  $\gamma > 0$ 。它的计算量较大，且距离矩阵是对称的。
  - `'knn'` : 如果  $x$  是  $y$  的  $k$  近邻，则距离为 1；否则距离为 0。它的计算量较小，且距离矩阵是稀疏矩阵，且距离矩阵不对称。
- `gamma` : 一个浮点数，指定 `rbf` 距离函数的参数。
- `n_neighbors` : 一个整数，指定 `knn` 距离函数的参数。
- `alpha` : 一个浮点数，为折中系数  $\alpha$ 。  
该参数在 `scikit-learn 0.21` 版本中被移除。因为在 `LabelPropagation` 算法中，该参数始终为 0。
- `max_iter` : 一个整数，指定最大的迭代次数。
- `tol` : 一个浮点数，指定收敛的阈值。
- `n_jobs` : 指定并行度。

## 2. 属性：

- `X_` : 一个形状为 `(n_samples, n_features)` 的数组，表示输入数据。
- `classes_` : 一个形状为 `(n_classes,)` 的数组，表示分类问题中，类别种类数组。
- `label_distributions_` : 一个形状为 `(n_samples, n_classes)` 的数组，给出了每个样本的标记在每个类别上的分布。
- `transduction_` : 一个形状为 `(n_samples,)` 的数组，给出每个样本计算出的标记。
- `n_iter_` : 一个整数，给出迭代次数。

## 3. 方法：

- `fit(X, y)` : 训练模型。
- `predict(X)` : 预测样本标记。
- `predict_proba(X)` : 预测每个样本在每个类别上的概率分布。
- `score(X, y[, sample_weight])` : 评估在测试集上的预测准确率。

## 1.2 LabelSpreading

### 1. `LabelSpreading` 是 `scikit-learn` 提供的 `LabelSpreading` 算法模型，其原型为：

```
class sklearn.semi_supervised.LabelSpreading(kernel='rbf', gamma=20,
n_neighbors=7, alpha=0.2, max_iter=30, tol=0.001)
```

参数：参考 `sklearn.semi_supervised.LabelPropagation`。

- 注意：这里的 `alpha` 参数表示折中因子，是有意义的（并不会被删除）。
  - `alpha=0` : 保留所有初始标签信息。
  - `alpha=1` : 修改所有初始标签信息。

### 2. 属性：参考 `sklearn.semi_supervised.LabelPropagation`。

### 3. 方法：参考 `sklearn.semi_supervised.LabelPropagation`。

