一、字典学习

1.1 DictionaryLearning

1、DictionaryLearning用于字典学习，原型为sklearn.decompositon.DictionaryLearning(n_components=None, alpha=1, max_iter=1000, tol=1e-08, fit_algorithm='lars', transform_algorithm='omp', transform_n_nonzero_coefs=None, transform_alpha=None, n_jobs=1, code_init=None, dict_init=None, verbose=False, split_sign=False, random_state=None)

- n_components：一个整数，指定了字典大小k。
- alpha：一个浮点数，指定了L1正则化项的系数 λ，它控制了稀疏性。
- max_iter：一个整数，指定了最大迭代次数。
- tol：一个浮点数，指定了收敛阈值。
- fit_algorithm：一个字符串，指定了求解算法。'lars'使用least angle regression算法；'cd'使用coordinate descent算法。
- transform_algorithm：一个字符串，指定了数据转换的方法。'lasso_lars'使用Lars算法；'lasso_cd'使用coordinate descent算法；'lars'使用least angle regression算法；'omp'使用正交匹配算法；'threshold'通过字典转换后的坐标中，小于transform_alpha的特征的值都设为0。
- transform_n_nonzero_coefs：一个整数，指定解中每一列中非零元素个数，默认为0.1*n_features。
- transform_alpha：一个浮点数，默认为1.0。若算法为lasso_lars或lasso_cd指定L1正则化项的系数；若算法为threshold指定特征为0的阈值；若算法为omp指定重构误差的阈值，此时覆盖transform_n_nonzero_coefs参数。
- n_jobs：一个整数，指定并行性。
- code_init：一个数组，指定初始编码，用于字典学习算法的热启动。
- dict_init：一个数组，指定初始字典，用于字典学习算法的热启动。
- verbose：一个整数，控制输出日志。
- split_sign：一个布尔值，指定是否拆分系数特征向量为其正向值和负向值的拼接。
- random_state：一个整数或一个RandomState实例或None，指定随机数种子。

2、属性有components_、error_、n_iter_

- components_：一个数组，存放学到的字典。
- error_：一个数组，存放每一轮迭代的误差。
- n_iter_：一个整数，存放迭代的次数。

3、方法有fit、transform、fit_transform

- fit(X, y)：学习字典。
- transform(X)：根据学到的字典进行编码。
- fit_transform(X, y)：学习字典并执行字典编码。

1.2 MiniBatchDictionaryLearning

1、MiniBatchDictionaryLearning也是字典学习，主要用于大规模数据。它每次训练一批样本，然后连续多次训练，原型为sklearn.decomposition.MiniBatchDictionaryLearning(n_components=None, alpha=1, n_iter=1000, fit_algorithm='lars', n_jobs=1, batch_size=3, shuffle=True, dict_init=None, transform_algorithm='omp', transform_n_nonzero_coefs=None, transform_alpha=None, verbose=False, split_sign=False, random_state=None)

- n_iter：一个整数，指定了总执行迭代数量。
- batch_size：一个整数，指定了每次训练时的样本数量。
- shuffle：一个布尔值，指定在训练每一批样本之前，是否对该批次样本进行混洗。
- 其余参数参考DictionaryLearning。

2、属性有components_、inner_stats、n_iter_

- components_：一个数组，存放学到的字典。
- inner_stats：数组的元组，存放算法的中间状态。
- n_iter_：一个整数，存放迭代的次数。

3、方法有fit、transform、fit_transform、partial_fit

- fit(X, y)：学习字典。
- transform(X)：根据学到的字典进行编码。
- fit_transform(X, y)：学习字典并执行字典编码。
- partial_fit(X[, y, iter_offset])：只训练一个批次的样本。

二、PipeLine

2.1 PipeLine

1、sklearn中的流水线流程通常为：

- 通过一组特征处理estimator来对特征进行处理（如标准化、正规化）。
- 通过一组特征提取estimator来提取特征。
- 通过一个模型预测estimator来学习模型并执行预测。
- 除了最后一个estimator外，其余estimator必须提供transform方法用于执行数据变换（如归一化、正则化、特征提取等）。

2、PipeLine将多个estimator组成流水线，原型为sklearn.pipeline.Pipeline(steps)

- steps：一个列表，元素为(name, transform)元组。name是estimator的名字用于输出和日志；transform是estimator。

3、属性有name_steps

- name_steps：一个字典。keys为steps中各元组的name元素，values为steps中各元组的transform元素。

4、方法有fit、transform、fit_transform、inverse_transform等

- fit(X, y)：启动流水线，依次对各个estimator（除最后一个）执行.fit和.transform方法转换数据；对最后一个estimator执行.fit方法训练学习器。
- transform(X)：启动流水线，依次对各个estimator（包括最后一个）执行.fit和.transform方法转换数据。
- fit_transform(X, y)：启动流水线，依次对各个estimator（除最后一个）执行.fit和.transform方法转换数据；对最后一个estimator执行.fit_transform方法准换数据。
- inverse_transform(X)：将转换后的数据逆转换成原始数据。要求每个estimator都实现了.inverse_transform方法。
- predict(X)/predict_log_proba(X)/predict_proba(X)：将X进行数据转换后，用最后一个学习器来预测。
- score(X, y)：将X进行数据转换后，训练最后一个estimator，并对最后一个estimator评分。