|  |  |  |
| --- | --- | --- |
| **common parameters** | | |
| parameter | default | description |
| train\_size: float | 0.75 | the size of the training dataset |
| label\_rates: float | [0.1] | nargs = '+'  the size of the labeled data = train\_size \* label\_rate |
| experiment\_num: int | 10 | number of experiments |
| data\_dir: str | None | if data\_dir == None: read data from sklearn.datasets package  else: read csv data from data directory |
| log\_dir: str | '\log\*method\_*  *name\_time*' | log directory |
| avg\_types: str | ['weighted'] | nargs = '+'  choices = ['micro', 'macro', 'weighted']  micro => micro average  macro => macro average  weighted => weighted macro average |
| save\_pred: bool | False | action = 'store\_true'  whether to save prediction and number of unlabeled |
| **self\_training** | | |
| parameter | default | description |
| groups: str | 'NB' | nargs = '+'  choices = ['NB', 'SVM', 'RF', 'AdaBoost', 'KNN', 'DT']  the type of base classifier |
| confidence\_thresholds: float | [0.1] | nargs = '+'  the confidence threshold which decide whether to add unlabeled data |
| use\_unlabeled\_pool: bool | False | action = 'store\_true'  whether to use the unlabeled pool |
| pool\_size: int | 75 | the size of the unlabeled pool |
| k: int | 30 | number of iterations of self-training algorithm |
| data\_pre\_type: str | 'category\_and\_numeric' | choices = ['all\_category', 'category\_and\_numeric']  category => 10 equal-width bins  numeric => log |
| **co\_training** | | |
| parameter | default | description |
| split\_types: str | 'entropy\_split' | nargs = '+'  choices = ['random\_split', 'entropy\_split', 'entropy\_hill', 'random\_hill']  the method which split the single view(feature subset) into two views |
| split\_small: bool | False | action = 'store\_true'  whether split view only based on labeled data or whole dataset |
| groups: str | ['NB'] | nargs = '+'  choices = ['NB', 'SVM', 'RF', 'AdaBoost', 'KNN', 'DT']  the two base classifiers use the same type of classifier |
| use\_unlabeled\_pool: bool | False | action = 'store\_true'  whether to use the unlabeled pool |
| pool\_size: int | 75 | the size of the unlabeled pool |
| k: int | 30 | number of iterations of self-training algorithm |
| data\_pre\_type: str | 'all\_category' | choices = ['all\_category']  category => 10 equal-width bins |
| **tri\_training** | | |
| parameter | default | description |
| groups: str | ['NB'] | nargs = '+'  choices = ['NB', 'SVM', 'RF', 'AdaBoost', 'KNN', 'DT']  the three base classifiers use the same type of classifier |
| data\_pre\_type: str | 'category\_and\_numeric' | choices = ['all\_category', 'category\_and\_numeric']  category => 10 equal-width bins  numeric => log |
| **multi\_train** | | |
| parameter | default | description |
| groups: str | ['NB\_AdaBoost\_DT'] | nargs = '+'  the type of base classifiers can be different  use \_ to separate the classifier type |
| confidence\_thresholds: float | [0.1] | nargs = '+'  the confidence threshold which decide whether to add unlabeled data |
| feature\_pre: str | None | choices = [None, 'PCA']  use principal component analysis for transforming the features |
| data\_pre\_type: str | 'category\_and\_numeric' | choices = ['all\_category', 'category\_and\_numeric']  category => 10 equal-width bins  numeric => log |
| **two\_phase\_auto\_fit** | | |
| parameter | default | description |
| groups: str | ['NB\_AdaBoost\_DT'] | nargs = '+'  the type of base classifiers can be different  use \_ to separate the classifier type |
| use\_auto\_select\_threshold: bool | False | action = 'store\_true'  whether to use auto select threshold when predict unlabeled data |
| confidence\_thresholds: float | [0.1] | nargs = '+'  the confidence threshold which decide whether to add unlabeled data |
| feature\_pre: str | None | choices = [None, 'PCA']  use principal component analysis for transforming the features |
| data\_pre\_type: str | 'category\_and\_numeric' | choices = ['all\_category', 'category\_and\_numeric']  category => 10 equal-width bins  numeric => log |