SUPPORT VECTOR CLASSIFIER API SUMMARY

Krish Jain J069

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and the regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

SVC, NuSVC and LinearSVC are classes capable of performing binary and multi-class classification on a dataset.

sklearn.svm.SVC

```
class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale',
coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=- 1,
decision_function_shape='ovr', break_ties=False, random_state=None
```

PARAMETERS:

- C:float, default=1.0- Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
- kernel{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}, default=’rbf’- Specifies the kernel type to be used in the algorithm. It must be one of ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’ or a callable. If none is given, ‘rbf’ will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n_samples, n_samples).

- degree:int, default=3- Degree of the polynomial kernel function (‘poly’). Ignored by all other kernels.
- gamma:{‘scale’, ‘auto’} or float, default=’scale’
    - if gamma='scale' (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma,
    - if ‘auto’, uses 1 / n_features.
- coef0: float, default=0.0- Independent term in kernel function. It is only significant in ‘poly’ and ‘sigmoid’.
- shrinking: bool, default=True
- probability: bool, default=False
- tol:float, default=1e-3
- cache_size: float, default=200
- class_weight: dict or ‘balanced’, default=None
- verbose:bool, default=False
- max_iter:int, default=-1
- decision_function_shape:{‘ovo’, ‘ovr’}, default=’ovr’
- break_ties:bool, default=False
- random_stateint, RandomState instance or None, default=None

ATTRIBUTES:

- class_weight_:ndarray of shape (n_classes,)
- classes_:ndarray of shape (n_classes,)
- coef_:ndarray of shape (n_classes * (n_classes - 1) / 2, n_features)
- dual_coef_:ndarray of shape (n_classes -1, n_SV)
- fit_status_:int
- intercept_:ndarray of shape (n_classes * (n_classes - 1) / 2,)
- support_:ndarray of shape (n_SV)
- support_vectors_:ndarray of shape (n_SV, n_features)
- n_support_:ndarray of shape (n_classes,), dtype=int32
- Number of support vectors for each class.
- probA_:ndarray of shape (n_classes * (n_classes - 1) / 2)
- probB_:ndarray of shape (n_classes * (n_classes - 1) / 2)
- shape_fit_:tuple of int of shape (n_dimensions_of_X,)