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.
- Kernel: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}, default=’rbf’
- degree: int, default=3
- gamma: {‘scale’, ‘auto’} or float, default=’scale’
- coef0: float, default=0.0
- 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_state: int, 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
- 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,)

sklearn handling of SVM

• 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 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 regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- The support vector machines in scikit-learn support both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input.
- However, to use an SVM to make predictions for sparse data, it must have been fit on such data.
- For optimal performance, use C-ordered numpy.ndarray (dense) or scipy.sparse.csr_matrix (sparse) with dtype=float64.