DEVYANI BEOHAR

J068

SVC (Support Vector Classificatin)

Code:

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*)

C*float, default=1.0*
Regularization parameter

kernel*{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf'*
Specifies the kernel type to be used in the algorithm.

degree*int, default=3*
Degree of the polynomial kernel function ('poly').

gamma*{'scale', 'auto'} or float, default='scale'*
Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

- 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.

shrinking*bool, default=True*
Whether to use the shrinking heuristic.

probability*bool, default=False*
Whether to enable probability estimates.

tol*float, default=1e-3*
Tolerance for stopping criterion.

cache_size*float, default=200*
Specify the size of the kernel cache (in MB).

class_weight*dict or 'balanced', default=None*
Set the parameter C of class i to class_weight[i]*C for SVC.

verbose*bool, default=False*
Enable verbose output.

max_iter*int, default=-1*
Hard limit on iterations within solver, or -1 for no limit.

decision_function_shape*{'ovo', 'ovr'}, default='ovr'*
Whether to return a one-vs-rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one ('ovo')

break_ties*bool, default=False*
If true, decision_function_shape='ovr', and number of classes > 2

random_state*int, RandomState instance or None, default=None*
Controls the pseudo random number generation for shuffling the data for probability estimates.

SVM (Support Vector Machines)

In machine learning, **support-vector machines** (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition. The main objective is to segregate the given dataset in the best possible way. They 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 (see Scores and probabilities, below).