## Objectives

This module will walk you through the main idea of how support vector machines construct hyperplanes to map your data into regions that concentrate a majority of data points of a certain class. Although support vector machines are widely used for regression, outlier detection, and classification, this module will focus on the latter.

* Identify common supervised machine learning algorithms
* Describe and use support vector machines for classification
* Realize the importance of the kernel trick for non-linear classification
* Build support vector machines models with sklearn

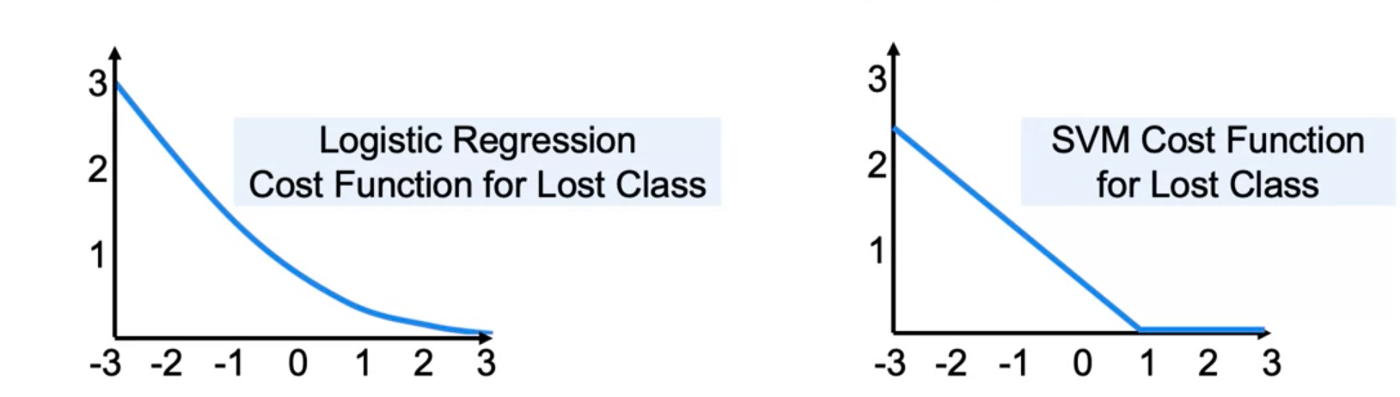
Maximize the region between classes

Logistic regression vs. Svm

The dots on graph are our support vectors

Logistic regression cost function for lost class

Svm cost function for lost class



Outlier sensitivity in svms

Regularization in svms

Linear svm: syntax

#impor tth eclass containing the classification method

From sklearn.svm import linearsvc

# create an instanc eof the class

LinSVC = LinearSVC (penalty=’12’, c=10, 0)

SVM Kernels

Non-linear data can be made linear with higher dimensionality

Transform datat so it is linearly separable

Svm gaussian kernel

Approach 1:

* Create higher order features to transform the data

Approach 2:

* Transform the spae to a different coordinate system.

Radial basis function (rbf)

Syntax

#import the class contain the classification mehtod

From sklearn.svm import SVC

#create an instance of the class

RbfSVC = SVC(kernel=’rbf’, gamma=1.0, c=10.0)

# fit instance on the data and then predict the expected value

Ml workflow

Problem

- svms with rbf kernels are very slow to trian with lots of features or data

Data collection

* Construct approximate kernel map with sgd using nystroem or rbf smapler
* Fit a linear classifier

|  |  |  |
| --- | --- | --- |
| features | data | Model choices |
| Many 10k feautres | Small 1k rows | Simple, logistic or linear svc |
| Few < 100 features | Medium 10k rows | Svc with rbf |
| Few < 100 features | Many > 100k points | Add features, logistic, linearSVC, or Kernel approx. |
|  |  |  |

Faster kernel transformations syntax

From sklearn.kernel\_approximation import Nystroam

NystroemSVC = Nystroem ( kernel=’rnbf’, gamma=1.0, n\_componenets=100)

X\_train = Nystroemsvc.fit\_transform(X\_train)

X\_test = NystroemSVC.transform(X\_train)

From sklearn.kernel\_approximation import RBFsampler

RbfSample=RBF(gamma=1.0, n\_componenets=100)

SVM Notebook pt.1

Plot correlations

Correlations.plot(king=’bar’, color=colors[0])

Corrleations.map(abs).sort\_values()

Train on the entire dataset

SVM pt.2

Linear decision boundary

Fit levc on all of our dataset

Np.arrange

Contra plot

SVM Pt. 3

## End of module review: Support Vector Machines

### **Support Vector Machines**

The main idea behind support vector machines is to find a hyperplane that separates classes by determining decision boundaries that maximize the distance between classes.

When comparing logistic regression and SVMs, one of the main differences is that the cost function for logistic regression has a cost function that decreases to zero, but rarely reaches zero. SVMs use the Hinge Loss function as a cost function to penalize misclassification. This tends to lead to better accuracy at the cost of having less sensitivity on the predicted probabilities.

Regularization can help SVMs generalize better with future data.

By using gaussian kernels, you transform your data space vectors into a different coordinate system, and may have better chances of finding a hyperplane that classifies well your data.SVMs with RBFs Kernels are slow to train with data sets that are large or have many features.