Chapter 9 Support Vector Machines

9.1. Maximal Margin Classifier

9.1.1 What Is a Hyperplane?

In a *p*-dimensional space, a hyperplane is a flat affine subspace of dimension .

A hyperplane in two dimensions is defined by:

A picture containing object, clock, building

Description automatically generated

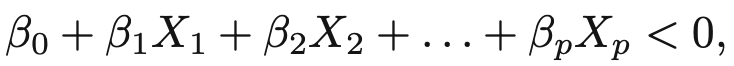
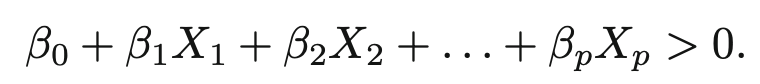
for parameters 🡪 is a point on the hyperplane

Hyperplane in a *p*-dimension setting:



If in *p*-dimensional space satisfies the equation, then lies on the hyperplane

If:



Then lies on one side or the other side of the hyperplane.

9.1.2 Classification Using a Separating Hyperplane

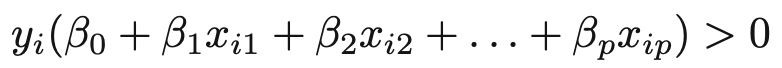
A separating hyperplane has the property that

**

And



Equivalently, a separating hyperplane has the property that



For all

1. A test observation is assigned a class depending on which side of the hyperplane it is located,
2. If is far from zero, this means that is far from the hyperplane, so we can be confident about our class assignment for
3. If is close to zero, then is located near the hyperplane, so we are less certain about the class assignment of
4. A classifier based on a separating hyperplane leads to a linear decision boundary

9.1.3 The Maximal Margin Classifier

1. Maximal margin hyperplane(Optimal separating hyperplane):

Separating hyperplane that is farthest from the training observations

2. Margin: The smallest perpendicular distance from the observation to the hyperplane🡪Maximal Margin Hyperplane is the separating hyperplane for which the margin is the largest

3. Maximal Margin Classifier: Classify a test observation based on which side of the maximal margin hyperplane it lies

4. Support vectors: A subset of the observations that affect the maximal margin hyperplane

a. A movement to any of the support vectors changes the maximal margin hyperplane

b. Movement to any other observations would not affect the separating hyperplane.

9.1.4 Construction of the Maximal Margin Classifier

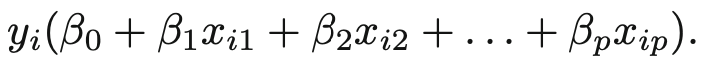
For a set of *n* training observations, and associated class labels

Maximal Margin Classifier is the solution to the optimization problem:

A picture containing knife

Description automatically generated

1. (9.11) guarantees that each observation will be on the correct side of the hyperplane, provided that *M* is positive
2. The perpendicular distance from the ith observation to the hyperplane is given by,



1. (9.10) and (9.11) ensures that each observation is on the correct side of the hyperplane and at least a distance *M* from the hyperplane

9.1.5 The Non-Separable Case

1. Maximal Margin Classifier is valid for classification if a separating hyperplane exists

2. Support Vector Classifier: The generalization of the maximal margin classifier to the non-separable case

9.2 Support Vector Classifiers

9.2.1 Overview of the Support Vector Classifier

Drawbacks of Maximal Margin Classifier:

1. The maximal margin hyperplane is extremely sensitive to a change in a single observation
2. May lead to overfitting in the training data

Support Vector Classifier(soft margin classifier):

1. Allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.
2. Observations on the wrong side of the hyperplane correspond to training observations that are misclassified by the support vector classifier.

9.2.2 Details of the Support Vector Classifier

The support vector classifier classifies a test observation depending on which side of the hyperplane it lies

The hyperplane is chosen as the solution to the optimization problem:

A close up of a piece of paper

Description automatically generated

Where is a nonnegative tunning parameter, is the width of the margin, are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane

\*We classify a test observation based on which side of the hyperplane it lies

Slack Variables:

1. Tells us where the ith observation is located, relative to the hyperplane and relative to the margin.
   1. If , ith observation is on the correct side of the margin
   2. If , then the ith observation is on the wrong side of the margin🡪ith observation has violated the margin
   3. If , the it is on the wrong side of the hyperplane

tunning parameter:

1. Determines the number and severity of the violations to the margin
2. If , there is no budget for violations to the margin, and it must be the case that 🡪 the problem becomes maximal margin hyperplane optimization problem
3. For , no more than observations can be on the wrong side of the hyperplane
4. As the budget increases, we become more tolerant of the violations to the margin
5. is a tunning parameter and is generally chosen via cross-validation
6. controls the bias-variance trade-off of the statistical learning technique
   1. Lower is a more flexible fit
   2. Higher is a less flexible fit🡪More bias, less variance

Property of Support Vector Classifier:

Only observations that lie on the margin or that violate the margin will affect the hyperplane

1. An observation that lies strictly on the correct side of the margin does not affect the support vector classifier

Support Vectors: Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as support vectors.

1. controls the bias-variance trade-off of the support vector classifier
2. When is large, then the margin is wide, many observations violate the margin🡪There are many support vectors🡪Many observations are involved in determining the hyperplane🡪 Low variance but high bias
3. When is smack, there will be fewer support vectors and hence the resulting classifier will have low bias and high variance

Support Vector Classifier depends only on a subset of the training observations🡪Robust to the behavior of the observations that are far away from the hyperplane

9.3 Support Vector Machines

9.3.1 Classification with Non-linear Decision Boundaries

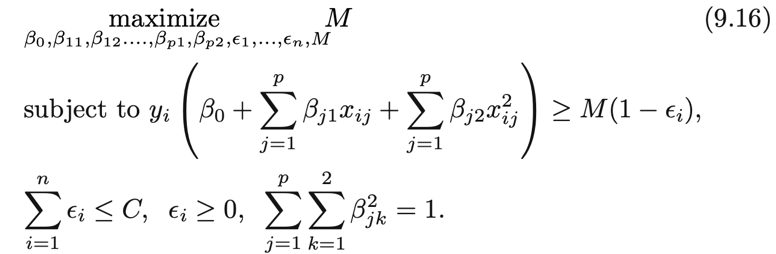
We could address the problem of possibly non-linear boundaries between classes by enlarging feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.

1. Fir a support vector using 2*p* features

A close up of a logo

Description automatically generated

The optimization problem becomes🡪



There are many possible ways to enlarge the feature space, and that unless we are careful, we could end up with a huge number of features.

9.3.2 The Support Vector Machine

SVM is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using kernels.

Inner products of two observation , is given by,

A close up of a clock

Description automatically generated

The linear support vector can be represented as

A picture containing object, clock

Description automatically generated

Where there are parameters one per training observation

1. To estimate the parameters and , all we need are inner products between all pairs of training observations
2. is non zero only for the support vector in the solution🡪

A picture containing object, clock

Description automatically generated

In order to evaluate the function f(x), we need to compute the inner product between the new point x and each of the training points

1. A kernel function is a function that quantifies the similarity of two observations

A drawing of a face

Description automatically generated

1. Linear kernel quantifies the similarity of a pair of observation using Pearson(standard) correlation

A close up of a clock

Description automatically generated

1. Polynomial kernel of degree d:
   1. Leads to a much more flexible decision boundary
   2. Fitting a support vector classifier in a higher-dimensional space involving polynomials of degree d, rather than in the original feature space
2. When the support vector classifier is combined with a non-linear kernel, the resulting classifier is a support vector machine

A picture containing object, clock

Description automatically generated

1. Radial Kernel:

A picture containing object, clock

Description automatically generated A close up of a map

Description automatically generated

where is a positive constant

1. Radial kernel has very local behavior, in the sense that only nearby training observations have an effect on the class label of a test observation.
2. As increases, the fit becomes more non-linear
3. Kernel is less computationally intensive than enlarged feature space.

9.4 SVMs with More than Two Classes

9.4.1 One-Versus-One Classification

When performing a classification using SVMs, and there are classes, a *one-versus-one or all-pairs* approach constructs SVMS, each of which compares a pair of classes.

9.4.2 One-Versus-All Classification

For a total of classes, fit SVMs, each time comparing one of the classes to the remaining classes

1. By assigning an observation for which *M* is largest, this amounts to a high level of confidence that the test observation belongs to the kth class rather than any other class.

9.5 Relationship to Logistic Regression

Fitting the support vector classifier can be rewritten as

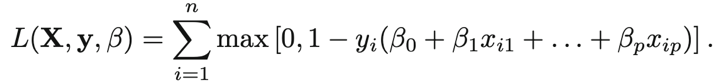
A close up of a logo

Description automatically generated

* + 1. is the tuning parameter🡪Larger allows less flexible model, smaller allows more flexible model
    2. ” Form

1. Loss function quantifies the extent to which the model fits the data
2. The effect of penalty function is controlled by

Hinge Loss:



1. Only support vectors plays a role in the classifier obtained; observations on the correct side of the margin do not affect it
2. Logistics regression is small for observations that are far from the decision boundary
3. When classes are well separated, SVMs tend to behave better than logistic regression; in more overlapping regimes, logistic regression is often preferred.

Support Vector Regression:

Only residuals larger in absolute value than some positive constant attribute contribute to the loss function.

9.6 Lab: Support Vector Machines

Svm(): can be used to fit a support vector classifier when the argument kernel=”linear”

1. Cost argument allows us to specify the cost of violation🡪 small cost allows wide margin, large cost restrict the margin to be narrower

Generating data sets that are not linearly separable

A picture containing people

Description automatically generated

Applying svm() function:

A picture containing knife, table

Description automatically generated

1. Scale=TRUE/FALSE: Tells svm() function to scale each feature

Plot():

1. Output of the call to svm()
2. the data used in the call to svm()



A close up of a logo

Description automatically generated1. The decision boundary between the two classes is linear

2. The support vectors are plotted as crosses and the remaining observations are plotted as circles.

Index gives the identities of the support vectors

A picture containing clock

Description automatically generated

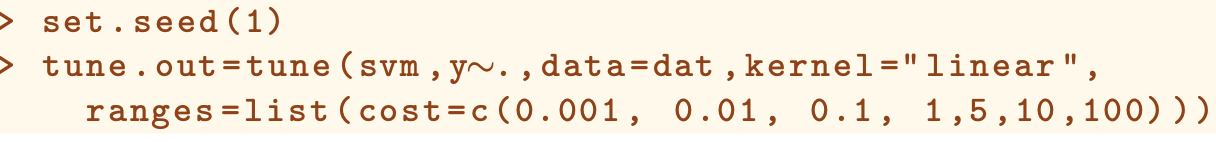
Summary(): Obtain the basic information about the support vectors

A screenshot of a cell phone

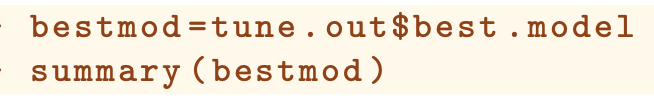
Description automatically generated

Svm() function does not explicitly output the coefficients of the linear decision boundary when the support vector classifier is fit, nor does it output the width of the margin

Tune(): Used to perform cross validation



1. We want to compare SVMs with a linear kernel, using a range of values of the cost parameter
2. Use summary() to access the cross-validation errors for each of these models
3. The information of the best model can be obtained using



1. Use predict function

A screenshot of a cell phone

Description automatically generated

9.6.2 Support Vector Machine

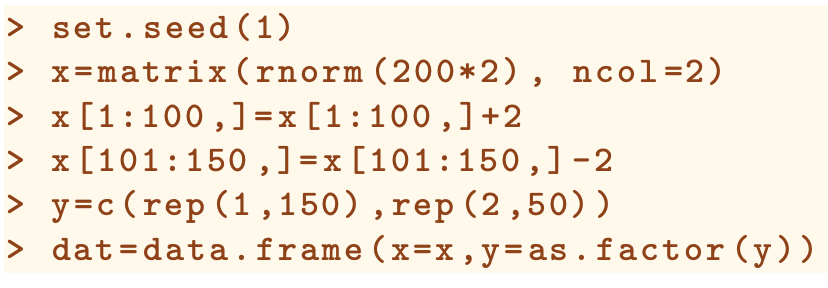
Svm()

Polynomial kernel:

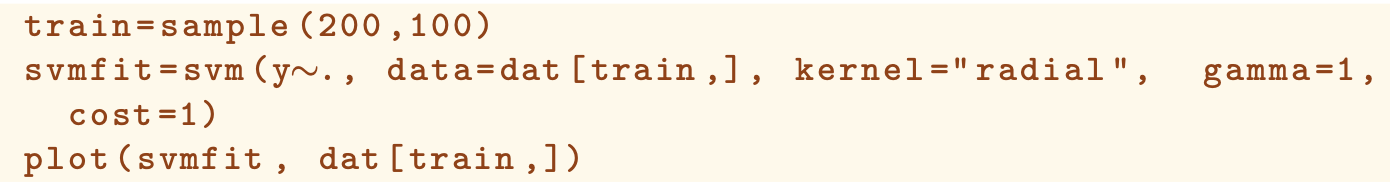
Kernel=”polynomial”, degree=, specify a degree for the polynomial kernel

Radial kernel:

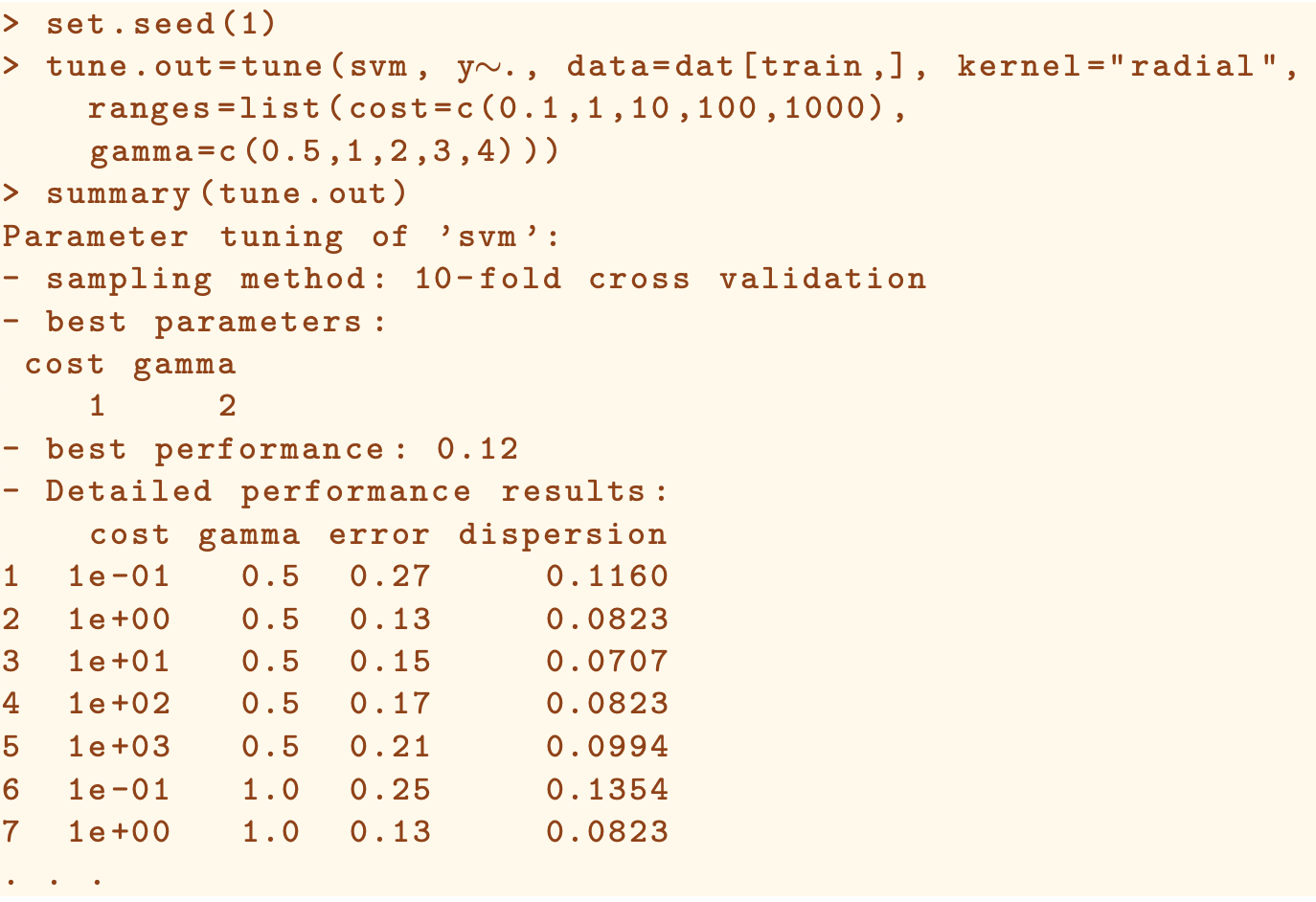
Kernel=”radial”, gamma=, specify the value of the radial basis



Fit the training data into a radial kernel:



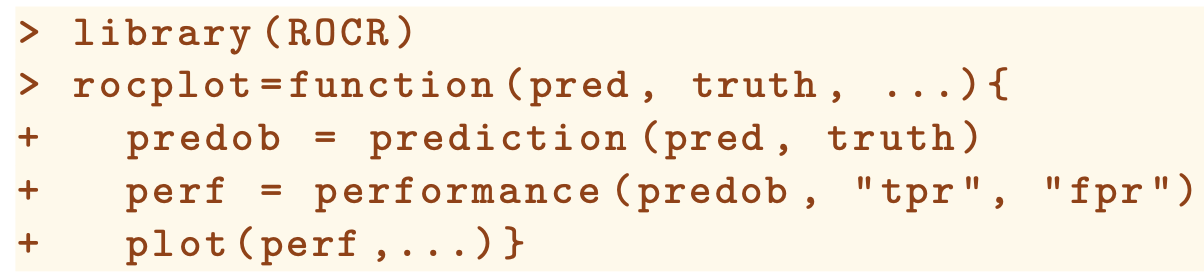
We can perform cross validation using tune to find the best cost and gamma.



9.6.3 ROC Curves

ROCR package can be used to produce ROC curves

1. Write a short function to plot an ROC curve given a vector containing a numerical score for each observation, pred, and a vector containing the class label for each observation, truth

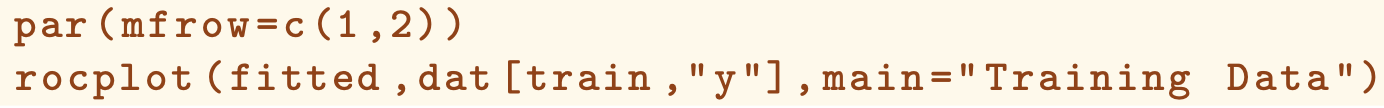


Svm() output class labels for each observation by default

1. To obtain the fitted values for each observation, we use decision.values=True when fitting svm
2. The predict() function will output the fitted values

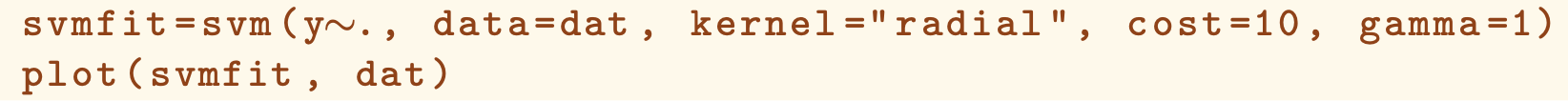
A screenshot of a cell phone

Description automatically generated



9.6.4 SVM with Multiple Classes

If the response is a factor containing more than two levels, then the svm() function will perform multi-class classification using the one-versus-one approach



1. If the response vector that is passed in to svm() is numerical rather than a factor, the svm() function performs support vector regression