Introduction to Statistical Classification: Support Vector Machines

Jon Ashbrock

March 14, 2018

1 Introduction

In this lab students will:

1. Understand what is meant by classification in a statistical sense
2. Understand many basic terms in machine learning and data science.
3. Understand and implement the theory of a support vector machine

2 Classification as a Statistics Problem

In the previous two labs we saw how to use random samples of data to predict future observations of data using the method of regression modeling. This method works well when the variable we are trying to predict (recall, this is called the dependent variable) is continuous. That is, the variable can be any number in a certain range. However, many variables we are interested in predicting are in fact discrete: they fall into one of a few categories. A model that predicts a discrete variable is typically called a classifier (or sometimes a classification algorithm, depending on context).

Classification is arguably the main area of research in modern machine learning theory because of how universal the classification problem is. To see how universal it is I will give a few examples. First, if I am an investor and am deciding which stocks to invest in, I could model this as a classification problem into the categories "stock will make me money" and "stock will lose

me money". A second example can be seen in the new iPhone with the facial recognition unlock feature. The phone runs an algorithm to classify the face it sees as "this iPhone's owner" and "not this iPhone's owner". A third example is the Amazon suggested items menu. When purchasing something on Amazon, the website shows you items that it thinks you may be interested in. When determining which items to show, Amazon runs a classification algorithm into the categories "user may be interested in this item" and "user is likely not interested in this item". We can see that classification problems are truly universal in their applications.

The important question is then: how do we mathematically represent categorical variables? We will see a multitude of ways to do this. The key difference is whether we are using continuous or discrete variables to do the prediction. In general, though, the problem of statistical classification boils down to minimizing the error in prediction. The next three or four classes will focus on a variety of different methods. Today we discuss the general idea of classification when we have continuous independent variables.

3 Dictionary of Machine Learning Terms

The following terms will be referred to freely throughout the class and the notes and are vital to understanding the importance and usage of particular types of algorithms/methods. Note that these definitions are not in general given in the precise mathematical sense.

Definition. *A **data set** or **sample** is a finite set of the form* $\{(x_i, y_i)\}_{i=1}^n$ *where* x_i *are independent variables and the* y_i *are the dependent or predicted variables.*

In the above definition, the points x_i can be a single real number, categorical information, or even a collection of n independent variables.

Definition. *The **dimension** of an independent variable is the number of distinct pieces of information given in our independent variable.*

This is also sometimes referred to as the dimension of our data. As an example, if we want to use height, weight, gender, age, and wrist size to predict body fat percentage then the dimension of our independent variable is 5 because we are using five distinct variables to do our prediction.

Definition. *The **parameters** of a model are those aspects of the model which we are free to choose in order to fit it to our data.*

As an example, the parameters of the linear regression model we have previously learned are the coefficients β_0, β_1 which are the slope and intercept of our line.

Definition. *A **loss function** or **error function** is a function with a positive real number output that measures how good of a predictor our model is. The inputs to the function are the parameters of the model.*

In the linear regression case, our loss function was the sum of the square errors. Notice that this quantity is a function $f(\beta_0, \beta_1)$ which has a positive real number output. In general, the choice of loss function is incredibly important.

The model we choose is the one which minimizes the loss function. If the model is complex (aka has many parameters) or the data is high dimensional, then the true loss minimizer may be impossible to compute and so we can use iterative algorithms to approximate the minimizer.

Definition. *Suppose we have a model ℓ which categorizes the data points into n categories $\{c_1, c_2, \ldots c_n\}$. The **confusion matrix** of ℓ is the nxn matrix where the entry $x_{i,j}$ is the number of observations whose true category is c_j and whose predicted category is c_i.*

The confusion matrix is a way of seeing how well a categorical model performs on the training set. Better models typically have most of the entries on the main diagonal of the confusion matrix.

4 Support Vector Machines

Throughout this section we will refer to a support vector machine as an SVM. A support vector machine is a classification model used to classify a data set where the independent variables are continuous. We begin by discussing the case where we wish to classify our data into one of two categories and where our data is 2 dimensional.

In this case, the support vector machine solves the following problem. Let $\{(x_i, y_i)\}_{i=1}^n$ be a data set where each $x_i = (x_{i,1}, x_{i,2})$ is a point in the Cartesian plane and each y_i is either a 0 or a 1 (with each representing

membership in one of the two categories we are considering). Th support vector machine computes the equation $y = mx + b$ of a line in the plane. The classification of points is then done by determining whether the points lies above or below the line. The line is then computed by finding the line which:

1. Minimizes the error in classifying the training set
2. Maximizes the distance between the line and the nearest point to the line

How much emphasis the loss function puts on each of these is outside the scope of the course. The distance in part number two above is called the "margin"

Discussion 1. *Why should we maximize the distance between the line and the nearest point? Consider the case of 2 points.*

Intuition: Said simply, the support vector machine uses a line to separate the plane into two regions with each corresponding to membership in one of our desired categories. To get to the idea of a general support vector machine we need to make three generalizations:

1. Our data may be higher dimensional so we need to compute the higher dimensional equivalent of a line. This is called a **hyperplane**
2. Our data may not be able to be separated with a flat hyperplane. Rather, we may wish to identified curved regions in space.
3. We may wish to classify into more than two categories.

Point number one does not require any mathematical machinery, we just recall that a hyperplane in $n-$dimensional Euclidean space is given by the equation $y = \beta_0 + \beta_1 x_1 + \ldots \beta_n x_n$. The other two points are more mathematically complex and we will discuss them in their own subsections

4.1 Separating by curved regions

Separating by curved regions is done by artificially increasing the dimension of the data to allow for separation by a hyperplane. Here is how it is done: Suppose we have two dimensional data which we wish to identify the region

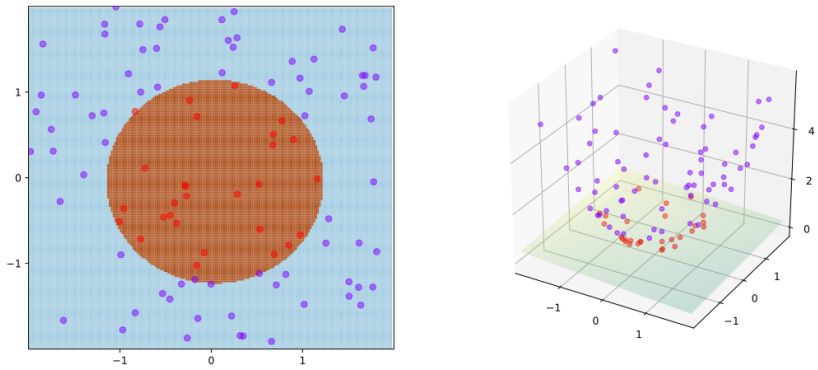

Figure 1: Figure taken from Wikipedia. The right picture is the "depressed" trampoline

$x_1^2 + x_2^2 < 1$ by using an SVM. To do this we transform a point (x_1, x_2) into a three-dimensional point $(x_1, x_2, x_1^2 + x_2^2)$. Now we simply need to find the equation of the hyperplane which separates space in the desired way.

Here is another way to visualize this transformation. Suppose we draw a circle on a trampoline which represents the region to be identified. However, we can only use straight line cuts to identify this region. When the trampoline is flat, this is impossible. However now suppose we push down in the exact center of this circle. Now use a flat plane to cut out the exact circle we want and then let the trampoline go back to resting position. We only used a plane to do the "cutting" but we have successfully identified a curved region.

One more thing to note is that our model will not necessarily separate the existing data perfectly. Therefore most models use so-called "soft-margin" classifiers which allow for a certain proportion of points to lie within the margin. These are called the support vectors in the model.

4.2 Classification into more than 2 categories

The support vector machine uses a hyper-plane to separate space into two regions. However not every classification problem involves exactly two categories. A simple example is having a computer translate written text into typed text. This requires (at the minimum) 26 different categories. The most widely used method is to use a "voting" scheme to determine the cor-

rect category.

Suppose we have k categories to classify objects into. Then there are $\binom{k}{2} = \frac{k(k-1)}{2}$ distinct pairs of categories. For each pair we compute its own SVM. Then for a new object to classify, we classify it using each of our $\binom{k}{2}$ support vector machines. Then we count how many times our object was classified into each category and the category with the highest total is the classification we select. It is called a voting scheme because each individual SVM gets to cast a "vote" for one of two categories and the category with the most votes wins. Note that two-way ties can be broken by the SVM corresponding to the particular pair of categories to do the classification. Multi-way ties are quite unlikely and other ways of dealing with this are outside the scope of this course.

5 Implementation of SVM in R

Discussion 2. *Develop an SVM using the iris dataset*

The package we need to use is called "e1071". The documentation for the svm function is available here:

$https://www.rdocumentation.org/packages/e1071/versions/1.6-8/topics/svm$

In particular, if x is a data frame with n rows and y is a *factor*-type variable corresponding to the observations in x, then the following code will compute the SVM model on this data:

```
model <- svm(x,y)
```

A factor-type variable is a one which is interpreted as being categorical. You can convert strings and numbers to factors by using the "as.factor" command in **R**

The remainder of this section will discuss a few useful functions to use the SVM model we have computed. The "summary" function works like before but provides information relevant to the model you built. More importantly is the "predict" function. This function is designed to work with most statistical models we will design. The only required input is the variable containing the statistical model. That is, I could use the function in this way:

```
model <-svm(x,y)
predictions <- predict(model)
```

In this case, the variable "predictions" will contain the predicted categories for the data observations stored in x. However, typically we are interested in using our model to predict categories for new data points. We can do this by providing a dataframe to the predict function as:

```
new_predictions <- predict(model, newdata= d)
# d is the dataframe containing the new data
```

.

It is absolutely necessary that this new dataframe has the same column headings as the frame x, otherwise the model cannot interpret which column is supposed to correspond to which predictor.

Lastly, the "table" function will be used in our case to generate the so-called confusion matrix of our model. The confusion matrix was defined in the definitions section above. Recall that in our setting the variable y contains the true categories while "predictions" contains the predicted categories. The following code then will generate the confusion matrix

```
table(predictions,y)
```