Post 01 - Linear Algebra in Data Science

Sandeep Tiwari

10/31/2017

Linear Algebra in Data Science and Machine Learning

Introduction

What is Linear Algebra and why is it relevant? Well, linear algebra is the branch of mathematics concerning vector spaces and linear mappings between such spaces. Simply put, linear algebra is math dealing with straight objects in space. Linear algebra can be used in machine learning– an excessive amount of machine learning tactics tie in with aspects of linear algebra, such as principal component analysis, eigenvalues, and regression. This is especially true when you start working with high dimensional data, as they tend to incorporate matrices. Linear algebra also has major applications in modeling, optimization, etc., but we will be focusing on machine learning.

Know you might be asking what Machine Learning is. Machine learning is a part of the more vast field of artificial intelligence. It allows us, humans, to teach computers how to program themselves so that we don't have to write explicit instructions for certain tasks. Machine learning usually tackles tasks like predicitive modeling, clustering and finding predictive patterns. The ultimate goal is to improve the learning to the extent that it becomes automatic, so that humans no longer will have to interfere.

Motivation

Being an Applied Math major, who is currently taking MATH 110, the upper-div Linear Algebra course, I noticed that in class, a Principal Component Analysis (PCA) is defined as "a multivariate method that llows us to study and explore a set of quantitative variables measured on some objects." PCAs reduce the dimensionality of the data, while retaining its variation. So, we want compute principal components as linear combinations (Linear Algebra!) of the original variables. Before this course, I never truly recognized the real-world applications of linear algebra, but now I see its relevance, and I wanted to delve into how else it translates to the real world.

Background

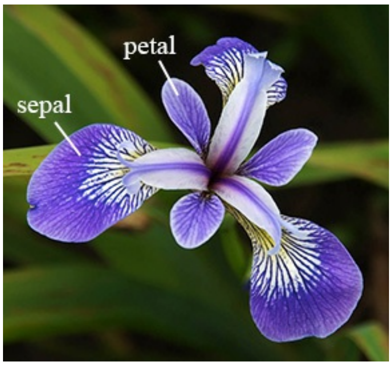
While one of the most fundamental examples of machine learning we all know is linear regression i.e. y=mx+b, we will be looking at a very simple and well know Machine Learning algorithm, called k-nearest neighbors (KNN). This is an example of a "supervised learning" algorithm. Supervised learning is the process of an algorithm learning from data, producing it's expected results, and then beig corrected by the user in order for the algorithm to improve in accuracy next run. The other type of learning is called "Unsupervised Learning," which is where algorithms are left on their own to discover and identify underlying structures in data. KNN will essentially take a data point and classify it based on the classifications of the k nearest data points by Euclidean distance

$$d = \sqrt{x^2 + y^2}$$

and takes the whatever classification is the majority of those points.

Example

We will be looking at the iris data set, which is one of the most popular data sets in data science. We will try and make a classifier that will predict classifications amongst "Iris-setosa," "Iris-versicolor," "Iris-virginica."

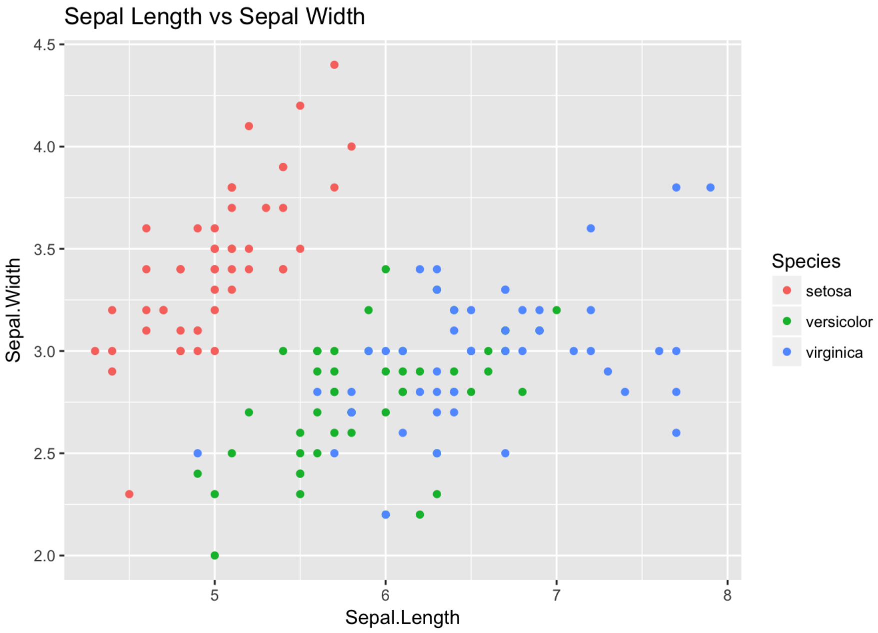


Obtaining and inspecting our data

Know we will obtain our data

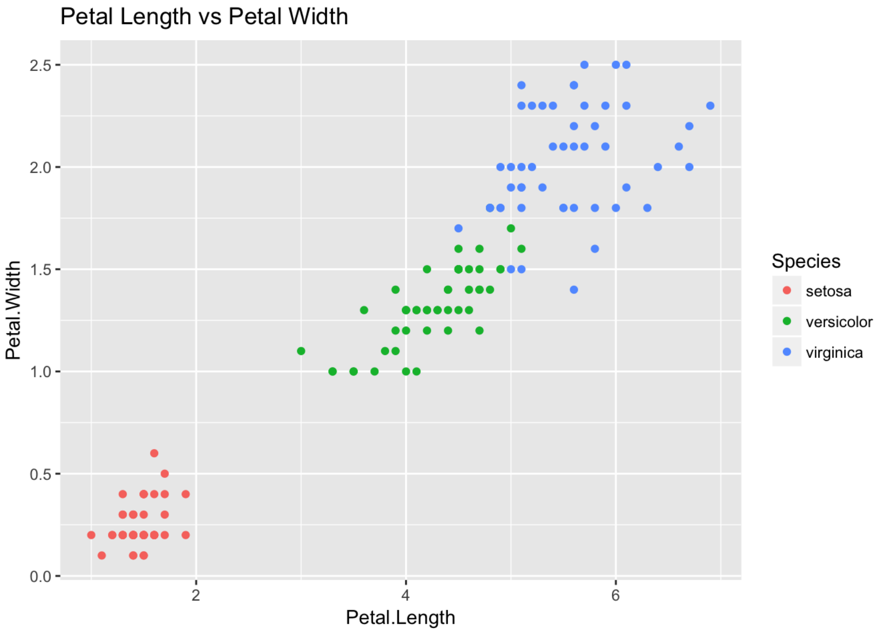
```
iris <- read.csv(url("http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"), header = FALSE)
names(iris) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species")
iris
```

```
library(ggplot2)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color=Species)) + geom_point() + ggtitle("Sepal Length vs Sepal Width"
)
```



We can see that there is a high correlation between sepal length and sepal width for the setosa flowers, while it is slightly lesser for the virginica and versicolor flowers.

```
ggplot(iris, aes(Petal.Length, Petal.Width, color=Species)) + geom_point() + ggtitle("Petal Length vs Petal Width"
)
```



This graph also indicates a positive correlation between petal length and width for all species.

Let us just get a little better overall understanding of our data.

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

```
summary(iris[c("Petal.Width", "Sepal.Width")])
```

```
##   Petal.Width     Sepal.Width
##  Min.   :0.100   Min.   :2.000
##  1st Qu.:0.300   1st Qu.:2.800
##  Median :1.300   Median :3.000
##  Mean   :1.199   Mean   :3.057
##  3rd Qu.:1.800   3rd Qu.:3.300
##  Max.   :2.500   Max.   :4.400
```

Traning our model with our data

Now we can beginning training our algorithm!

```
normalize <- function(x) {
  num <- x - min(x)
  denom <- max(x) - min(x)
  return(num/denom)
}
iris_norm <- as.data.frame(lapply(iris[1:4], normalize))
ind <- sample(2, nrow(iris), replace = TRUE, prob=c(0.67, 0.33))
iris_train <- iris[ind==1, 1:4]
iris_test <- iris[ind==2, 1:4]
```

Building our classifier

```
iris_pred <- knn(train = iris_train, test = iris_test, cl = iris[ind==1, 5], k=3)
#iris_pred
merge <- data.frame(iris_pred, iris[ind==2, 5])
names(merge) <- c("Predicted Species", "Observed Species")
merge
```

We can see here that the model makes pretty accurate predictions, except for one wrong prediction in row 29 where it predicts Iris-virginica instead of Iris-versicolor.

We can observe fundamental examples of Machine Learning and how linear algebra and cross tabulations can help prepare, visualize, and analyze data.

Discussion

In the example we just did, we normalized our data; we then randomly assigned each value to the training and test sets, and, using cross validation, were able to train our model so that it could recognize a pattern in the training set. This managed to correctly predict the classifications of 47 flowers.

Conclusion

It is clear that Machine Learning is an extremely dynamic field, and can be argued to be a study of the future. Furthermore, Linear Algebra comes up everywhere, such PCA, Singular Value Decomposition (SVD), QR Decomposition, Symmetric Matrices, Orthogonalization, Eigenvalues/vectors, etc.

References

1. https://medium.com/towards-data-science/a-definitive-guide-to-the-world-within-data-science-90300bf6330
2. http://fastml.com/math-for-machine-learning/
3. https://www.quantstart.com/articles/matrix-algebra-linear-algebra-for-deep-learning-part-2
4. http://courses.washington.edu/css490/2012.Winter/lecture_slides/02_math_essentials.pdf
5. https://www.analyticsvidhya.com/blog/2017/05/comprehensive-guide-to-linear-algebra/
6. http://www.ritchieng.com/linear-algebra-machine-learning/
7. p.com/community/tutorials/machine-learning-in-r