Unsupervised prediction

Jeffrey Leek, Assistant Professor of Biostatistics

May 18, 2016

Key ideas

- Sometimes you don't know the labels for prediction
- To build a predictor
- Create clusters
- Name clusters
- Build predictor for clusters
- In a new data set
- Predict clusters

Iris example ignoring species labels

```
data(iris); library(ggplot2); library(caret)
```

```
## Loading required package: lattice
```

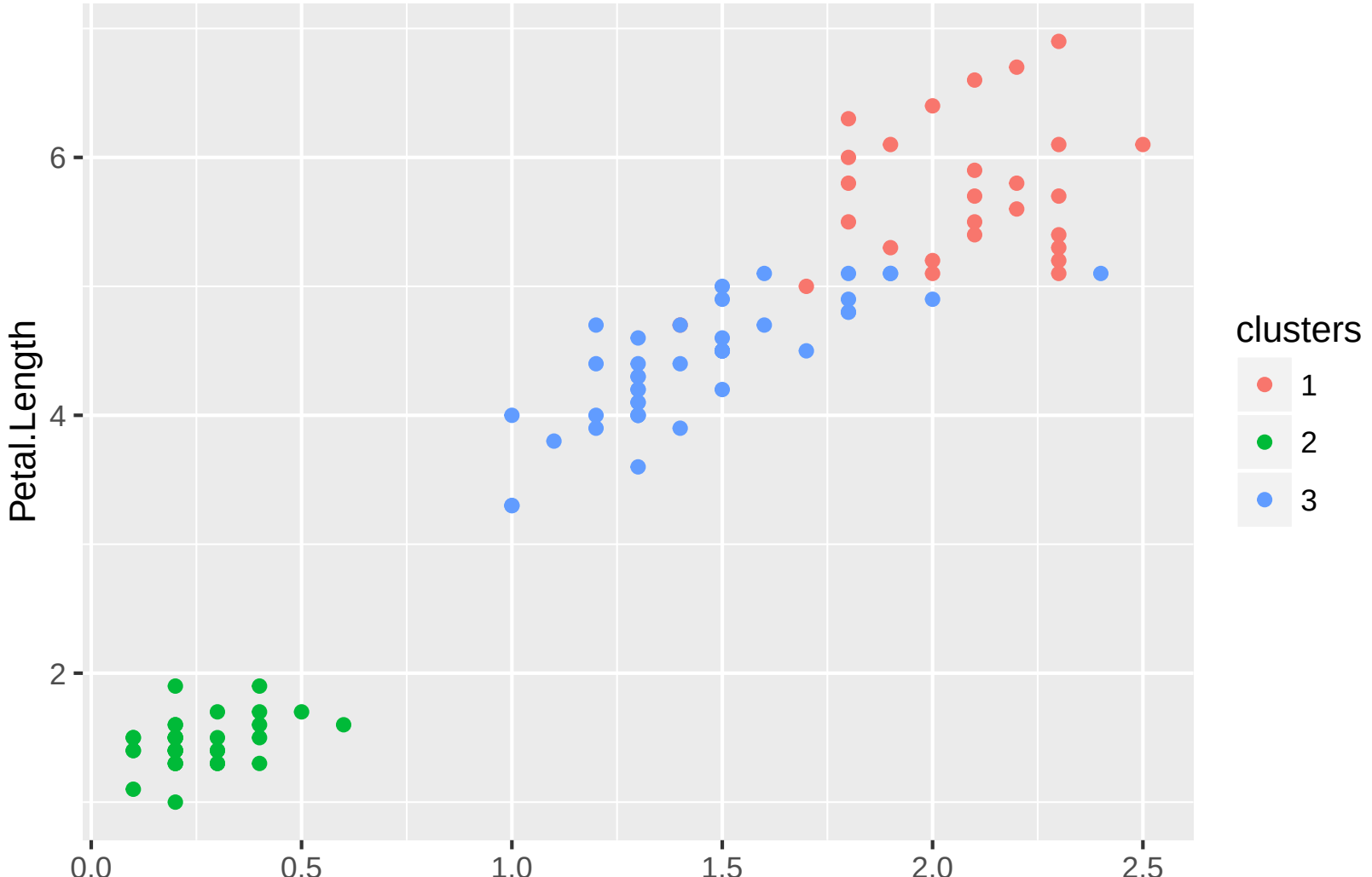
```
inTrain <- createDataPartition(y=iris$Species,
                               p=0.7, list=FALSE)
training <- iris[inTrain,]
testing <- iris[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 105   5
```

```
## [1] 45  5
```

Cluster with k-means

```
kMeans1 <- kmeans(subset(training,select=-c(Species)),cente
training$clusters <- as.factor(kMeans1$cluster)
qplot(Petal.Width,Petal.Length,colour=clusters,data=trainin
```



Compare to real labels

```
table(kMeans1$cluster,training$Species)
```

```
##
##      setosa versicolor virginica
##   1       0          2        25
##   2      35          0         0
##   3       0         33        10
```

Build predictor

```
modFit <- train(clusters ~.,data=subset(training,select=-c
```

```
## Loading required package: rpart
```

```
table(predict(modFit,training),training$Species)
```

```
##
##      setosa versicolor virginica
##   1      0          0        23
##   2     35          0         0
##   3      0         35        12
```

Apply on test

```
testClusterPred <- predict(modFit,testing)
table(testClusterPred ,testing$Species)
```

```
##
## testClusterPred setosa versicolor virginica
##               1      0          0        11
##               2     15          0         0
##               3      0         15         4
```

Notes and further reading

- The cl_predict function in the clue package provides similar functionality
- Beware over-interpretation of clusters!
- This is one basic approach to recommendation engines
- Elements of statistical learning
- Introduction to statistical learning