Machine Learning Activities

Greg Medlock

11/9/2017

Unsupervised machine learning

exercise 1

Let's work with a toy dataset to walk through an unsupervised learning algorithm.

```
sample1 = c(0,1,1,0,0,0,1,1)
sample2 = c(1,0,0,0,0,0,1,0)
sample3 = c(0,1,1,0,0,1,1,1)
sample4 = c(1,0,0,1,0,0,0,1)
sample5 = c(0,1,1,1,0,1,1,1)
sample6 = c(1,1,0,0,1,0,1,0)
sample7 = c(1,1,1,1,1,0,1,1)
sample_matrix = rbind(sample1,sample2,sample3,sample4,sample5,
                      sample6,sample7)
colnames(sample_matrix) = c('var1','var2','var3','var4','var5','var6','var7','var8')
sample_matrix
```

```
##         var1 var2 var3 var4 var5 var6 var7 var8
## sample1    0    1    1    0    0    0    1    1
## sample2    1    0    0    0    0    0    1    0
## sample3    0    1    1    0    0    1    1    1
## sample4    1    0    0    1    0    0    0    1
## sample5    0    1    1    1    0    1    1    1
## sample6    1    1    0    0    1    0    1    0
## sample7    1    1    1    1    1    0    1    1
```

Given these samples, apply the following algorithm to cluster the data in a hierarchical fashion:

1. Make each sample its own cluster.
2. Find the most similar pair (use the manhattan distance) of clusters and merge them.
3. Continue merging clusters as in step 2. When a cluster has more than one sample, use the maximum similarity between cluster members to determine cluster similarity. If there are ties (e.g. multiple pairs with the same similarity), randomly pick one. We'll see how variability here affects the final result by comparing across groups in the class.
4. When there is a single, high-level cluster that contains all samples, stop clustering.

What does your final clustering look like? Which samples are in your two largest clusters? According to the clustering, which samples are the most similar and most different from each other?

exercise 2

Let's do some clustering of real biological data. Load a subset of expression data from a human study as shown below. Three data structures will be generated, named as follows: sampleInfo, geneAnnotation, and geneExpression. We will cluster and visualize the data with heatmap.2, so load the gplots package. matrixStats will also come in handy.

```
library(GSE5859Subset)
data(GSE5859Subset)
library(gplots)
library(matrixStats)
?rowMads
```

Subset the data to include only the 25 genes with the highest standard deviation. Then, use heatmap.2 to make a heatmap showing the sampleInfo$group with color, the date as labels, the rows labelled with chromosome, and scaling the rows.

- Which genes appear most variable?
- Does the date seem to influence group membership?
- Try changing the clustering agglomeration method and the associated distance metric (hint: look at the distfun and hclustfun arguments in heatmap.2). Can you get the groups to stop clustering together? What does your result suggest about the data?
- Based on the clustering, what do you think the two groups in the experiment are?

Supervised learning

exercise 3

Let's demonstrate one of the simplest examples of supervised learning: using linear regression to predict new values not seen while constructing the model.

Load the training and test sets from https://github.com/gregmedlock/datascience_teaching (train_regression.csv and test_regression.csv, respectively).

Use the lm() function to generate a linear model that predicts y from x using the training set. Reminder: formulas look like "y ~ x", where the predictor variables are to the right of the ~.

- Determine R^2 for the fitted model. At this point, use the training data and predict() to calculate this, or access one of the attributes of the object returned by lm(). Does this model fit the data well?

Now, use predict() to generate predicted values of y using the values for x in the test set.

- Determine R^2 again, this time calculating it using the predicted y values and the actual y values for the test set. How good are the predictions? Does the model fit the test set as well as it did the training set?

exercise 4

Now, we'll build the same linear model as in exercise 3, but we will use multiple predictor variables. Load the new training and test sets from https://github.com/gregmedlock/datascience_teaching (train_3dimensions_regression.csv and test_3dimensions.csv). Fit a new linear model that uses both x, x2, and x3 as predictors.

- Determine R^2 on the test set. Is the model better?
- Investigate the contribution of each variable to the model. Which variable appears to be most important? Why? hint: use summary() with the linear model as input to view useful summary metrics.
- To test your answer to the previous question, construct a new linear model to predict y from only the most important variable. Plot the actual value versus predicted value for the full model and make the same plot for the model that includes only the most important variables. Are the plots qualitatively different?