Week 8

SUBFORUMS

All

Assignment: K-Means Clustering and PCA

Week 8

ex7 tutorial for findClosestCentroids()

Tom Mosher Mentor Week 8 · 3 years ago · Edited

This tutorial gives a method for findClosestCentroids() by iterating through the centroids. This runs considerably faster than looping through the training examples.

- Create a "distance" matrix of size (m x K) and initialize it to all zeros. 'm' is the number of training examples, K is the number of centroids.
- Use a for-loop over the 1:K centroids.
- Inside this loop, create a column vector of the distance from each training example to that centroid, and store it as a column of the distance matrix. One method is to use the bsxfun) function and the sum() function to calculate the sum of the squares of the differences between each row in the X matrix and a centroid.
- When the for-loop ends, you'll have a matrix of centroid distances.
- Then return idx as the vector of the indexes of the locations with the minimum distance. The result is a vector of size (m x 1) with the indexes of the closest centroids.

==========

Additional implementation tips on how to use bsxfun():

bsxfun() means "broadcast function". It applies whatever function you specify (such as @minus) between the 1st argument and the second argument (typically a row vector).

Here's an example you can enter in your console:

```
Q = magic(3)
D = bsxfun(@minus, Q, [1 2 3])
```

It returns a value that is the same size as the first argument.

As implemented for this function, each row in the matrix returned by bsxfun() is the differences between that row of X and each of the features of one of the centroids.

Then, for each row, you compute the sum of the squares of those differences. Here is code that does it.

```
sum(D.^2,2)
```

Note that if you leave off the ',2' part, then sum() will sum over the rows, and you'll instead get a (1xn) vector. Using ',2' tells sum() to work over the columns, so you'll get a (m x 1) result.

We repeat this process for each of the centroids. Once we have the squared distance between each example and each centroid, we then return the index where the min() value was found for each example.

keywords: ex7 tutorial findclosestcentroids