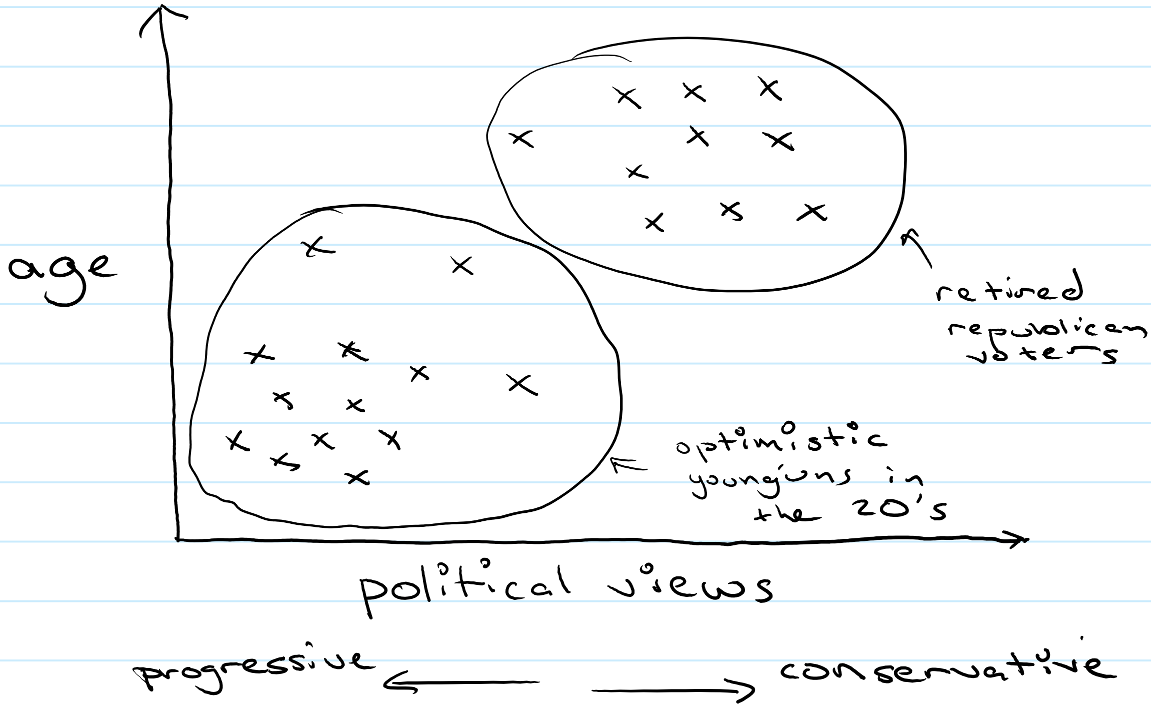
K-means

Thursday, December 10, 2015 8:23 PM

Let's start with defining the problem that k-means attempts to solve

In unsupervised learning, when we don't have labels for our data, sometimes we want to automatically detect structure in the data



In the contrived example above, we are not given which of the two groups a person belongs to, but we're hoping an algorithm can automatically cluster the data and find these groups.

Coming back to k-means, it is an algorithm used for clustering.

K-means

```
1. Select the hyperparameter k, the
   number of clusters

2. Initialize K centroids by randomly
   picking training examples

3. Repeat until convergence {

       a) Compute distence between
          each training example and
          all k centroids

       b) Assign each training point to
          the closest centroid

       c) For each centroid, average
          the training examples
          marked to that centroid

       d) Update the centroids with
          the average computed

   }
```

The one thing to note is that k-means can get stuck in a local minima.

To combat this, attempting multiple different random initializations helps

The cost function k-means is minimizing is:

$$J(c^{(1)}, c^{(2)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$
$$= \frac{1}{m} \sum_{i=1}^{m} || x^{(i)} - \mu_{c^{(i)}} ||^2$$

where $c^{(i)}$ is what centroid example $x^{(i)}$ is assigned to and μ_j is centroid j

To pick which clustering to use, use the one that minimizes J