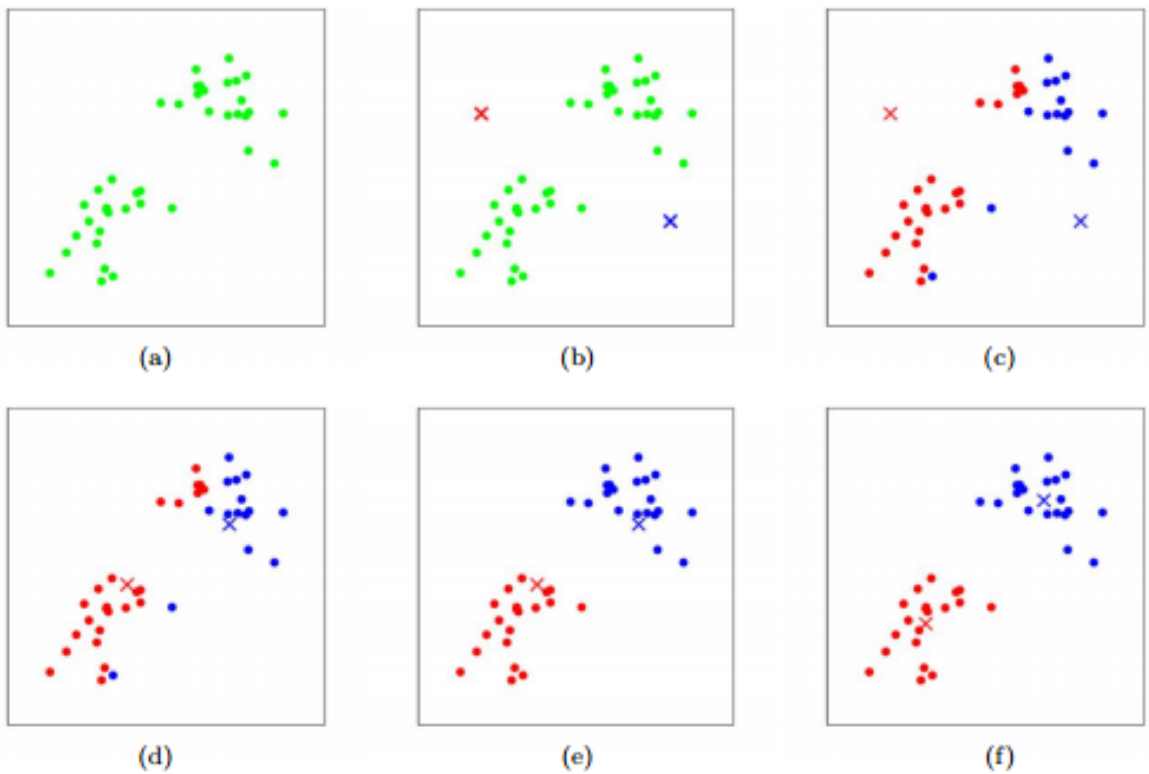
A2 Overview

Chris Kauffman

Last Updated:
Thu Feb 24 09:39:08 AM CST 2022

K-Means Clustering

- A standard ML / Data Mining / Stats problem
- Input: data + #of clusters desired
- Output: assignment of each data to a cluster + cluster centers
- Algorithm: Iterates between
  1. Calculate cluster centers
  2. Calculate cluster assignments



Source: K-Means by Chris Piech. Based on a handout by Andrew Ng.

Determine Cluster Centers

```
1     # DETERMINE NEW CLUSTER CENTERS
2     for c in range(nclust):         # reset cluster ndatas to 0
3       clust_count[c] = 0
4     for c in range(nclust):         # reset cluster centers to 0.0
5       for d in range(dim):
6         clust_cents[c][d] = 0.0
7
8     for i in range(ndata):          # sum up data in each cluster
9       c = data_clust[i]
10      clust_count[c] += 1
11      for d in range(dim):
12        clust_cents[c][d] += data[i][d]
13
14    for c in range(nclust):         # divide by count in clust for center
15      for d in range(dim):
16        clust_cents[c][d] = clust_cents[c][d] / clust_count[c]
```

Determine Cluster Assignments

```
1     # DETERMINE NEW CLUSTER ASSIGNMENTS FOR EACH DATA
2     nchanges = 0
3     for i in range(ndata):          # iterate over all data
4       best_clust = None
5       best_distsq = float("inf")
6       for c in range(nclust):       # compare to each center, assign to closest
7         distsq = 0.0
8         for d in range(dim):        # squared dist in each data dimension
9           diff = data[i][d] - clust_cents[c][d]
10          distsq += diff*diff
11        if distsq < best_distsq:    # closer than current best?
12          best_clust = c
13          best_distsq = distsq
14      if best_clust != data_assign[i]: # assign to a different cluster?
15        nchanges += 1                  # cluster assignment changed
16        data_assign[i] = best_clust
```

Overall

```
# ASSIGN RANDOM CLUSTER TO EACH DATA
...

maxiter = 100        # bounds the iterations
curiter = 1          # current iteration
nchanges = ndata     # count changes in assignment each iter

while nchanges > 0 and curiter <= maxiter: # loop until convergence
  # DETERMINE NEW CLUSTER CENTERS
  ...

  # DETERMINE NEW CLUSTER ASSIGNMENTS FOR EACH DATA
  ...
```

Parallel Versions

- Algorithm deals with Data and Clusters, each a matrixy thing
- How would you divide up this data in a distributed parallel version?
- Would data redistribution be required in your scheme?
- What information needs to be exchanged at each iteration?