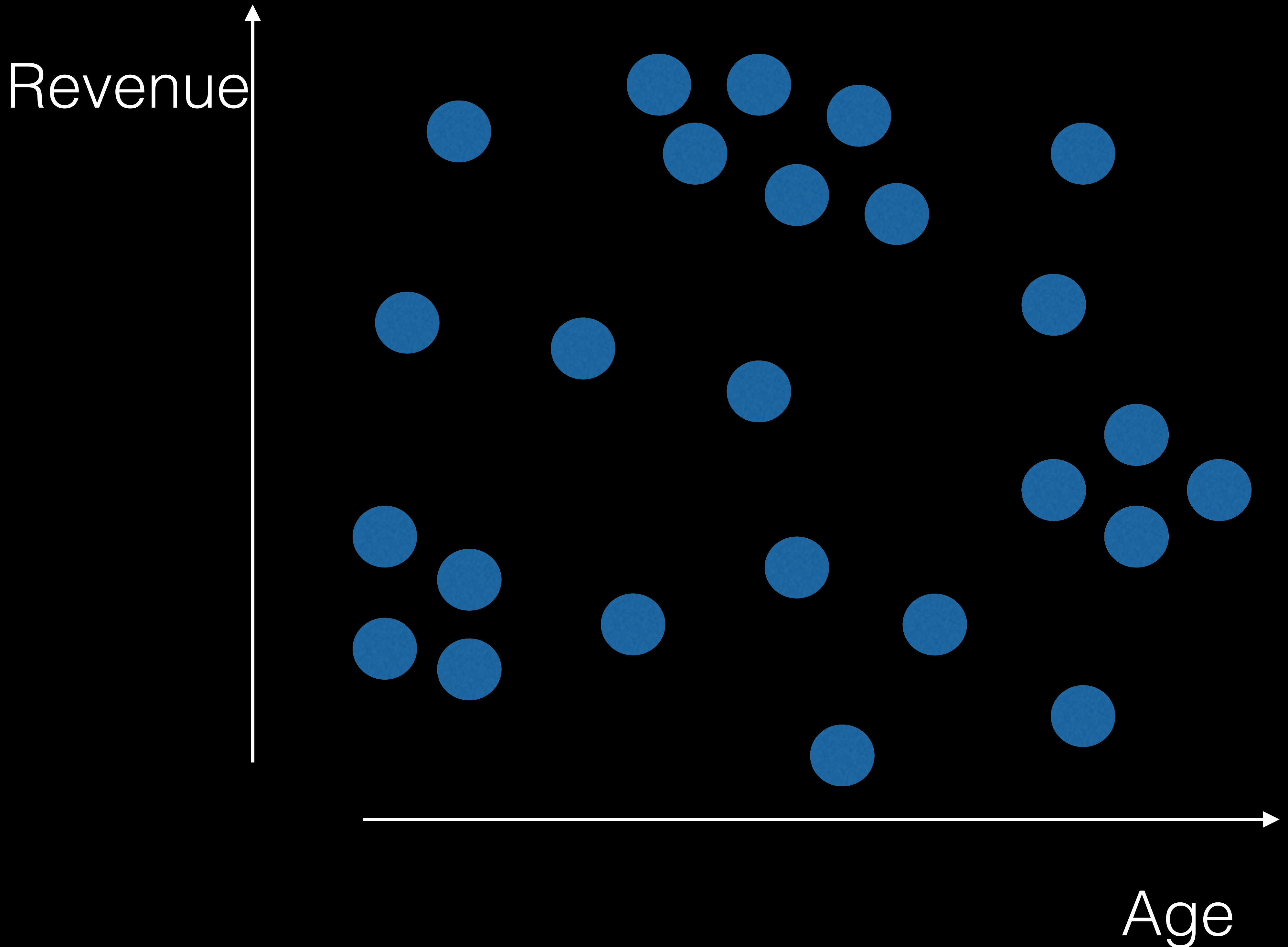
Mahout 102

Clustering

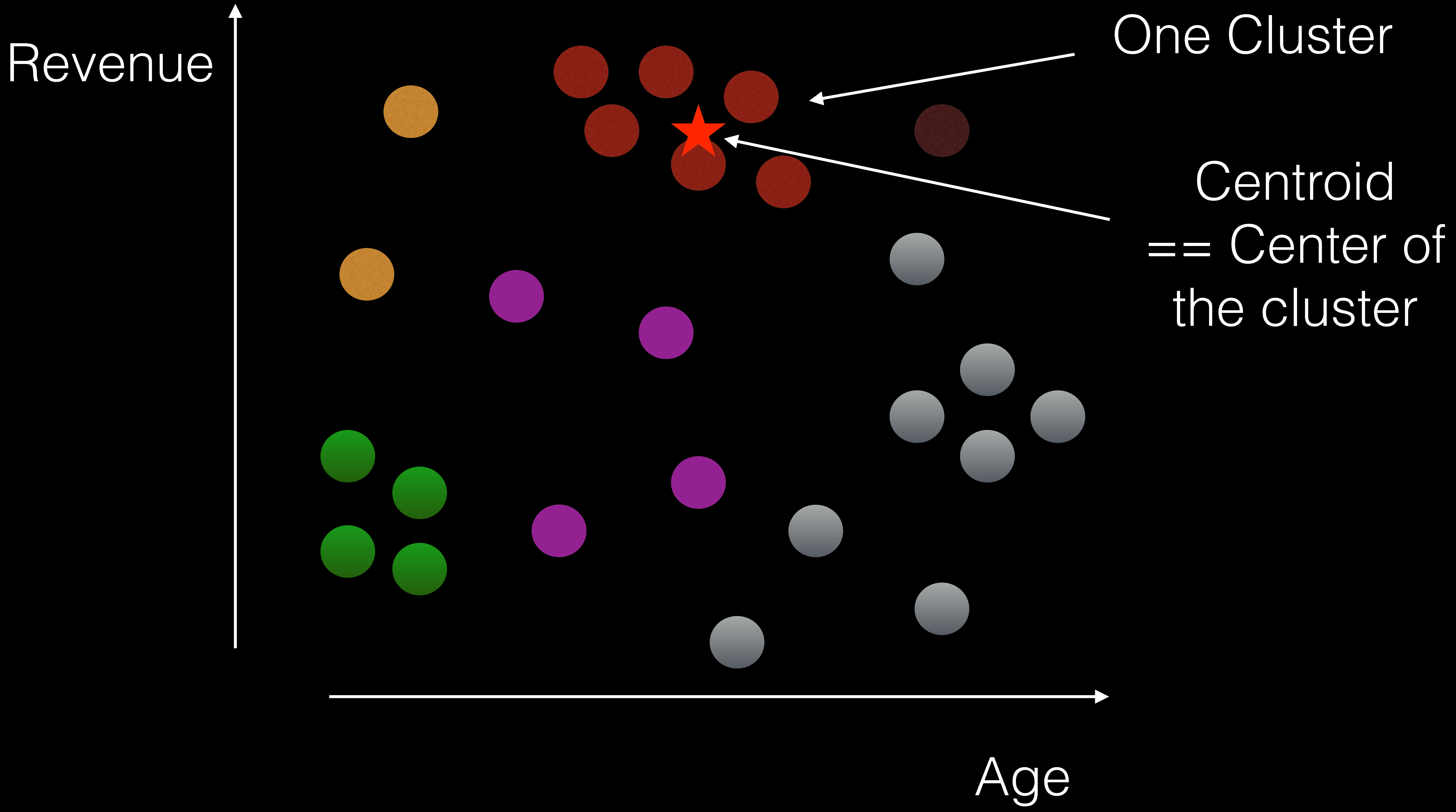
Goal for Today

- Quick Introduction To Clustering
- How does it work in Practice
- How does it work in Mahout
- Overview of Mahout Algorithms

Clustering



Clustering



clustering applications

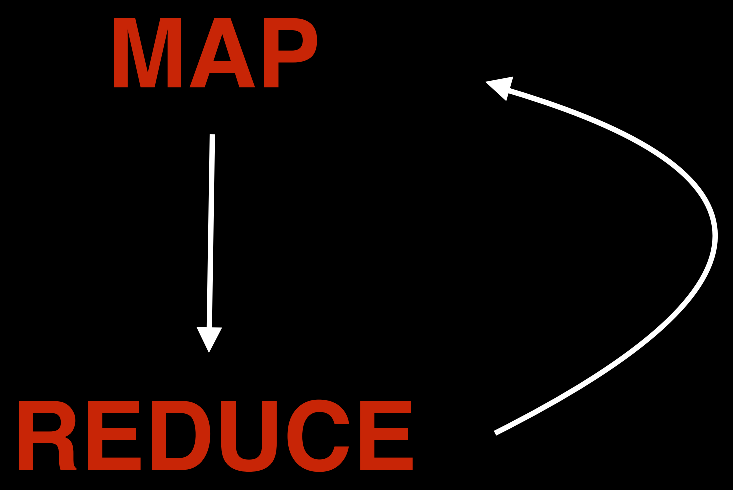
- Fraud: Detect Outliers
- CRM : Mine for customer segments
- Image Processing : Similar Images
- Search : Similar documents
- Search : Allocate Topics

K-Means

Guess an initial placement for centroids

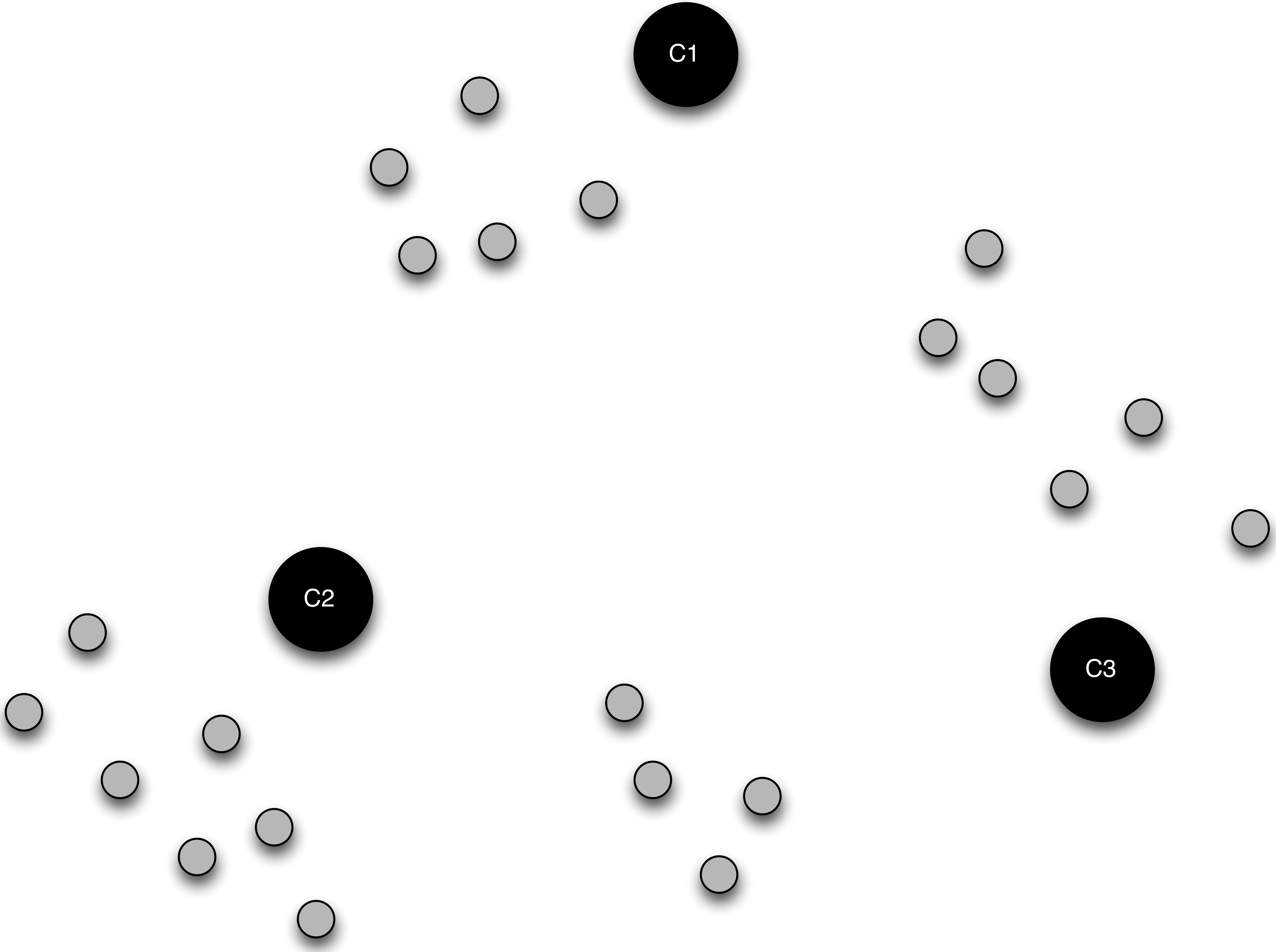
Assign each point to closest Center

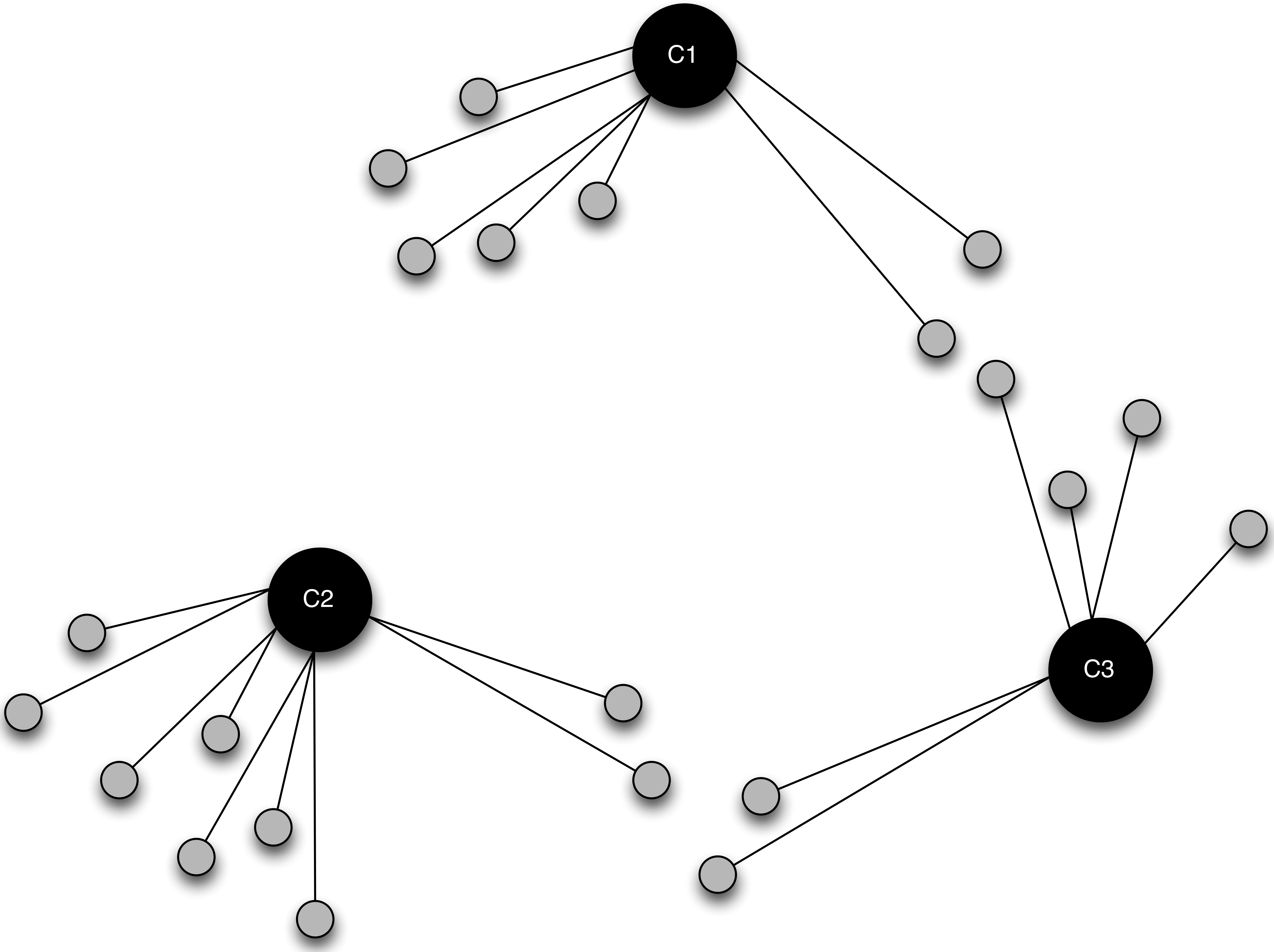
Reposition Center

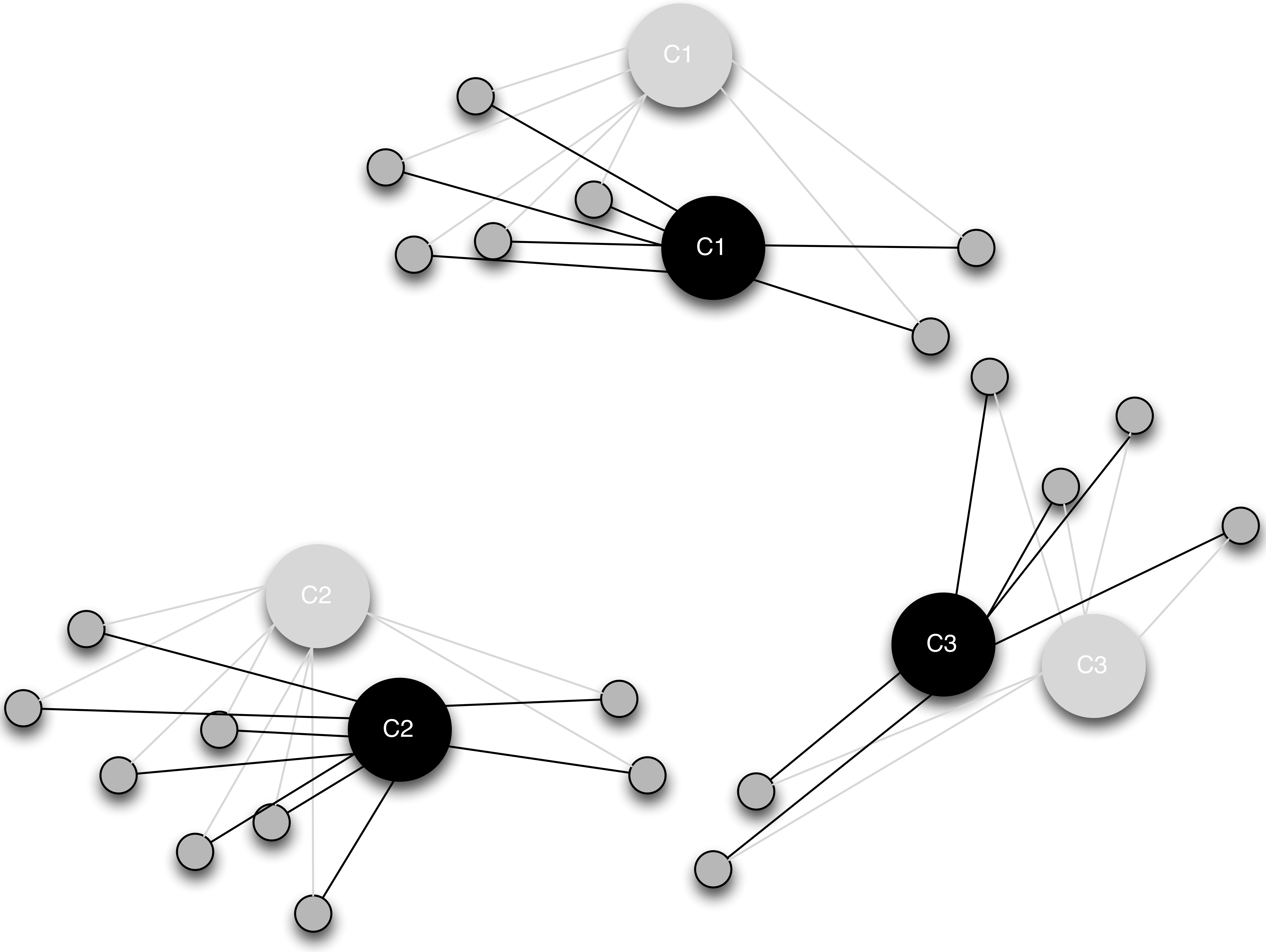


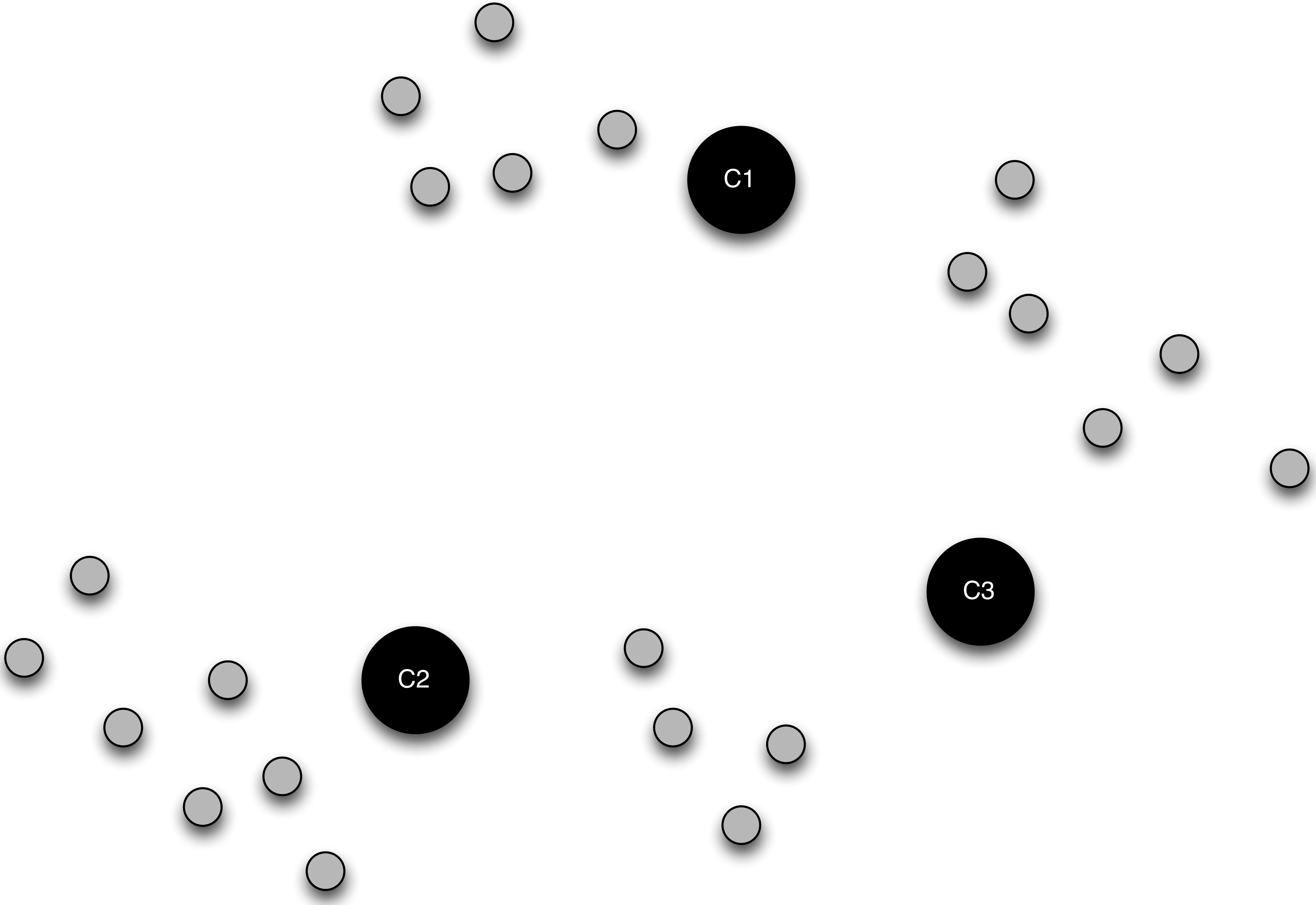
MAP

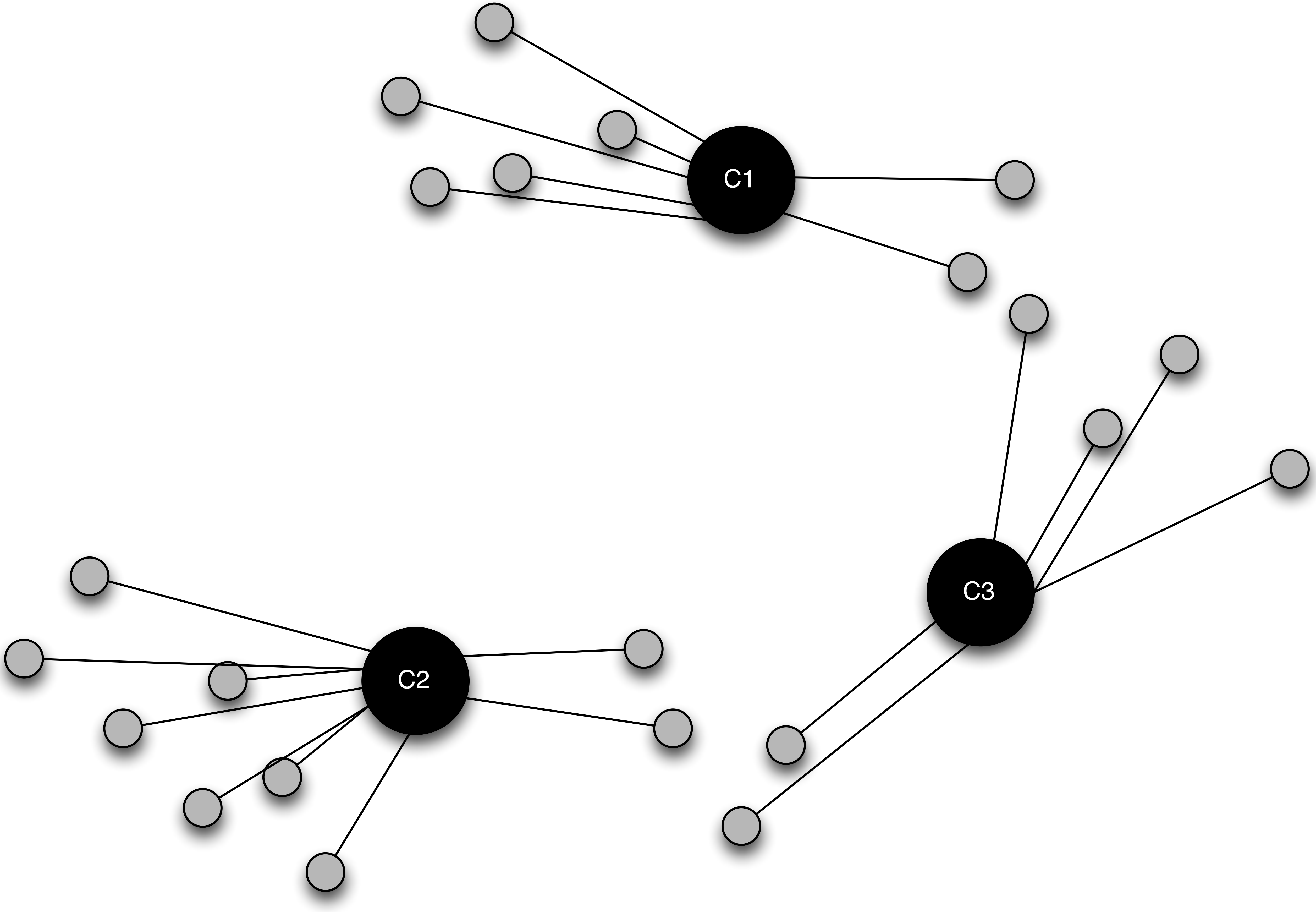
REDUCE

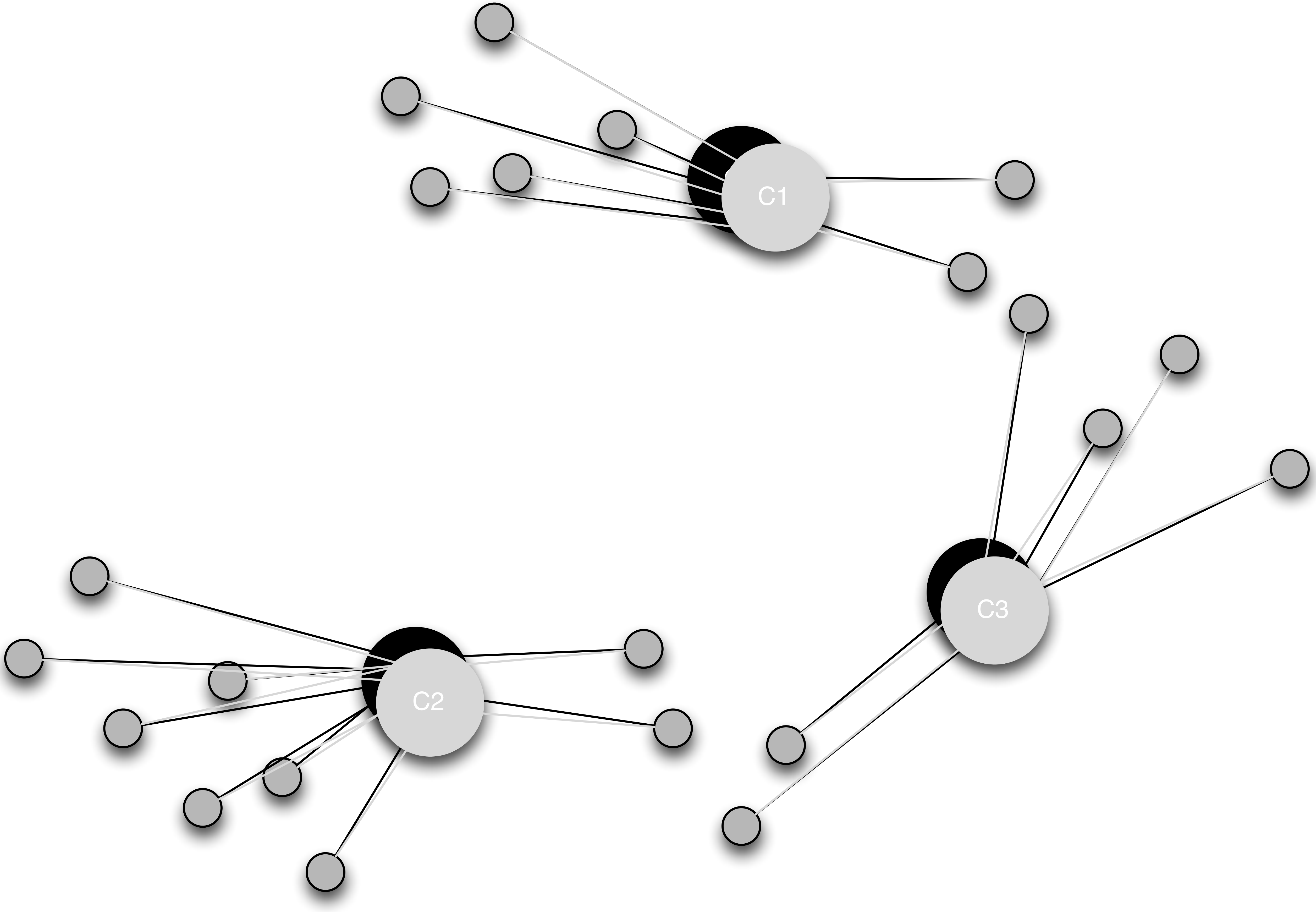

C1
C2
C3

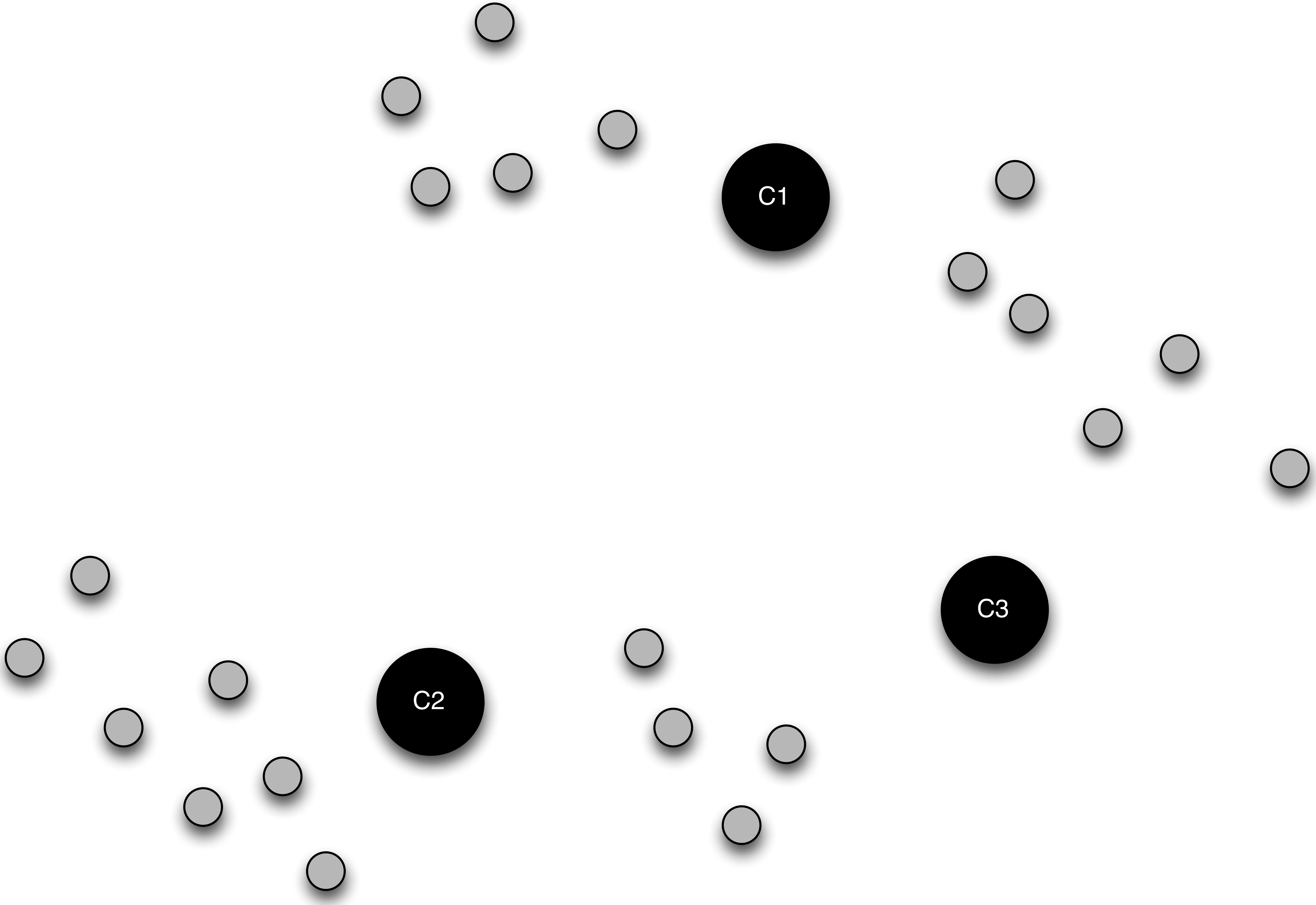

C1
C2
C3

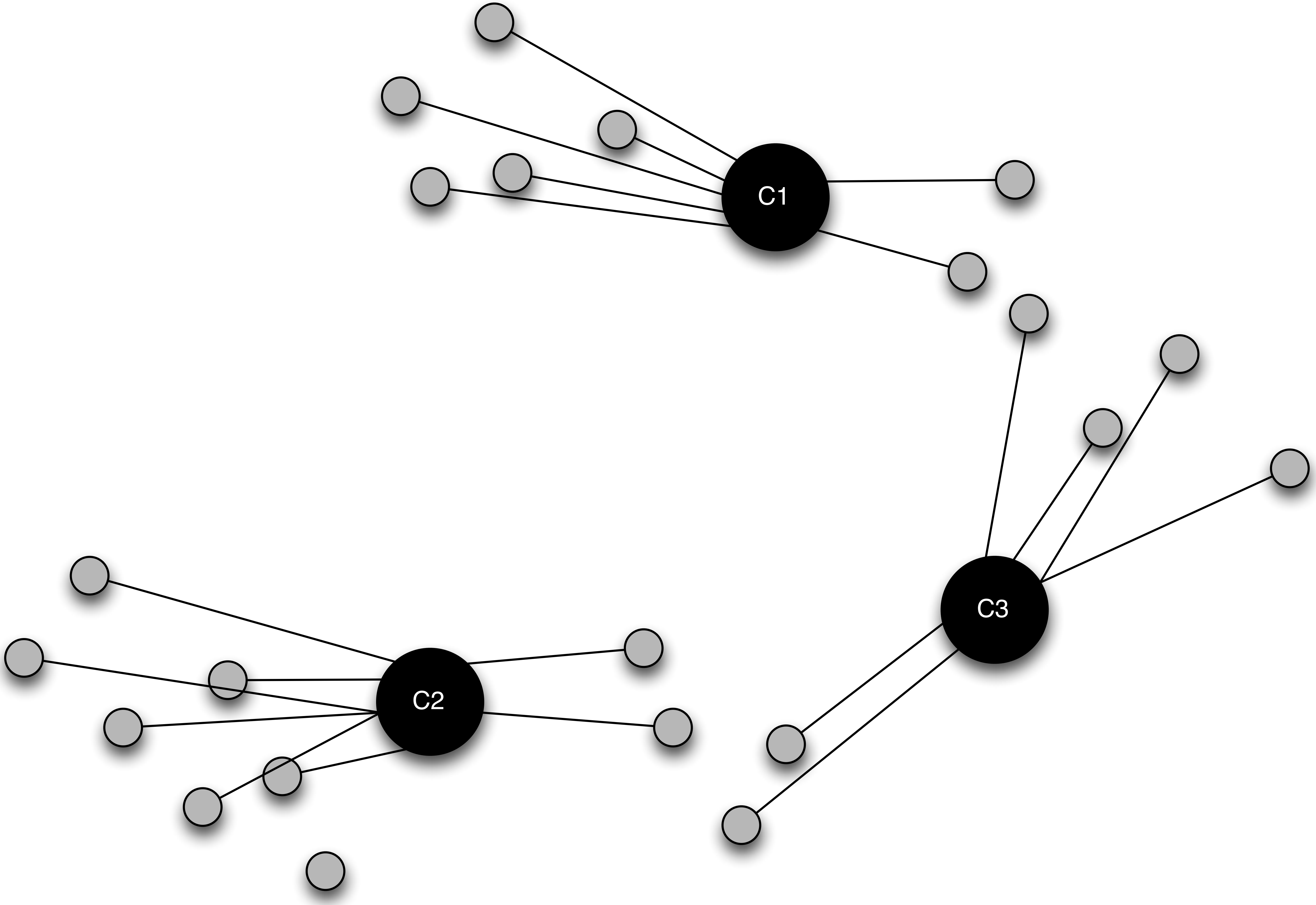

C1
C1
C2
C2
C3
C3

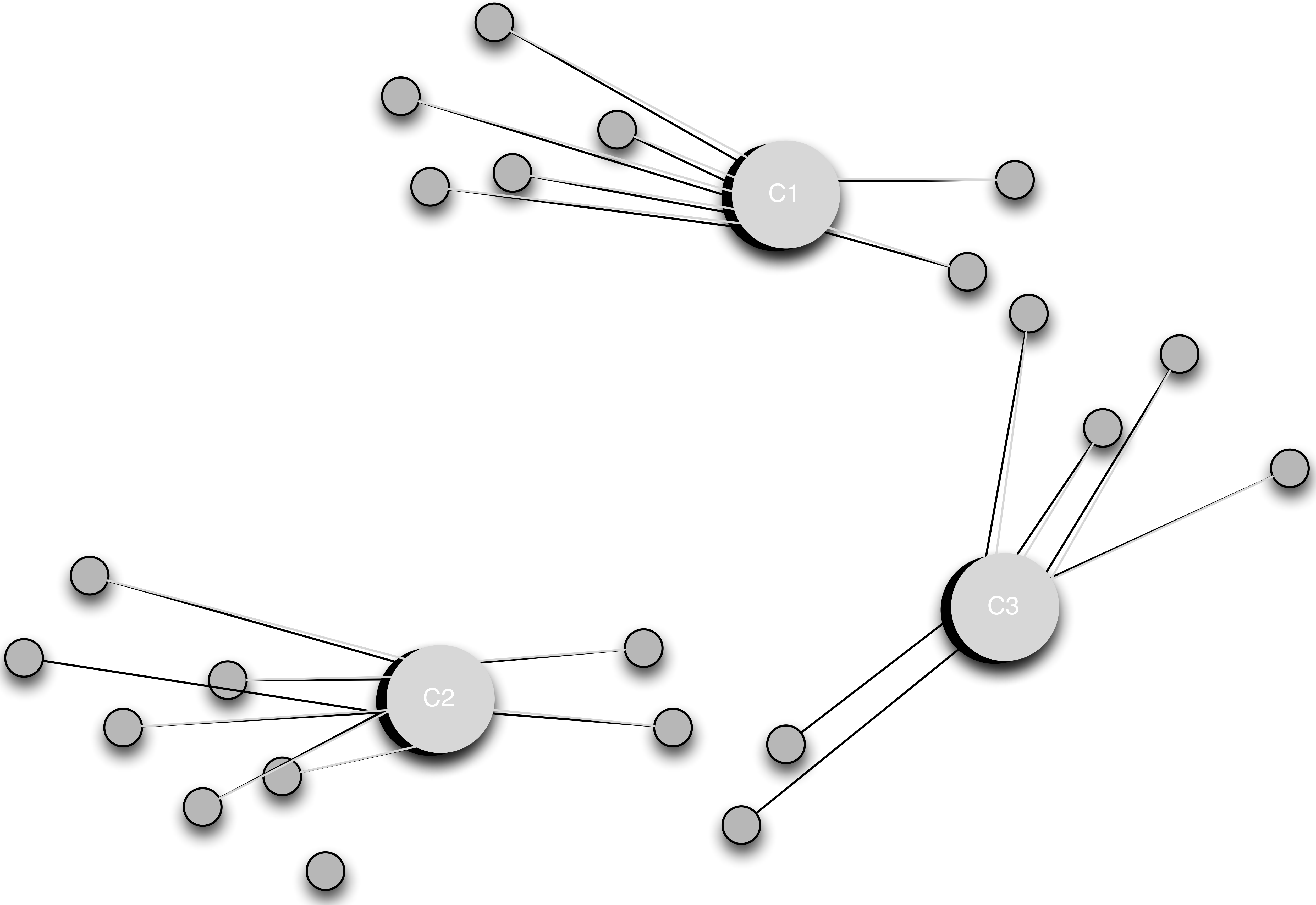

C1
C3
C2


C1
C2
C3


C1
C2
C3


C1
C3
C2


C1
C2
C3


C1
C2
C3

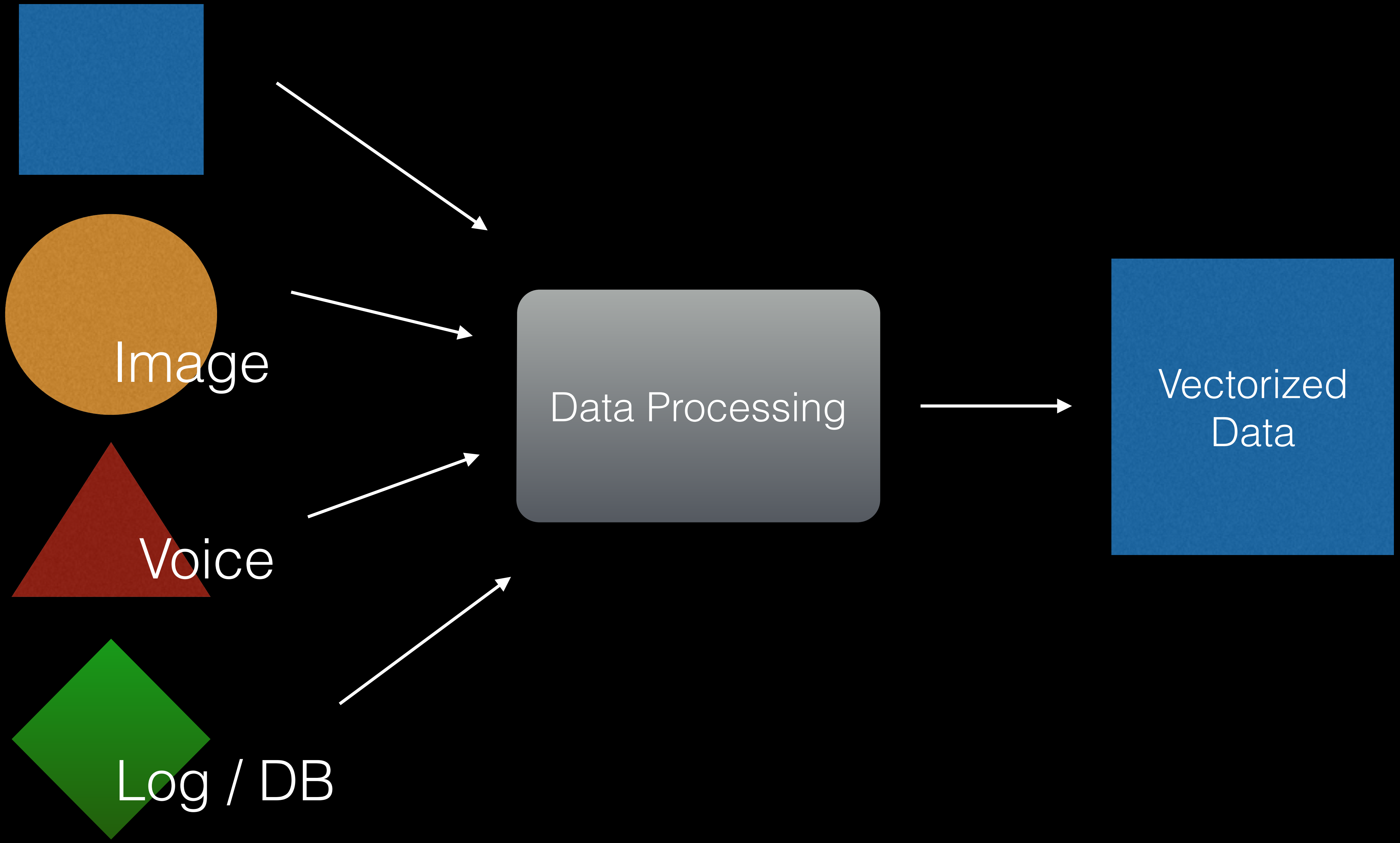
clustering challenges

- Curse of Dimensionality
- Choice of distance / number of parameters
- Performance
- Choice # of clusters

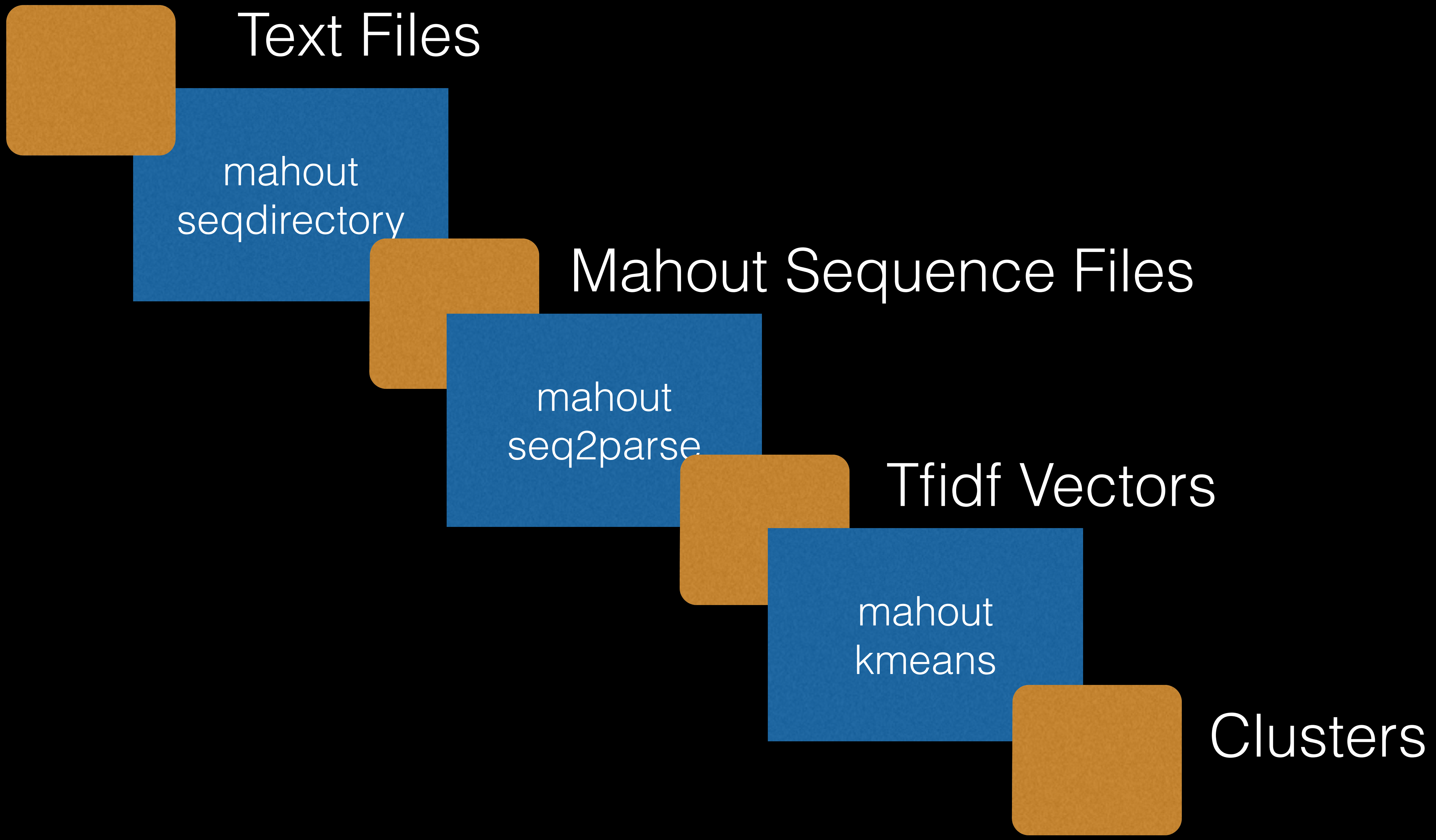
Mahout Clustering Challenges

- No Integrated Feature Engineering Stack: Get ready to write data processing in Java
- Hadoop SequenceFile required as an input
- Iterations as Map/Reduce read and write to disks: Relatively slow compared to in-memory processing

Data Processing



Mahout K-Means on Text Workflow



Text Files

mahout seqdirectory

Mahout Sequence Files

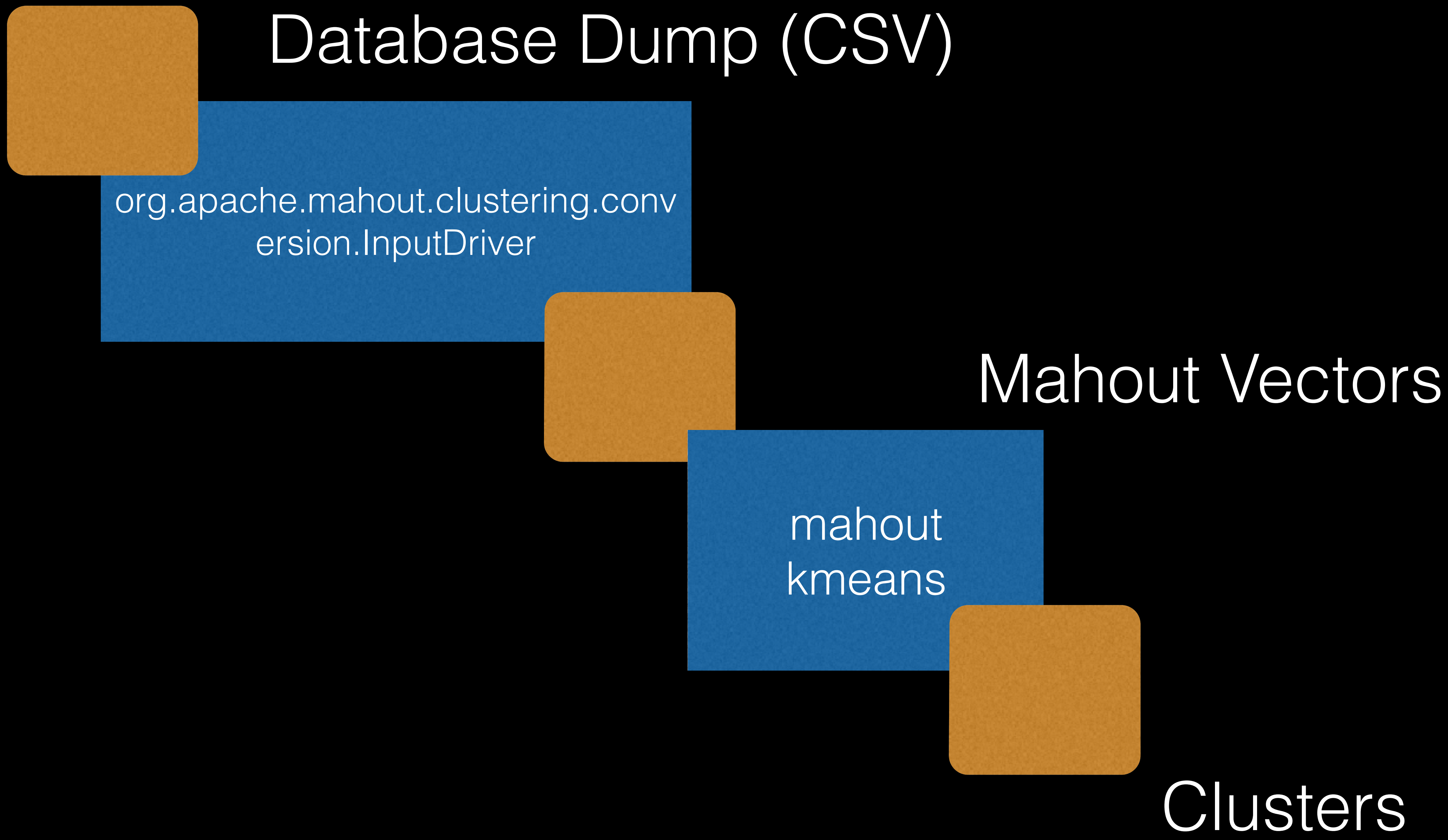
mahout seq2parse

Tfidf Vectors

mahout kmeans

Clusters

Mahout K-Means on Database Extract Worflow



Database Dump (CSV)

org.apache.mahout.clustering.conversion.InputDriver

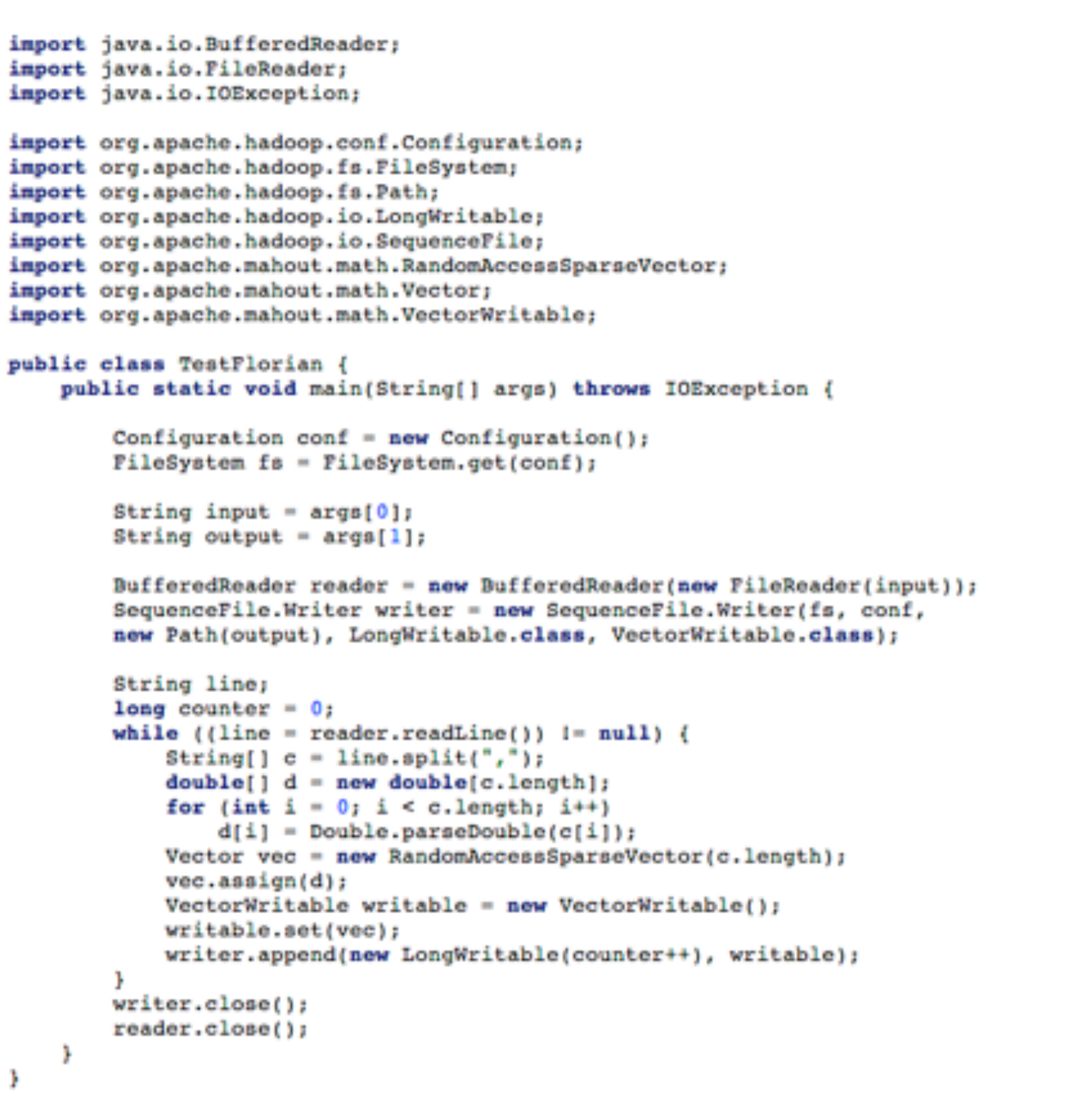
Mahout Vectors

mahout kmeans

Clusters

Convert a CSV File to Mahout Vector

- Real Code would have
  - Converting Categorical variables to dimensions
  - Variable Rescaling
  - Dropping IDs (name, forname …)



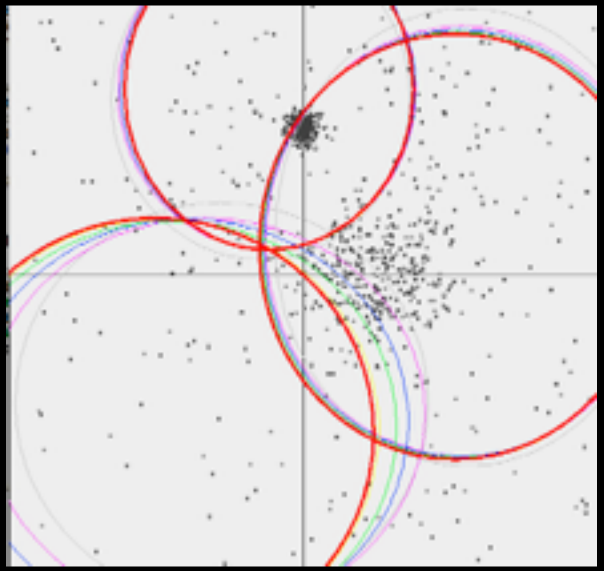
```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.SequenceFile;
import org.apache.mahout.math.RandomAccessSparseVector;
import org.apache.mahout.math.Vector;
import org.apache.mahout.math.VectorWritable;

public class TestFlorian {
    public static void main(String[] args) throws IOException {

        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(conf);

        String input = args[0];
        String output = args[1];

        BufferedReader reader = new BufferedReader(new FileReader(input));
        SequenceFile.Writer writer = new SequenceFile.Writer(fs, conf,
        new Path(output), LongWritable.class, VectorWritable.class);

        String line;
        long counter = 0;
        while ((line = reader.readLine()) != null) {
            String[] c = line.split(",");
            double[] d = new double[c.length];
            for (int i = 0; i < c.length; i++)
                d[i] = Double.parseDouble(c[i]);
            Vector vec = new RandomAccessSparseVector(c.length);
            vec.assign(d);
            VectorWritable writable = new VectorWritable();
            writable.set(vec);
            writer.append(new LongWritable(counter++), writable);
        }
        writer.close();
        reader.close();
    }
}
```

Mahout Algorithms

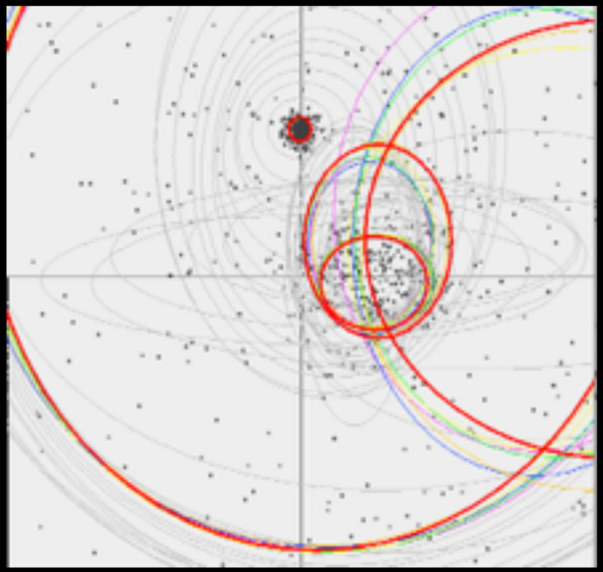
	Parameters	Implicit Assumption	Ouput
K-Means	K (number of clusters) Convergence	Circles	Point -> ClusterId
Fuzzy K-Means	K (number of clusters) Convergence	Circles	Point -> ClusterId * , Probability
Expectation Maximization	K (Number of clusterS) Convergence	Gaussian distribution	Point -> ClusterId*, Probability
Mean-Shift Clustering	Distance boundaries, Convergence	Gradient like distribution	Point -> Cluster ID
Top Down Clustering	Two Clustering Algorithns	Hierarchy	Point -> Large ClusterId, Small ClusterId
Dirichlet Process	Model Distribution	Points are a mixture of distribution	Point -> ClusterId, Probability
Spectral Clustering	-	-	Point -> ClusterId
MinHash Clustering	Number of hash / keys Hash Type	High Dimension	Point -> Hash*

Comparing Clustering

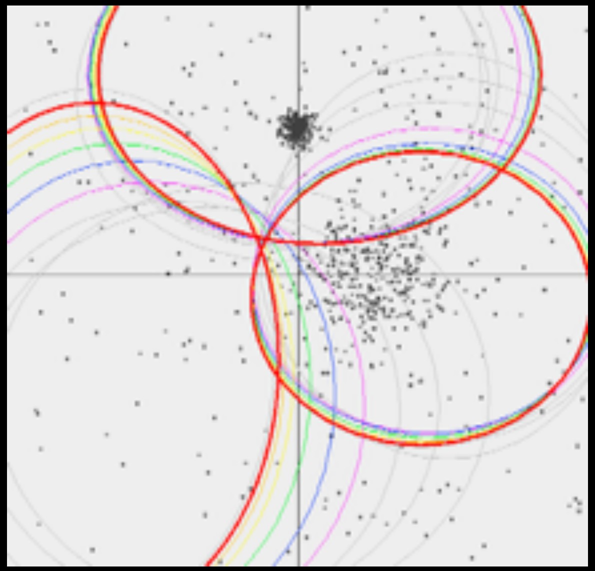
KMeans



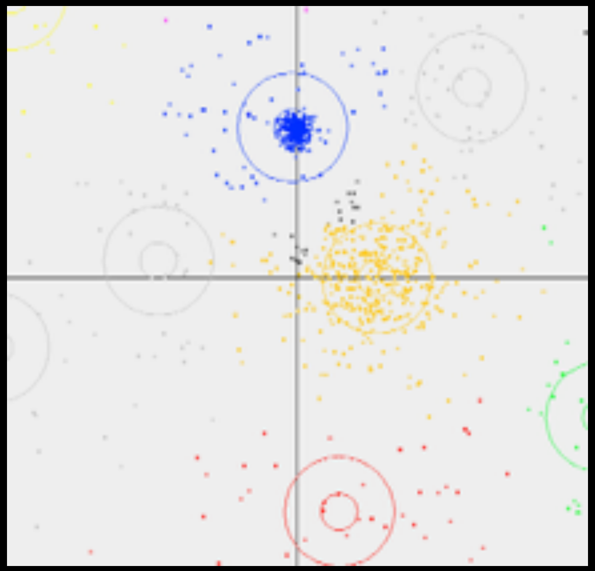
Dirichlet



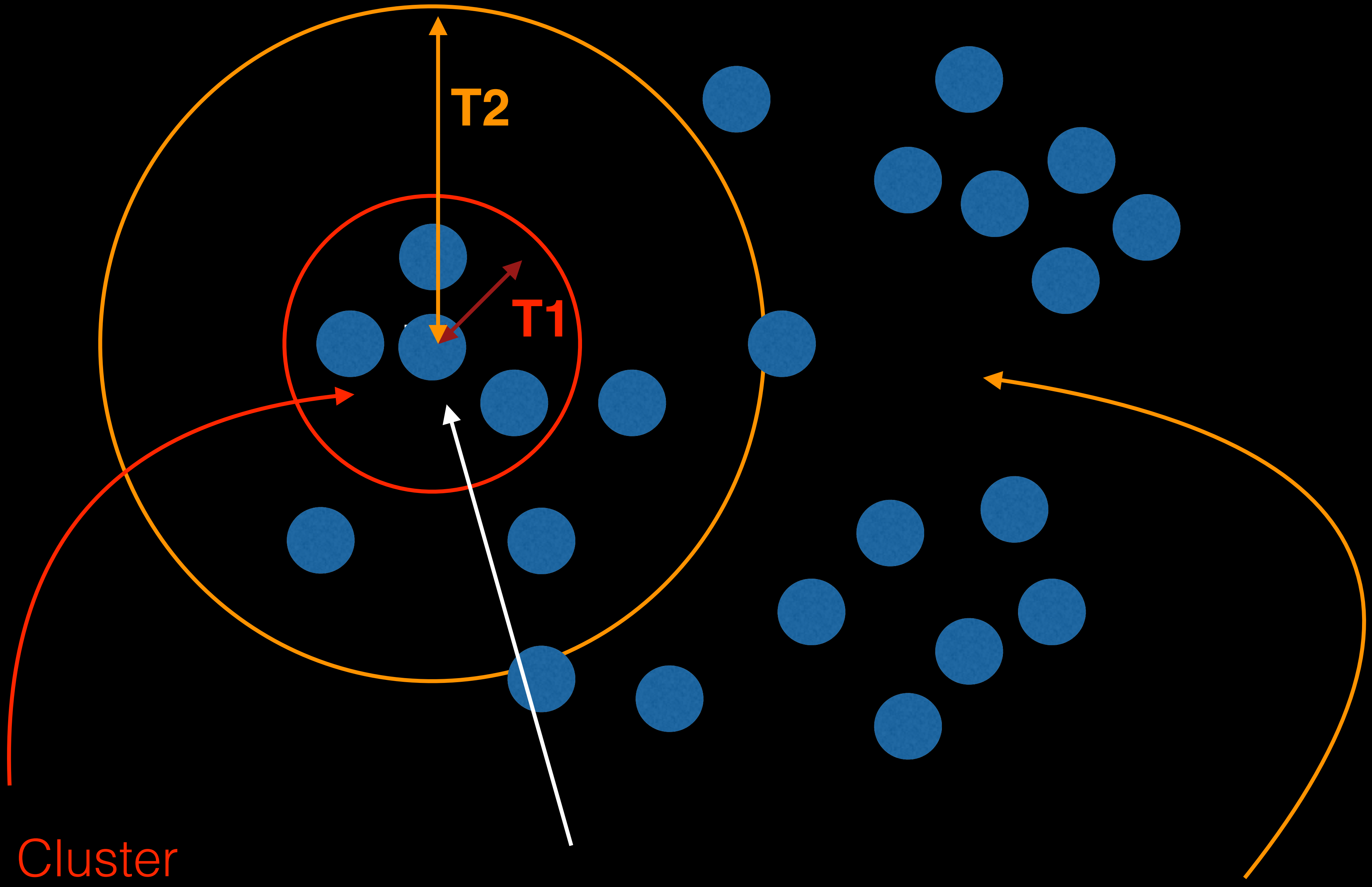
Fuzzy KMeans



MeanShift



Canopy Optimization



Surely in Cluster

Pick a random point

Surely not in cluster