CS4054
Bioinformatics

Spring 2025

Rushda Muneer

Gene Expression

- Gene expression is the process our cells use to convert the instructions in our DNA into a functional product, such as a protein.
- Our DNA stores the information our cells need to function.
- It is organized into small sections, called genes, which contain instructions for making a specific product, usually a protein.
- Gene expression provides the information about transcripts generated by particular genes in an organism at various time checkpoints

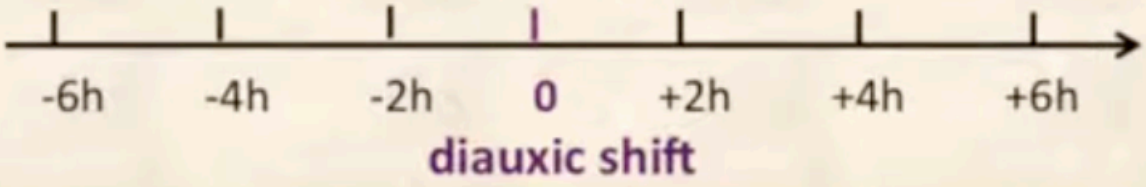
Example

- Yeast is commonly used for fermentation.
- It observes the phenomenon of diauxic shift to convert acetaldehyde into ethanol and vice versa.
- Diauxic shift is the interval where the metabolic process reverses.
- In an experiment, expression levels (the relative measurement of transcripts) was observed before and after the diauxic shift

Analyzing Gene Expression Through Expression Matrices

- **Gene Expression Matrix (E):**
  - Rows = Genes (n)
  - Columns = Time Checkpoints (m)
  - $E_{i,j}$ = Expression level of gene *i* at checkpoint *j*
  - Expression vector = Row corresponding to a single gene
- **Patterns Across the Diauxic Shift:**
  - Constant expression
  - Spike before shift, drop after
  - Sudden rise post-shift
  - Other unique patterns

Measure expression of various yeast genes at 7 checkpoints:



YLR258W	1.1	1.4	1.4	3.7	4.0	10.0	5.9
YPL012W	1.1	0.8	0.9	0.4	0.3	0.1	0.1
YPR055W	1.1	1.1	1.1	1.1	1.1	1.1	1.1

expression level
of gene *i* at checkpoint *j*

Visualizing Yeast Gene Expression Vectors

- **Genes & Expression Vectors:**
  - YLR258W**:** (1.07, 1.35, 1.37, 3.70, 4.00, 10.00, 5.88)
  - YPL012W**:** (1.06, 0.83, 0.90, 0.44, 0.33, 0.13, 0.12)
  - YPR055W**:** (1.11, 1.11, 1.12, 1.06, 1.05, 1.06, 1.05)

Interpretation:

Above 1.0 → Increased expression

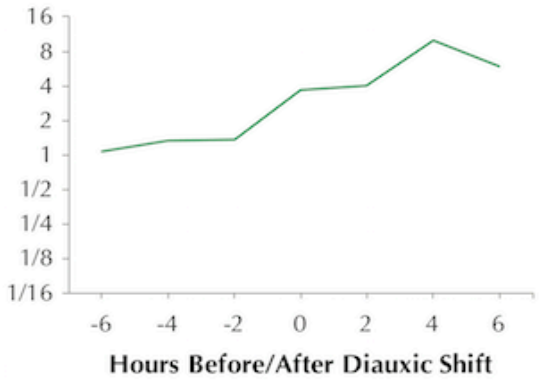
Below 1.0 → Decreased expression

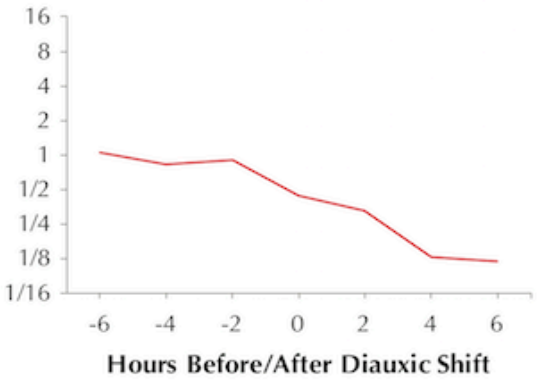
Observe differences in behavior:

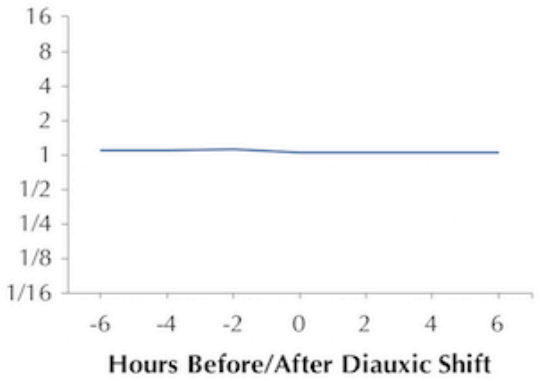
YLR258W**:** Strong increase

YPL012W: Strong decrease

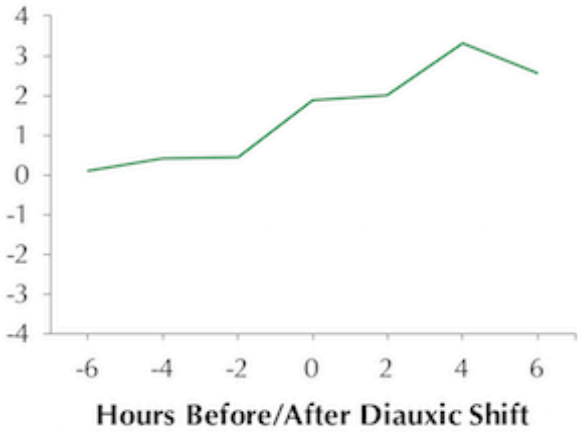
YPR055W: Stable expression

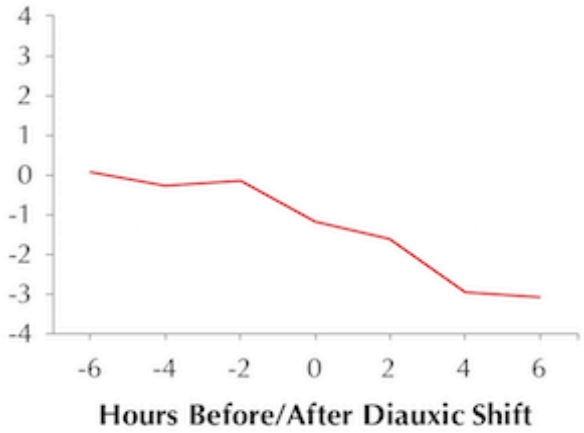


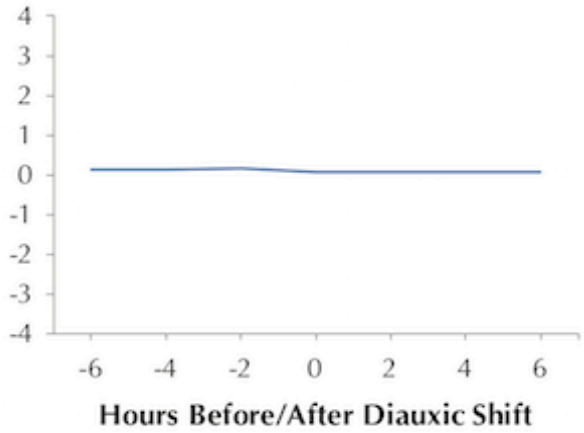




Switching to Logarithmic expression levels







The expression vectors of the three genes from the previous step with expression levels substituted by their base-2 logarithms:
(0.11, 0.43, 0.45, 1.89, 2.00, 3.32, 2.56),
(0.09, -0.28, -0.15, -1.18, -1.59, -2.96, -3.08),
and (0.15, 0.15, 0.17, 0.09, 0.07, 0.09, 0.07).

Introduction to Clustering

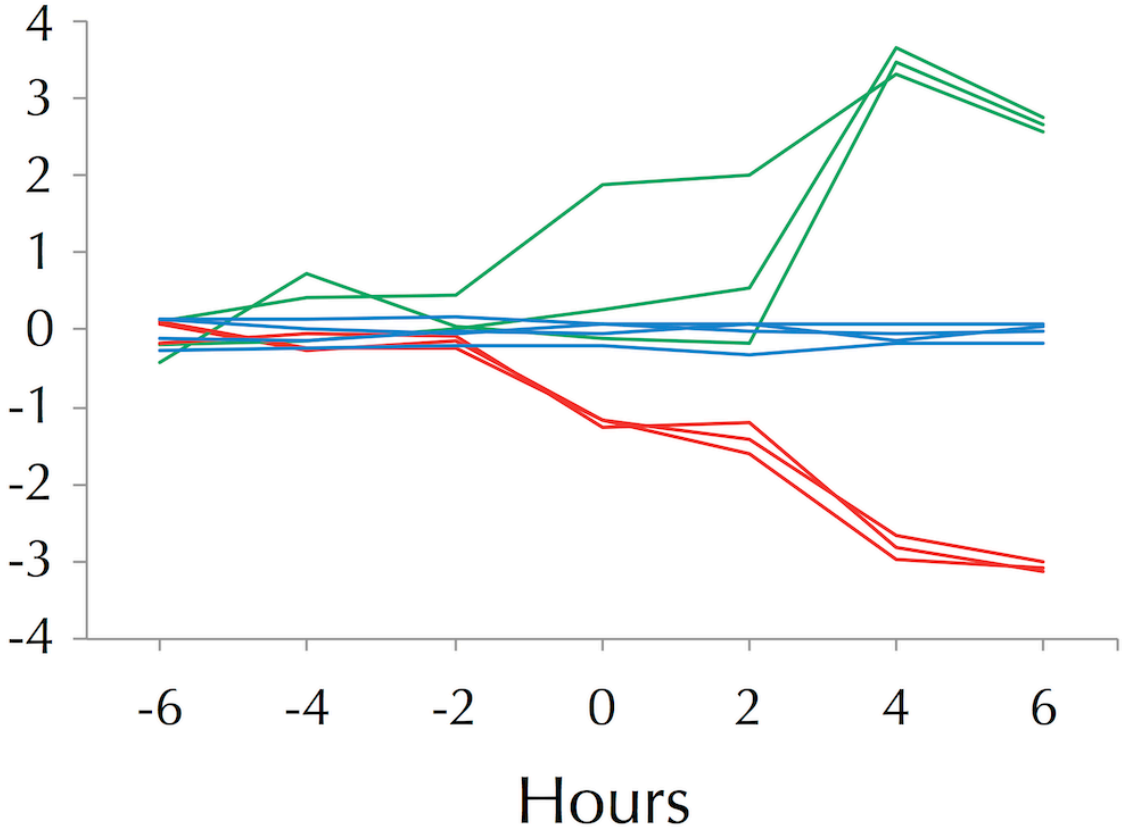
- An expression matrix of ten yeast genes after taking logarithms

Gene	Expression Vector						
YLR361C	**0.14**	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	**-3.00**
YNR065C	-0.10	**-0.14**	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	**3.47**	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	**3.32**	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	**-3.08**
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	**-3.13**
YJL028W	-0.28	-0.23	-0.19	-0.19	**-0.32**	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	**3.64**	2.74
YPR055W	0.15	0.15	**0.17**	0.09	0.07	0.09	0.07

Clustering yeast genes

- Our goal is to **partition** the set of all yeast genes into *k* disjoint **clusters**
- Genes in the same cluster have similar expression vectors
- In practice, the number of clusters is not known a priori, and so biologists typically apply clustering algorithms to gene expression data for various values of *k*, selecting the value of *k* that makes sense biologically.
- For simplicity, we will assume that *k* is fixed.
- We partition the genes into three clusters indicating increased, decreased, and flat expression during the diauxic shift

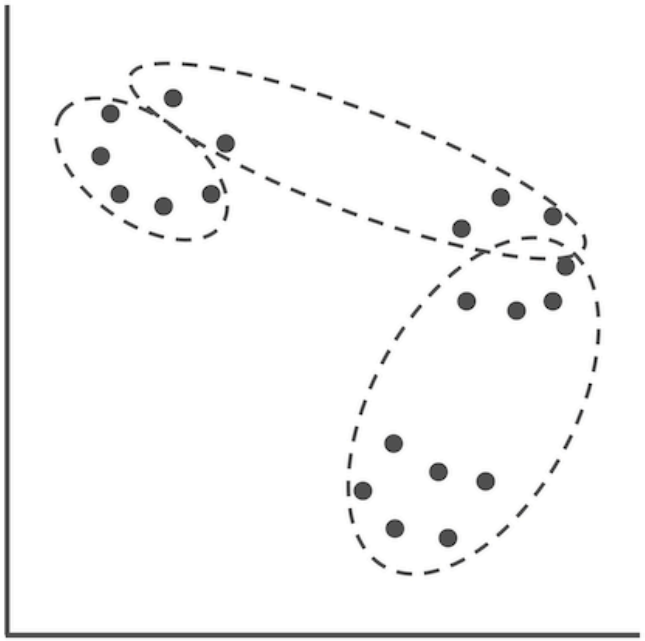
Green genes exhibit increased expression,
Red genes exhibit decreased expression,
and Blue genes exhibit flat expression and are unlikely to be associated with the diauxic shift.



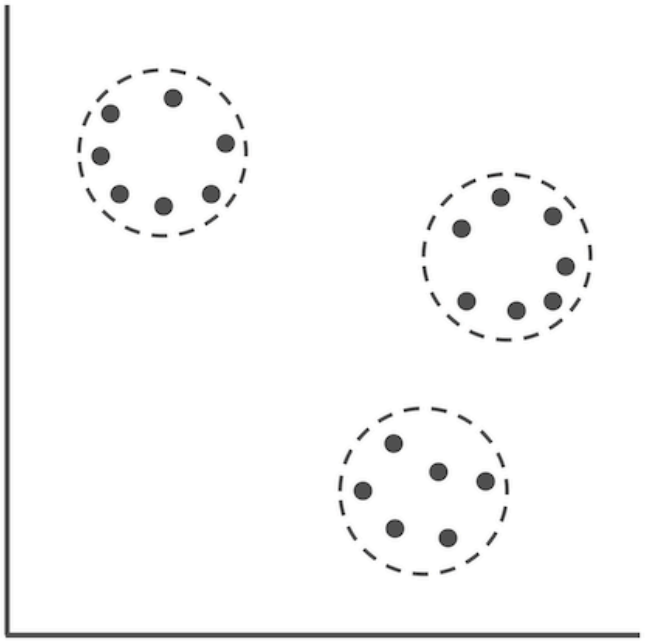
The Good Clustering Principle

- To identify groups of genes with similar expression patterns, we will think of an expression vector of length *m* as a point in *m*-dimensional space;
- Genes with similar expression vectors will therefore form clusters of nearby points.
- **Good Clustering Principle:** *Every pair of points from the same cluster should be closer to each other than any pair of points from different clusters.*

The Good Clustering Principle



Left



Right

Which one satisfies the good clustering principle?

Clustering as an Optimization Problem

- Rather than thinking about clustering as dividing data points *Data* into *k* clusters, we will instead try to select a set *Centers* of *k* points that will serve as the **centers** of these clusters.
- We would like to choose *Centers* so that they minimize some distance function between *Centers* and *Data* over all possible choices of centers.
- First, we define the **Euclidean distance** between points $v = (v_1, \ldots, v_m)$ and $w = (w_1, \ldots, w_m)$ in m-dimensional space, denoted $d(v, w)$, as the length of the line segment connecting these points,

$$d(v, w) = \sqrt{\sum_{i=1}^{m} (v_i - w_i)^2}.$$

Clustering as an Optimization Problem

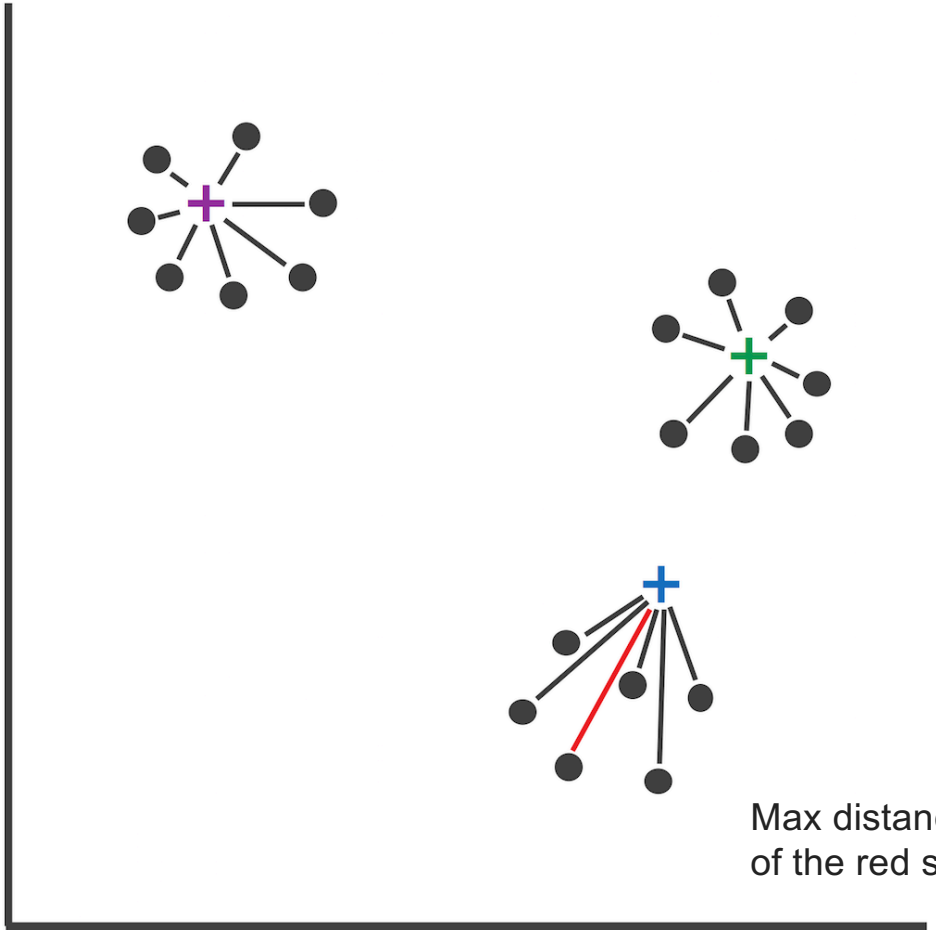
- Next, given a single point *DataPoint* in multi-dimensional space and a set of *k* points Centers, we define the distance from *DataPoint* to *Centers*, denoted *d*(*DataPoint*, *Centers*), as the Euclidean distance from ***DataPoint*** to its **closest center**

$$d(DataPoint,Centers) = \min_{\text{all points } x \text{ from } Centers} d(DataPoint, x).$$

- We now define the distance between all data points *Data* and centers *Centers*. This distance, denoted *MaxDistance*(*Data*, *Centers*), is the maximum of *d*(*DataPoint*, *Centers*) among all data points *DataPoint*,

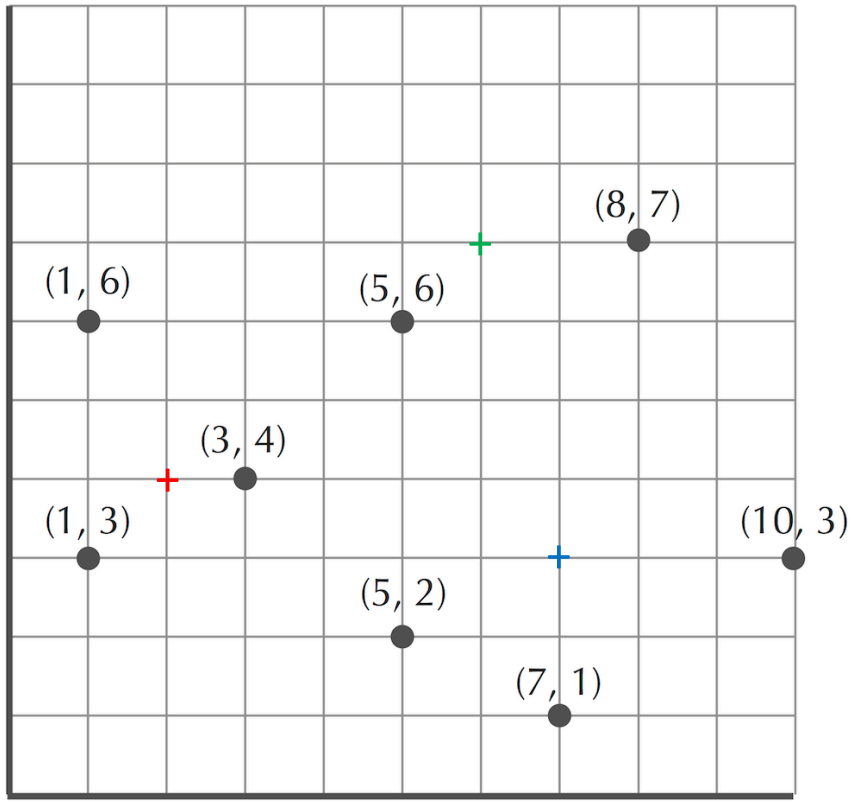
$$MaxDistance(Data, Centers) = \max_{\text{all points } DataPoint \text{ from } Data} d(DataPoints,Centers).$$

The length of the segments in the figure below correspond to *d*(*DataPoint*, *Centers*) for each point *DataPoint*.



Max distance corresponds to the length of the red segment.

- **Exercise Break:** Compute *MaxDistance*(*Data*, *Centers*) for *Data* shown in the figure below and *Centers* (2, 4), (6, 7), and (7, 3).



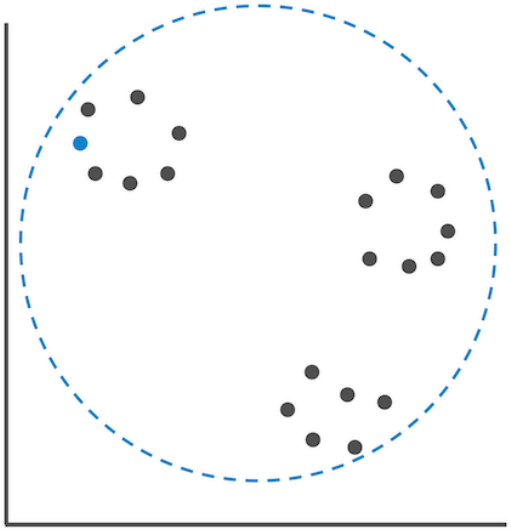
Clustering as an Optimization Problem

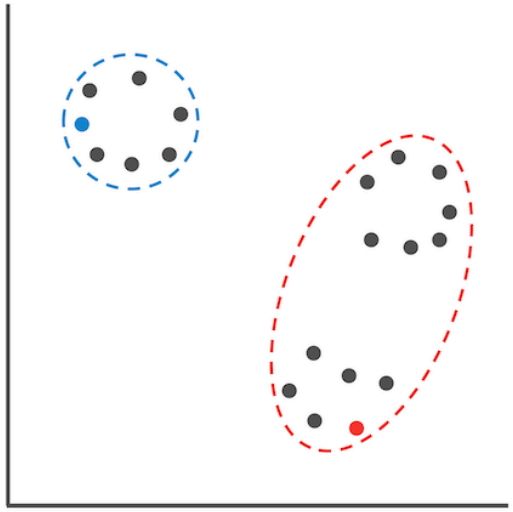
- We can now formulate a well-defined clustering problem.
- ***k*-Center Clustering Problem:** *Given a set of data points, find k centers minimizing the maximum distance between these data points and centers.*
- **Input:** A set of points *Data* and an integer *k.*
- **Output:** A set *Centers* of *k* centers that minimize the distance *MaxDistance*(*DataPoints, Centers*) over all possible choices of *k* centers.

Farthest First Traversal

- The **Farthest First Traversal** heuristic, selects centers from the points in *Data* (instead of from all possible points in *m*-dimensional space).
- It begins by selecting an arbitrary point in *Data* as the first center and iteratively adds a new center as the point in *Data* that is farthest from the centers chosen so far, with ties broken arbitrarily

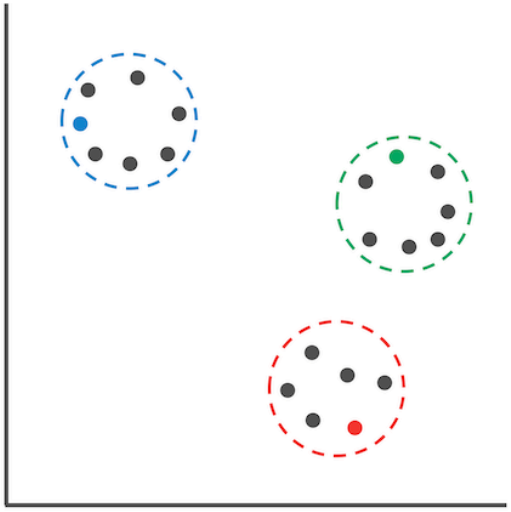
An arbitrary point from the dataset (shown in blue) is selected as the first center. All points belong to a single cluster.

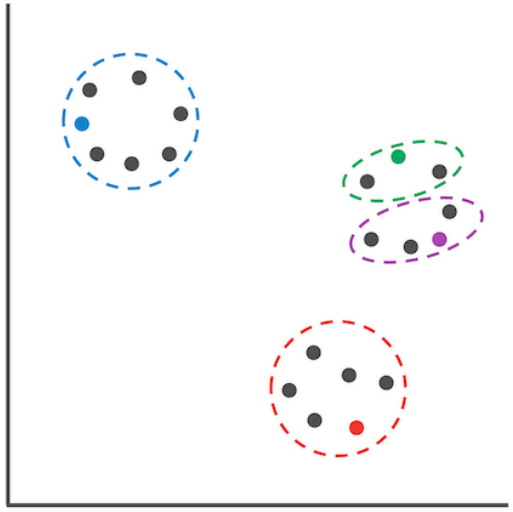




The red point is selected as the second center, since it is the farthest from the blue point

After computing each data point's minimum distance to each of the first two centers, we find that the point with the largest such distance is the green point, which becomes the third center.





The fourth center is shown in purple.

Squared error distortion

- Given a set *Data* of *n* data points and a set *Centers* of *k* centers, the **squared error distortion** of *Data* and *Centers*, denoted *Distortion*(*Data*, *Centers*), is defined as **the mean squared distance from each data point to its nearest center**,

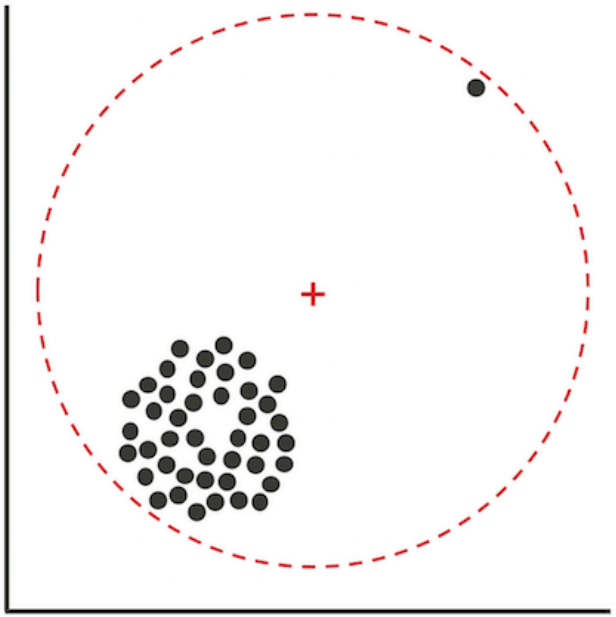
$$Distortion(Data,Centers) = (1/n) \sum_{all\ points\ DataPoint\ in\ Data} d(DataPoint, Centers)^2$$

K-means clustering

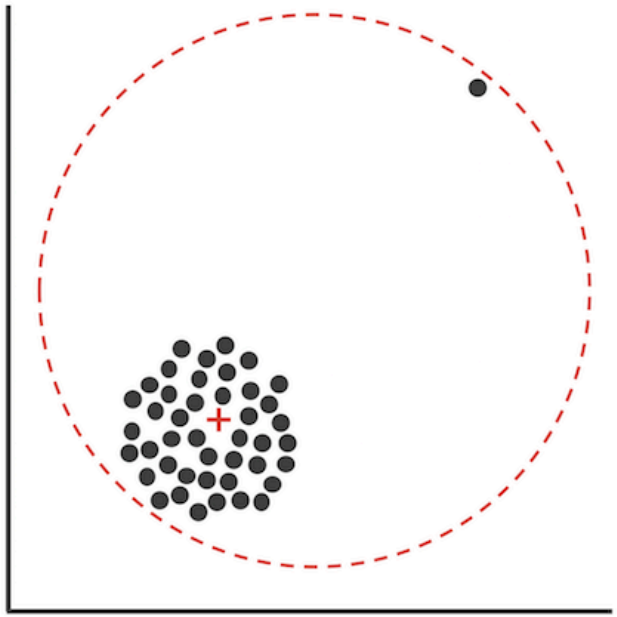
- The squared error distortion leads us to the following modification of the *k*-Centers Clustering Problem.
- ***k*-Means Clustering Problem:** *Given a set of data points, find k center points minimizing the squared error distortion.*

- **Input:** A set of points *Data* and an integer *k*.
- **Output:** A set *Centers* of *k* centers that minimize *Distortion*(*Data*, *Centers*) over all possible choices of *k* centers.

- The key difference between the *k*-Centers and *k*-Means Clustering Problems is that in the latter, the placement of a center is far less affected by outliers



(Left) In the ***k*-Center Clustering Problem**, a cluster's center is chosen so that the maximum distance between the center and any point in the cluster is minimized. As a result, the position of the center can be greatly influenced by outliers.



(Right) In the ***k*-Means Clustering Problem**, the outlier's influence over the placement of the center is much smaller. This is preferable when analyzing biological datasets, in which outliers often correspond to erroneous data.

- **Exercise Break:** Compute the values of ***MaxDistance*(*Data*, *Centers*)** **and *Distortion*(*Data*, *Centers*)** for the eight data points in the figure below and the **three centers (3, 4.5), (6, 1.5), and (9, 5)**. How do these values differ if the centers are instead **(5/3, 13/3), (6.5, 6.5), and (22/3, 2)**?