AIML

Attendance Code:
41096890

K-means clustering: algorithm

- **Step 1**. *Initialization*: Select an initial guess for all μ^0_k for $k=1,2,\ldots,K$, set iteration number $n=0$

Attendance Code:
41096890

K-means clustering: algorithm

- **Step 1**. *Initialization*: Select an initial guess for all μ^0_k for $k=1,2,...,K$, set iteration number $n=0$
- **Step 2**. *Update configuration*: Set $X^{n+1}=0$, except where $k = \text{argmin}_{k'=1,2,...,K} \|x_i - \mu_{k'}\|^2$ for which $X^{n+1}_{ik}=1$

Attendance Code:
41096890

K-means clustering: algorithm

- **Step 1**. *Initialization*: Select an initial guess for all $\mu^0{}_k$ for k=1,2,...,K, set iteration number n=0
- **Step 2**. *Update configuration*: Set X^{n+1}=0, except where $k = \text{argmin}_{k'=1,2,...,K} \|x_i - \mu_{k'}\|^2$ for which $X^{n+1}{}_{ik}$=1
- **Step 3**. *Update centroids*: Compute cluster averages, $\mu^{n+1}{}_k = 1/N_k \sum_{i=1,2,...,N} X^{n+1}{}_{ik} x_i$ where $N_k = \sum_{i=1,2,...,N} X^{n+1}{}_{ik}$,

Attendance Code:
41096890

K-means clustering: algorithm

- **Step 1**. *Initialization*: Select an initial guess for all μ^0_k for k=1,2,...,K, set iteration number n=0
- **Step 2**. *Update configuration*: Set X^{n+1}=0, except where $k = \mathrm{argmin}_{k'=1,2,\ldots,K} \|x_i - \mu_{k'}\|^2$ for which X^{n+1}_{ik}=1
- **Step 3**. *Update centroids*: Compute cluster averages, $\mu^{n+1}_k = 1/N_k \sum_{i=1,2,\ldots,N} X^{n+1}_{ik} x_i$ where $N_k = \sum_{i=1,2,\ldots,N} X^{n+1}_{ik}$,
- **Step 4**. *Convergence check*: If n>0 and $X^{n+1} = X^n$ then exit with solution $X^*=X^{n+1}$ and $\mu^*=\mu^{n+1}$,

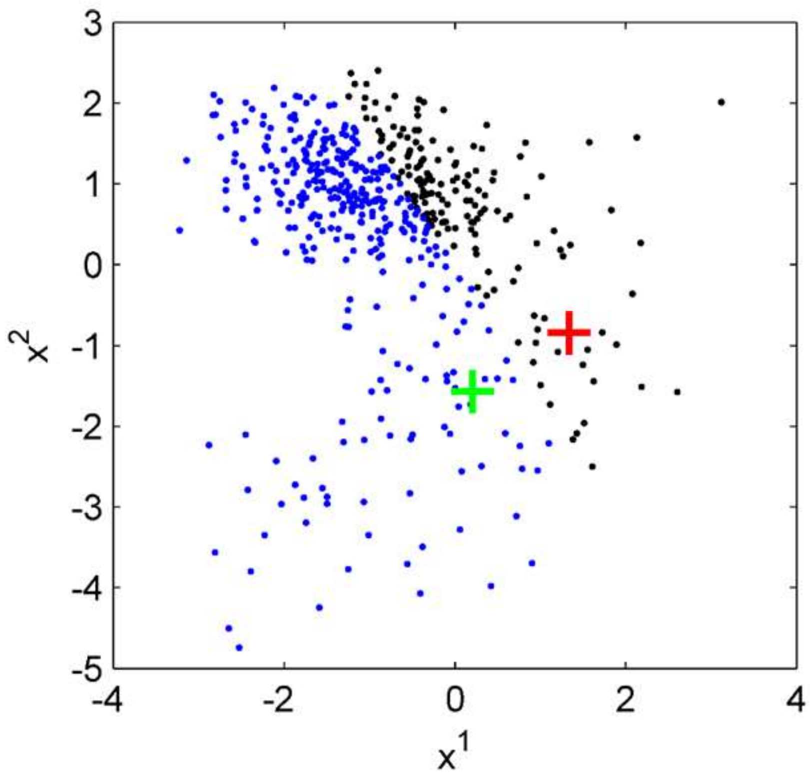
Attendance Code:
41096890

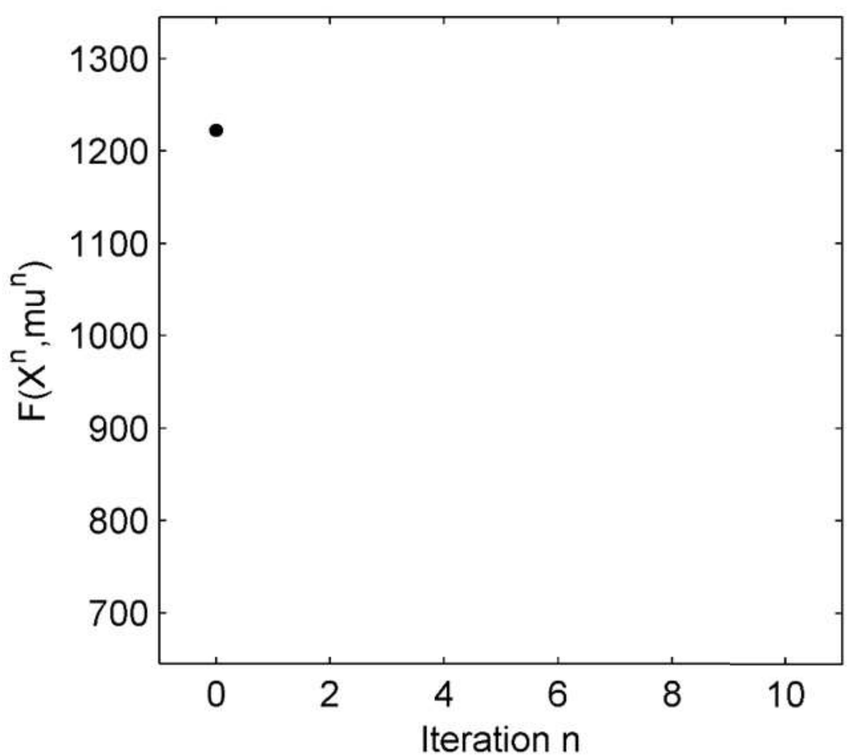
K-means clustering: algorithm

- **Step 1**. *Initialization*: Select an initial guess for all $\mu^0{}_k$ for k=1,2,...,K, set iteration number n=0
- **Step 2**. *Update configuration*: Set X^{n+1}=0, except where $k = \mathrm{argmin}_{k'=1,2,...,K} \|x_i - \mu_{k'}\|^2$ for which $X^{n+1}{}_{ik}$=1
- **Step 3**. *Update centroids*: Compute cluster averages, $\mu^{n+1}{}_k = 1/N_k \sum_{i=1,2,...,N} X^{n+1}{}_{ik} x_i$ where $N_k = \sum_{i=1,2,...,N} X^{n+1}{}_{ik}$,
- **Step 4**. *Convergence check*: If n>0 and $X^{n+1} = X^n$ then exit with solution X^*=X^{n+1} and μ^*=μ^{n+1},
- **Step 5**. *Iteration*: update n=n+1 and go back to step 2.

K-means clustering algorithm in action

$K = 2$, n=0, initial guess for μ_1^0, μ_2^0 (initialize)
Assign X_{ik}^1 to all data points ($i = 1, 2, ..., N$)

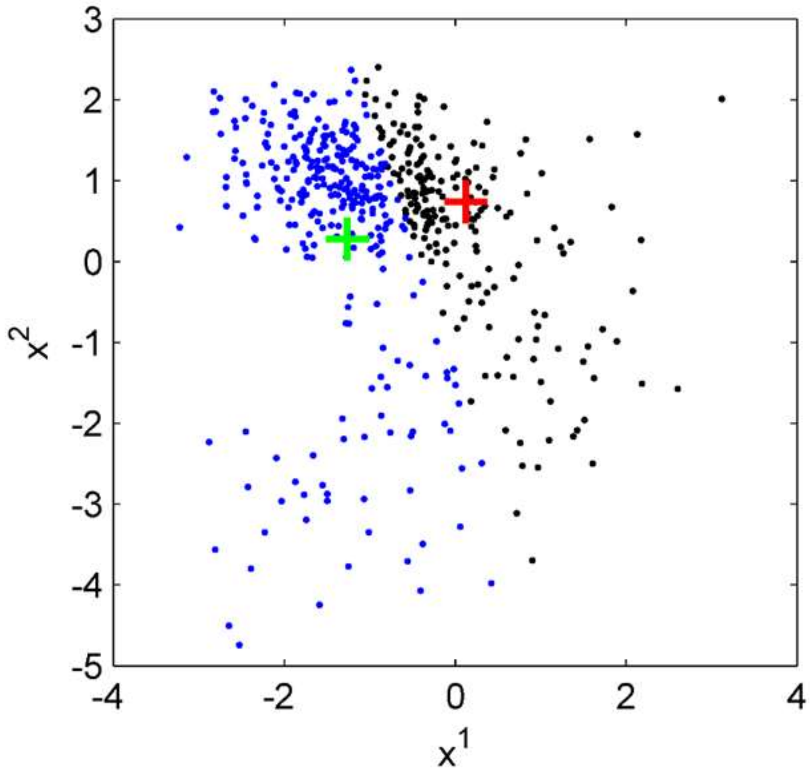


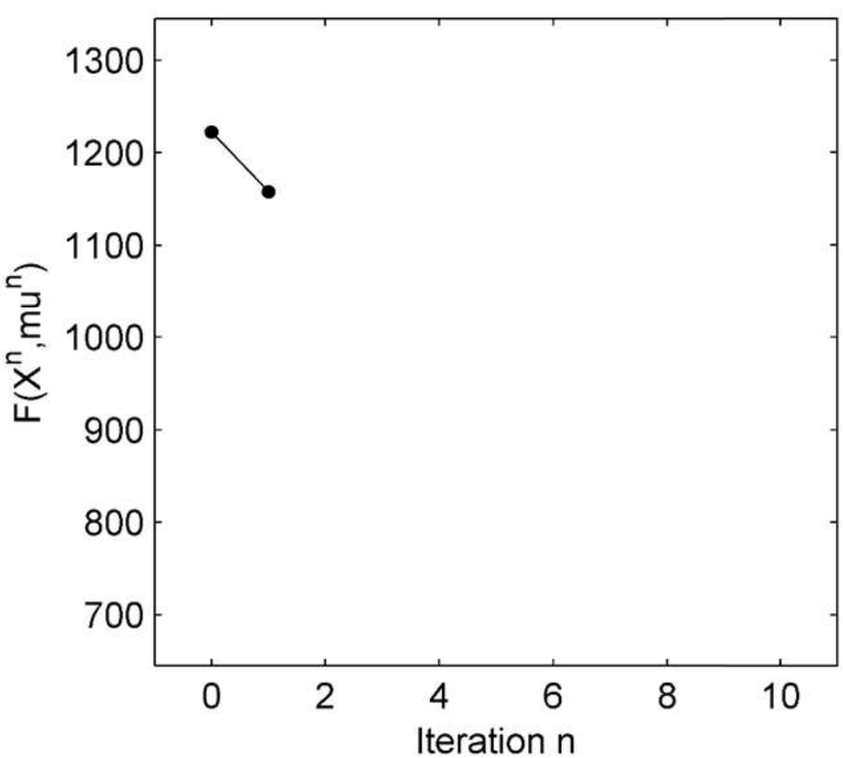


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^1, μ_2^1 with current configuration X_{ik}^1
Assign X_{ik}^2; new configuration is different from X_{ik}^1

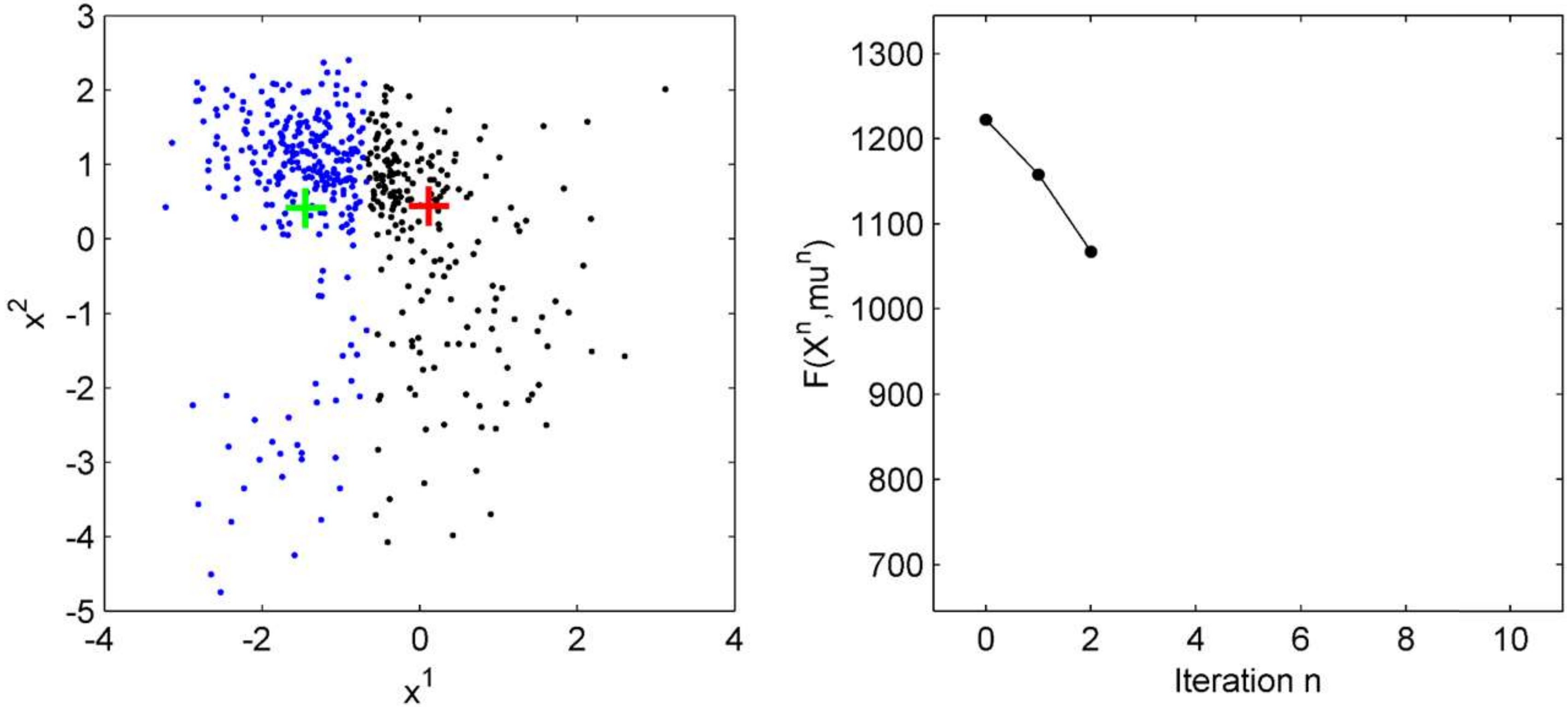




Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^2, μ_2^2 with current configuration X_{ik}^2
Assign X_{ik}^3; new configuration is different from X_{ik}^2

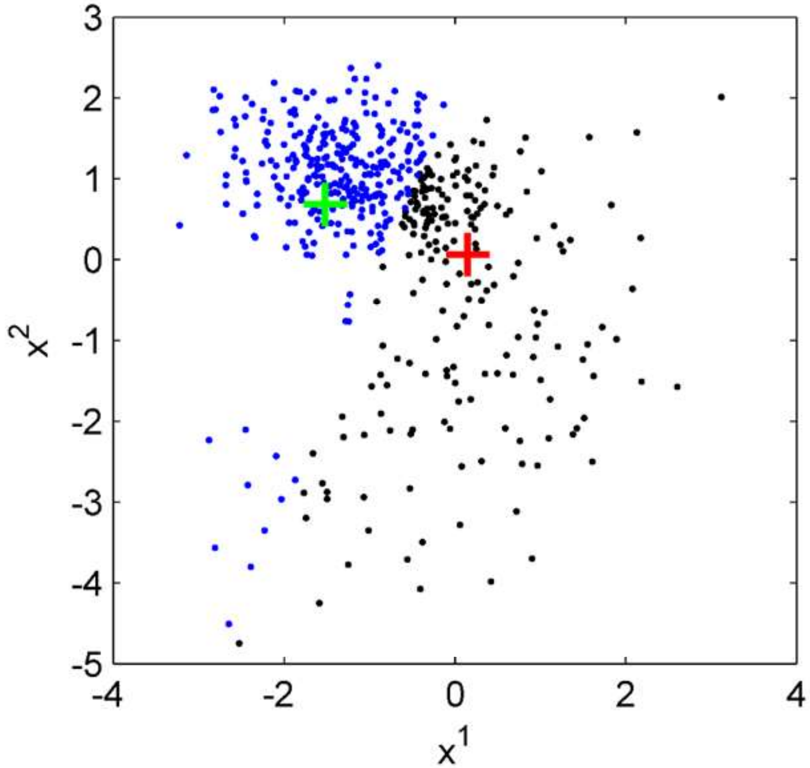


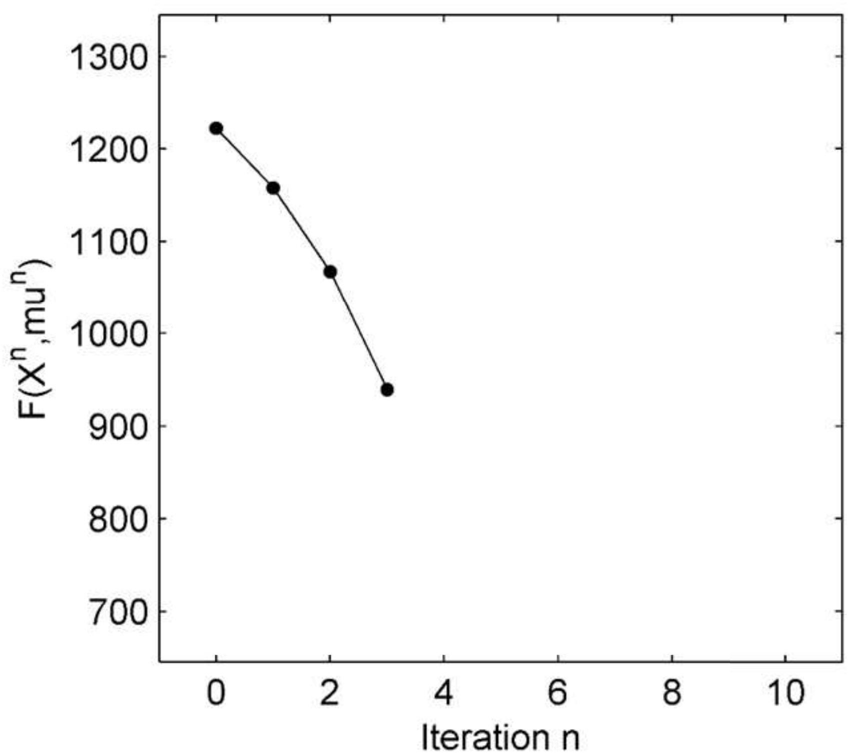
Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^3, μ_2^3 with current configuration X_{ik}^3

Assign X_{ik}^4; new configuration is different from X_{ik}^3

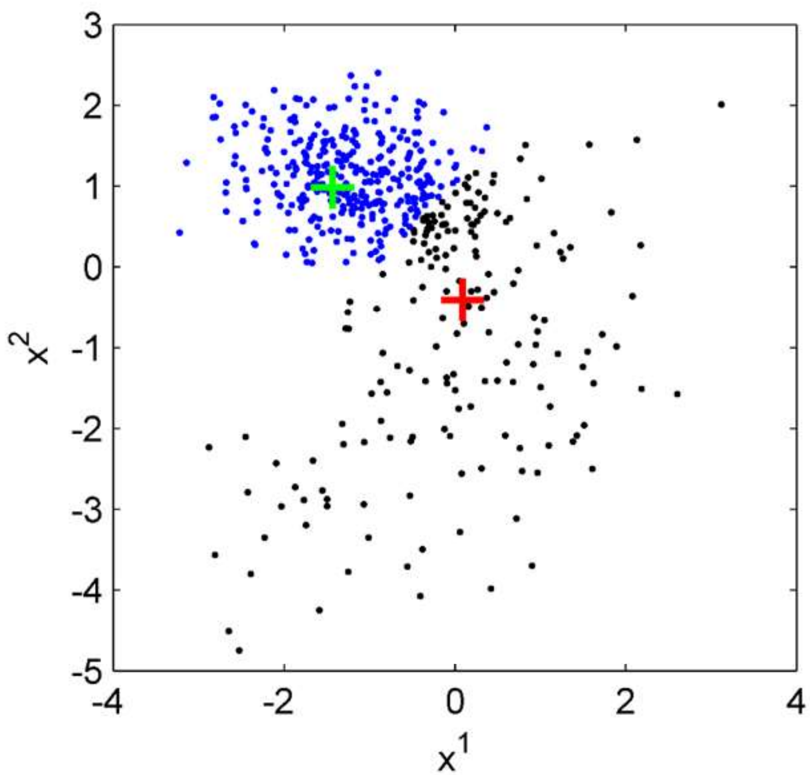


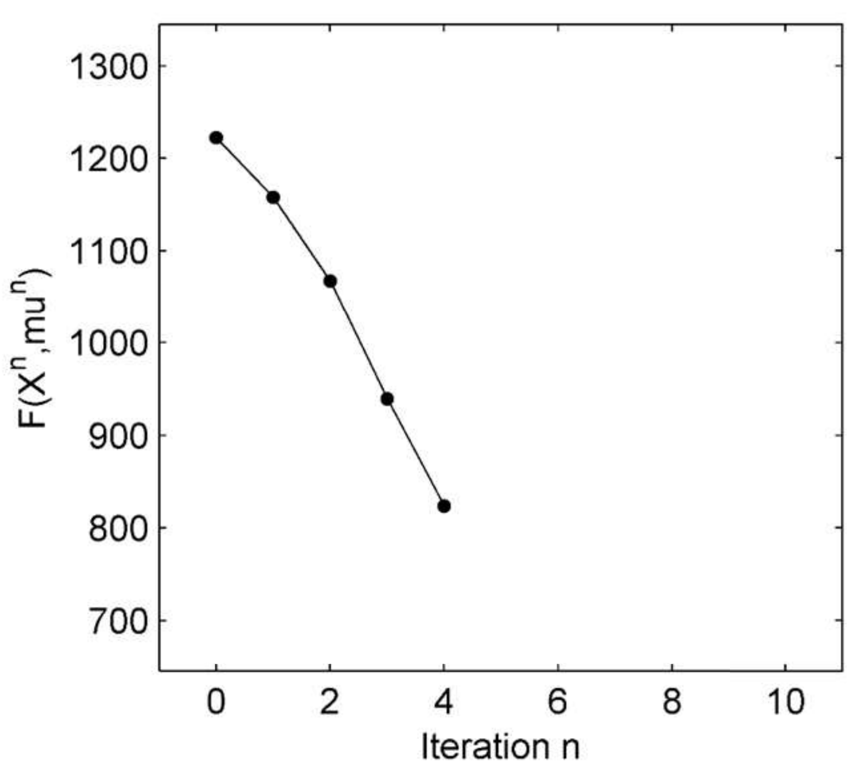


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^4, μ_2^4 with current configuration X_{ik}^4
Assign X_{ik}^5; new configuration is different from X_{ik}^4

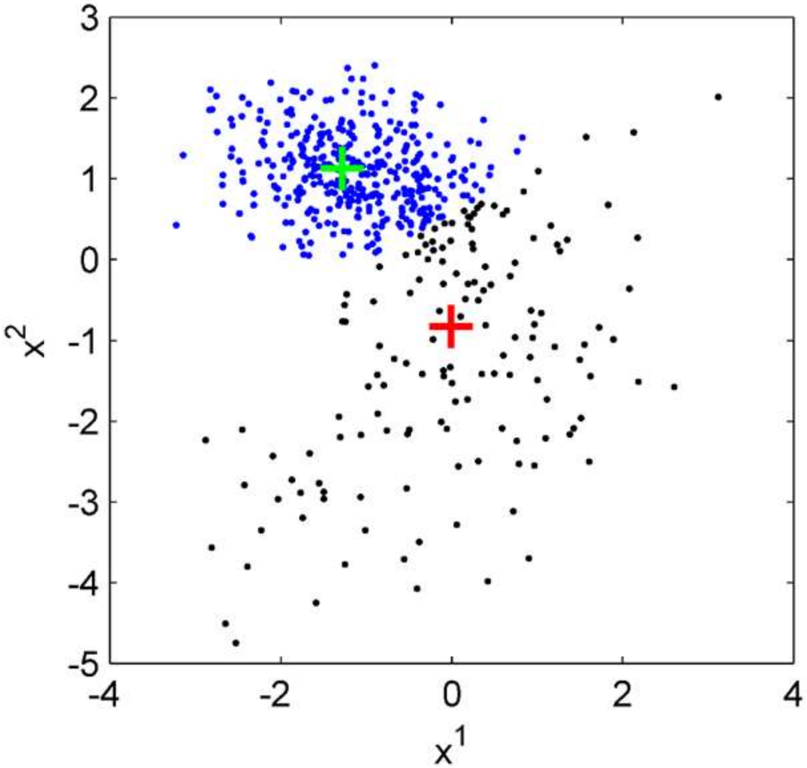


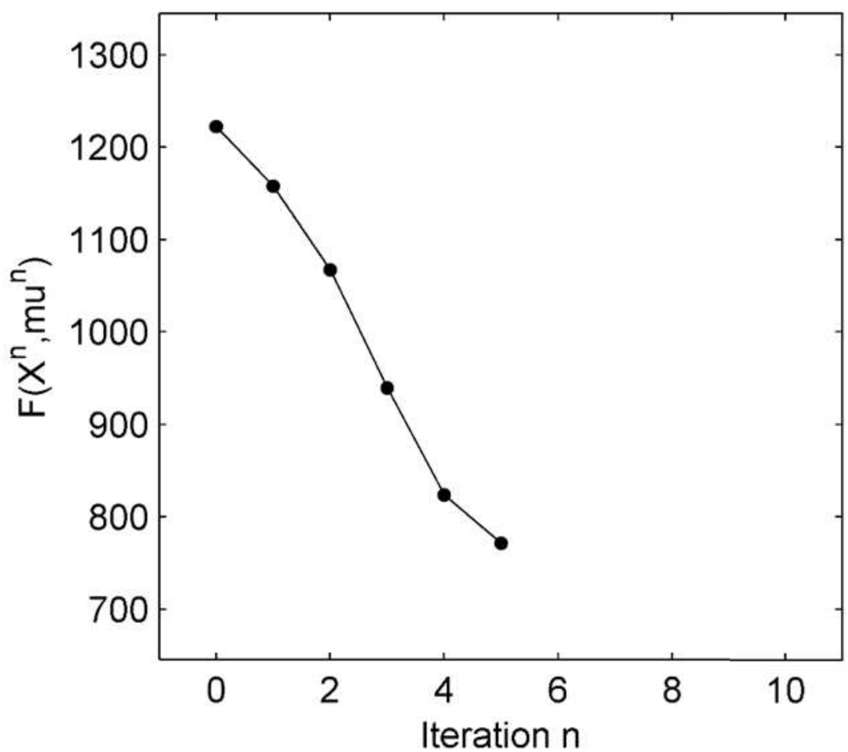


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^5, μ_2^5 with current configuration X_{ik}^5
Assign X_{ik}^6; new configuration is different from X_{ik}^5

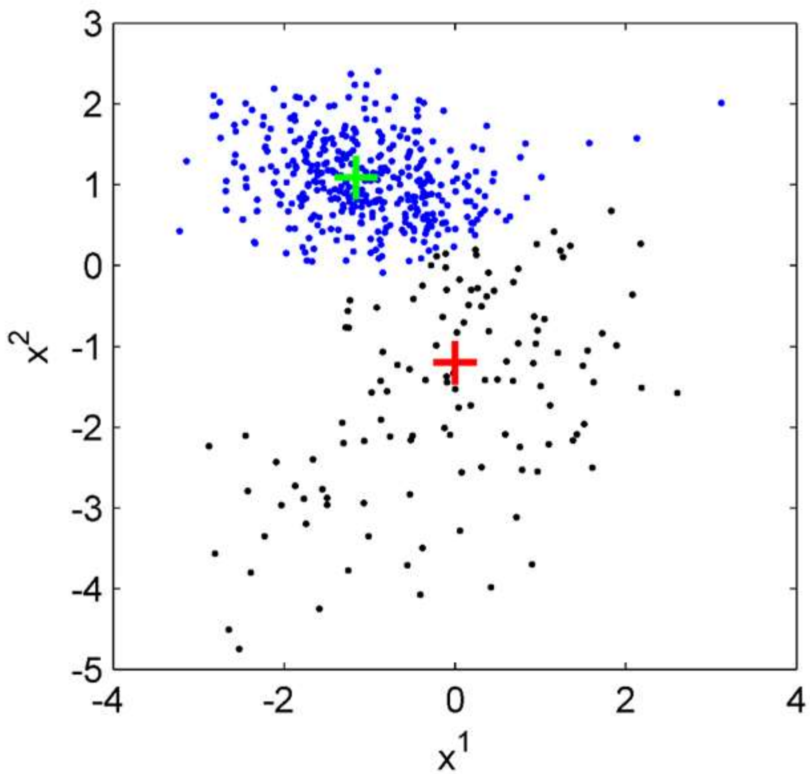


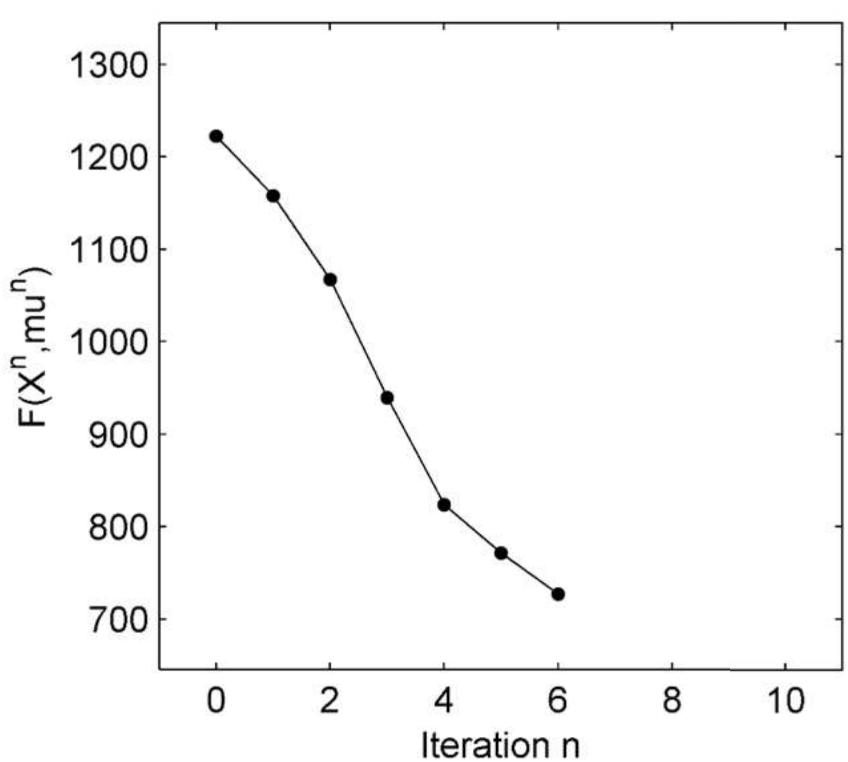


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^6, μ_2^6 with current configuration X_{ik}^6
Assign X_{ik}^7; new configuration is different from X_{ik}^6

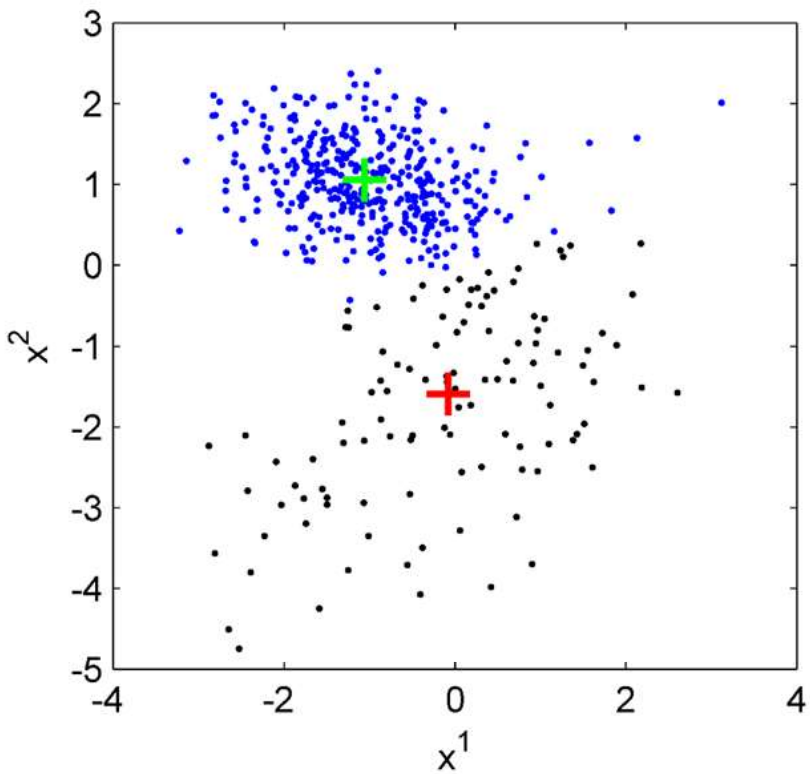


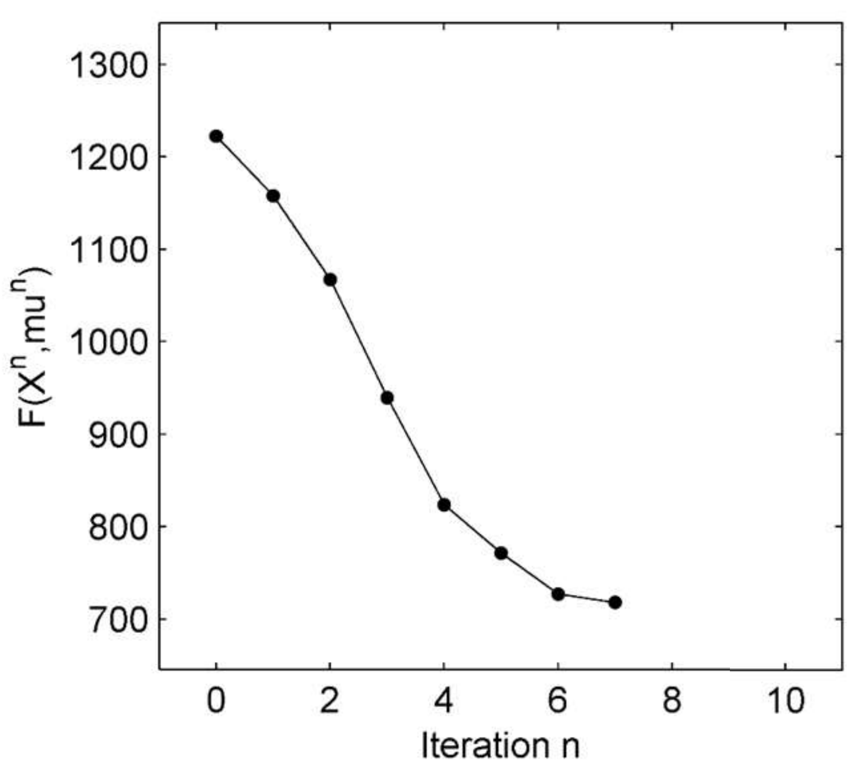


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^7, μ_2^7 with current configuration X_{ik}^7
Assign X_{ik}^8; new configuration is different from X_{ik}^7

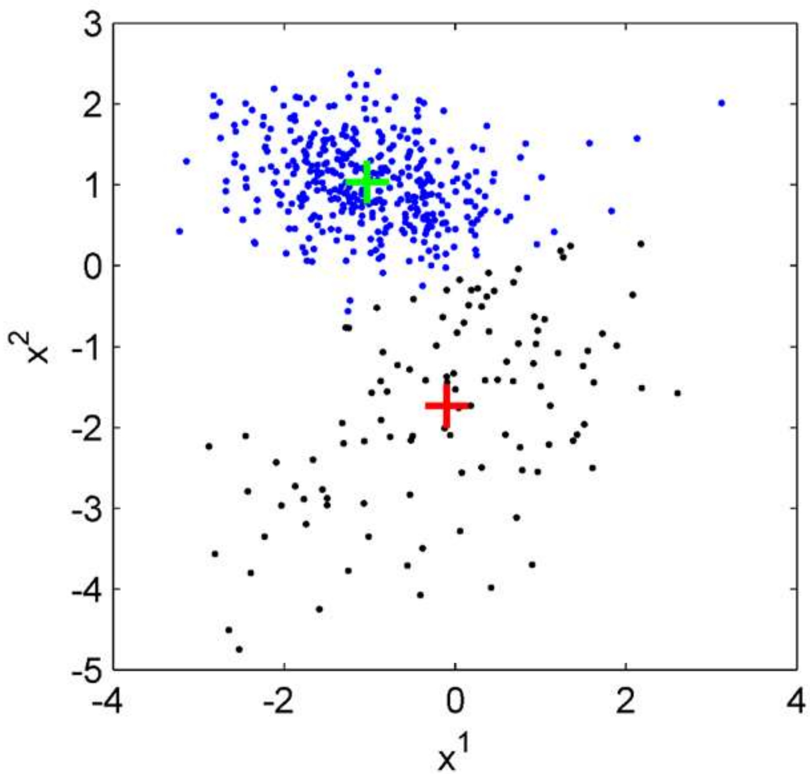


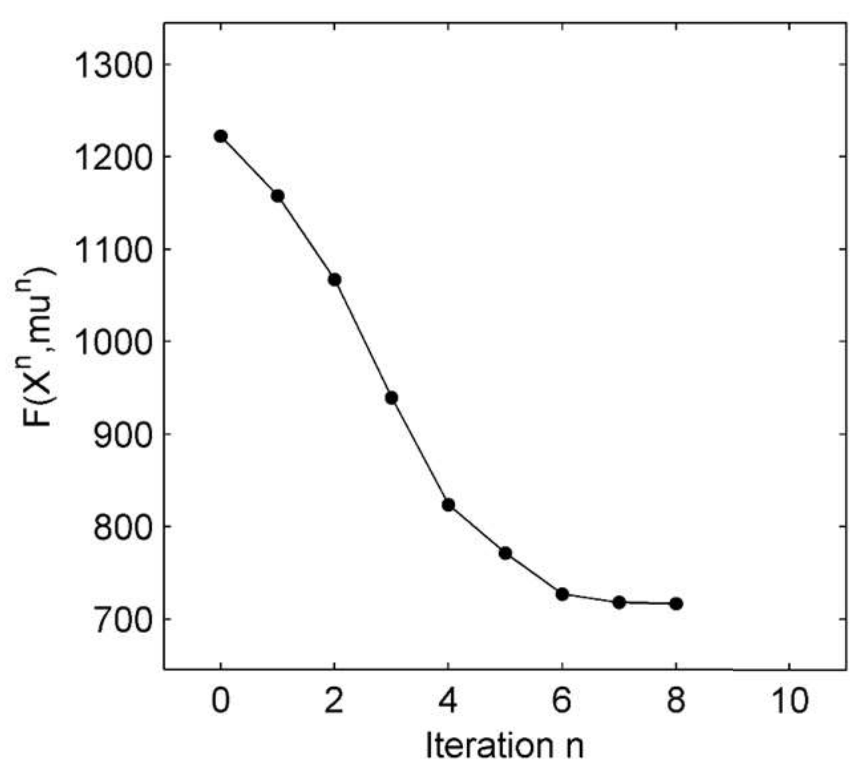


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^8, μ_2^8 with current configuration X_{ik}^8
Assign X_{ik}^9; new configuration is different from X_{ik}^8

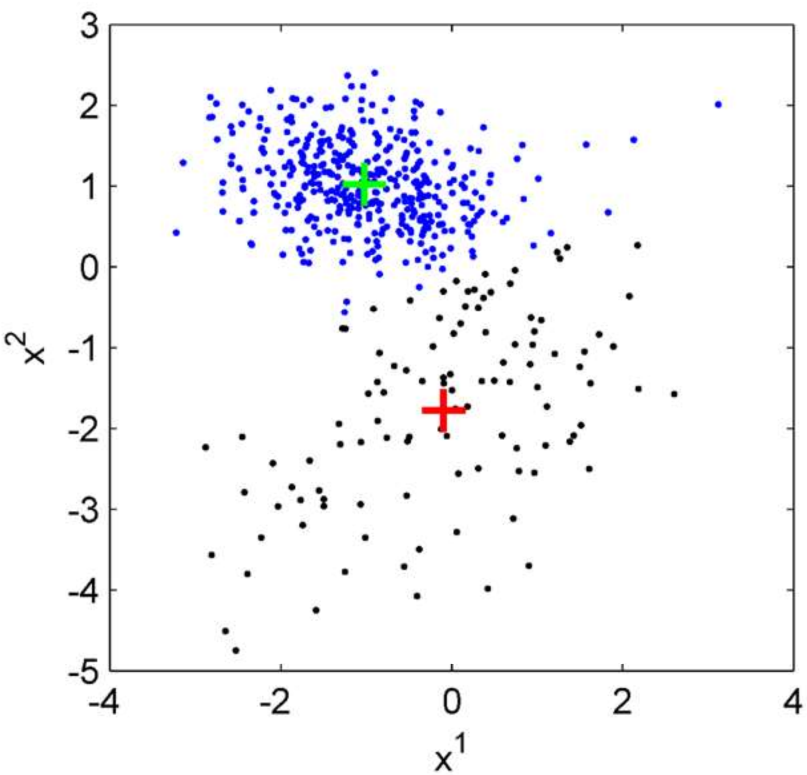


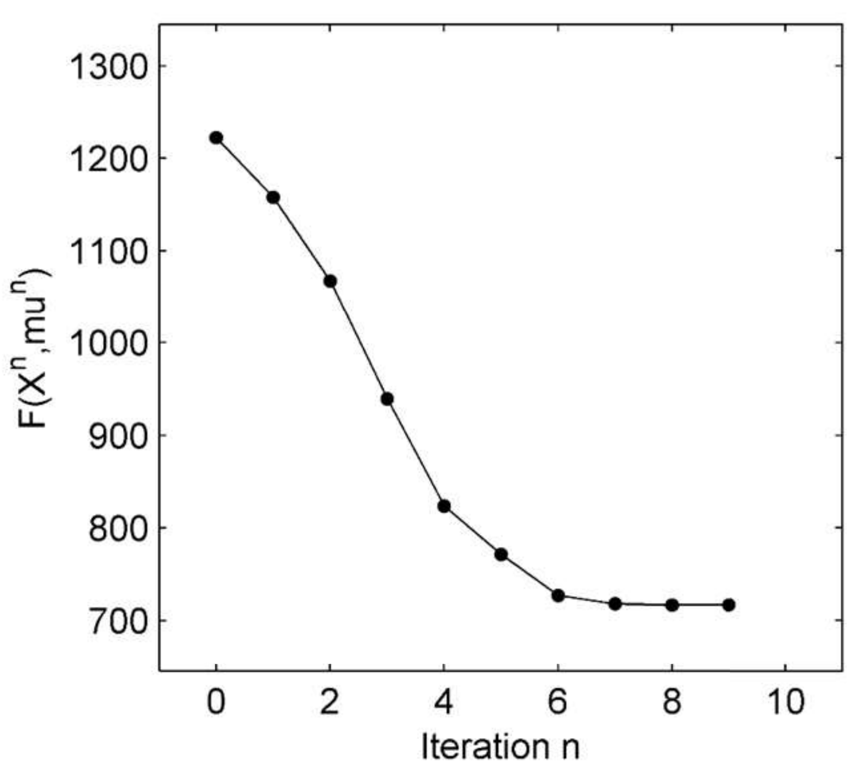


Attendance Code:
41096890

K-means clustering algorithm in action

Update centroids μ_1^9, μ_2^9 with current configuration X_{ik}^9
Assign $X_{ik}^{10} = X_{ik}^9 = \mathrm{X}^\star$





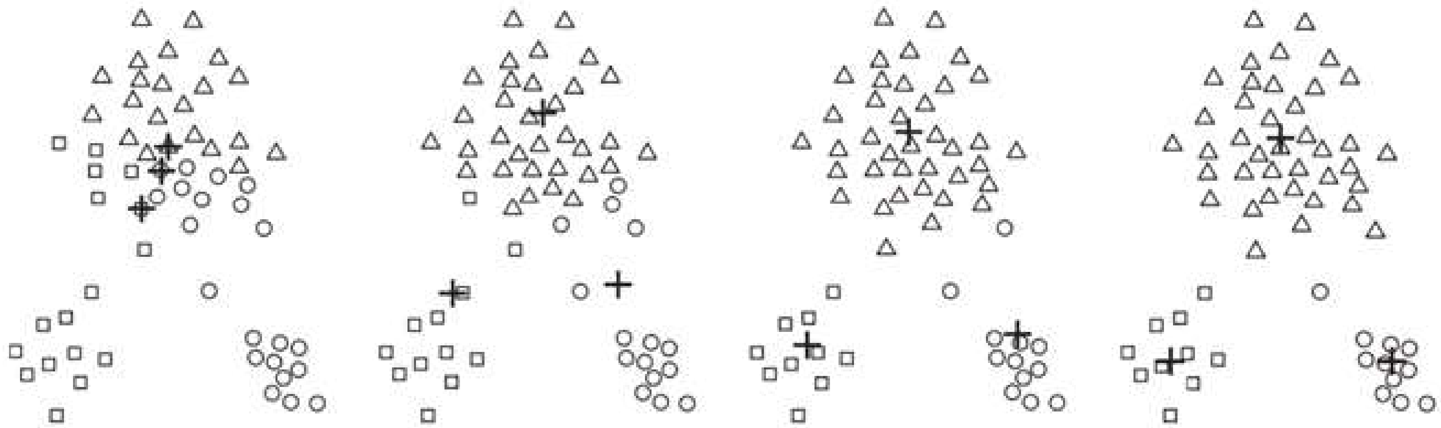
Attendance Code:
41096890

K-means clustering: analysis

- We can show that $F(X^{n+1}, \mu^{n+1}) \leq F(X^n, \mu^n)$ i.e. the *K*-means objective function is **never increasing** (because given μ we can find the X with globally smallest F, and vice-versa)
- Thus, the K-means algorithm always **converges** on a **fixed point** (that is, it finds a **minima** of F and it would stay there for all subsequent iterations)
- But we cannot know if the local minima is also a global minima (this is an approximate method)
- Do not know how long it takes to converge (typically 7-15 iterations from experience, but no guarantees)
- Potential for pathological results (i.e. empty clusters)

Illustration

#Code



Initialization Iteration 1 Iteration 2 Iteration 3

UNIVERSITY OF
BIRMINGHAM
DUBAI

Example 1: Clustering of Medicines (K=2)

#Code

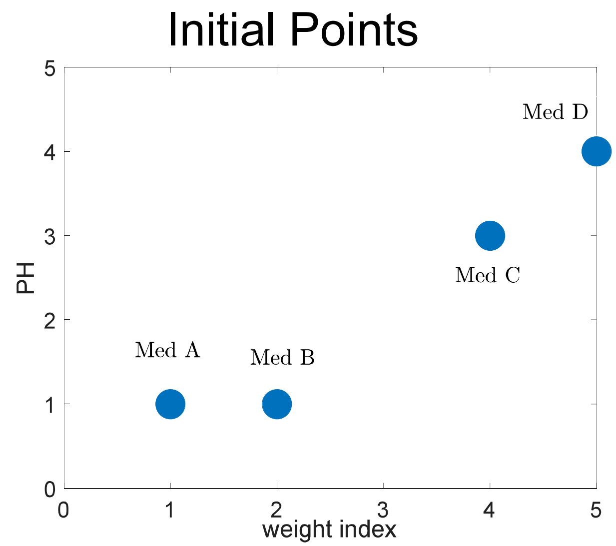
	Weight index	PH
Med A	1	1
Med B	2	1
Med C	4	3
Med D	5	4

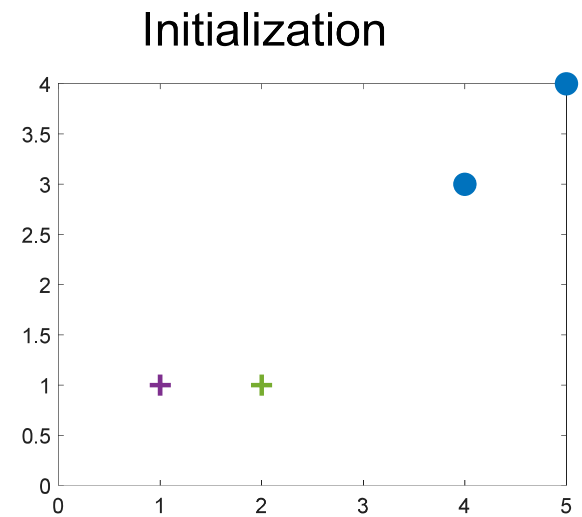
UNIVERSITY OF BIRMINGHAM DUBAI

Example 1: Clustering of Medicines (K=2)

#Code

	Weight index	PH
Med A	1	1
Med B	2	1
Med C	4	3
Med D	5	4





Initialization: Initial centroids be Med A and Med B i.e, $c_1 = (1,1)$ and $c_2 = (2,1)$

UNIVERSITY OF BIRMINGHAM DUBAI

Iteration 1: Step 1

#Code

1. Calculate (Euclidean) distance of each point to cluster centroids to form an Object-Centroid Distance Matrix:

	Med A	Med B	Med C	Med D
c_1	0		13	25
c_2		0	8	18

$$d_{Euc}(Med\ C, C_1)^2 = (4-1)^2 + (3-1)^2 = 13$$
$$d_{Euc}(Med\ C, C_2)^2 = (4-2)^2 + (3-1)^2 = 8$$
$$d_{Euc}(Med\ D, C_1)^2 = (5-1)^2 + (4-1)^2 = 25$$
$$d_{Euc}(Med\ D, C_2)^2 = (5-2)^2 + (4-1)^2 = 18$$

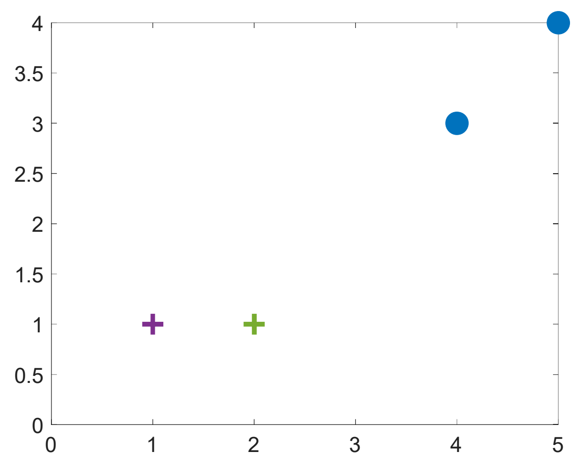
	Weight index	PH
Med A	1	1
Med B	2	1
Med C	4	3
Med D	5	4

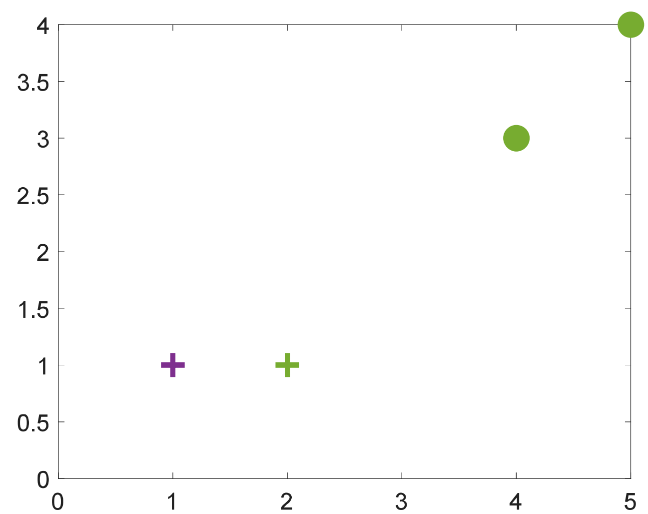
Thus, Medicines B, C and D assigned to Cluster 2.

UNIVERSITY OF BIRMINGHAM DUBAI

#Code

After step 1 of iteration 1





UNIVERSITY OF BIRMINGHAM DUBAI

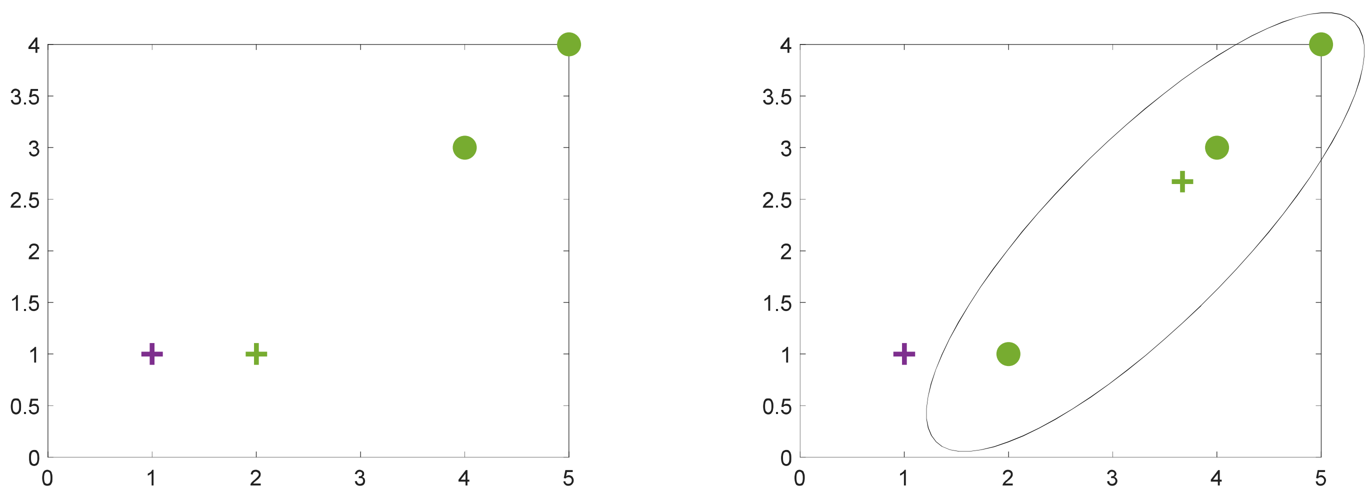
#Code

Iteration 1: Step 2

2. Update the centroids of the cluster.

$$c_1 = c_1 \ (same);\ c_2 = \frac{Med\ B + Med\ C + Med\ D}{3}$$

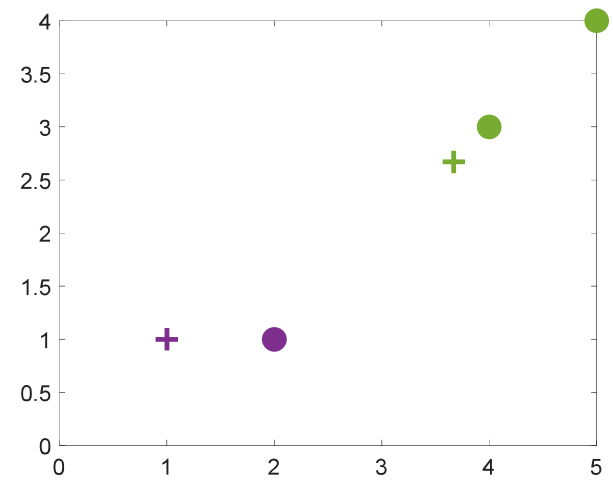
$$= \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = (3.67, 2.67)$$



Iteration 2: Step 1

1. Calculate distance of each point to new cluster centroids.

	Med A	Med B	Med C	Med D
c_1	0	1	13	25
c_2	9.92	5.56	0.22	3.53



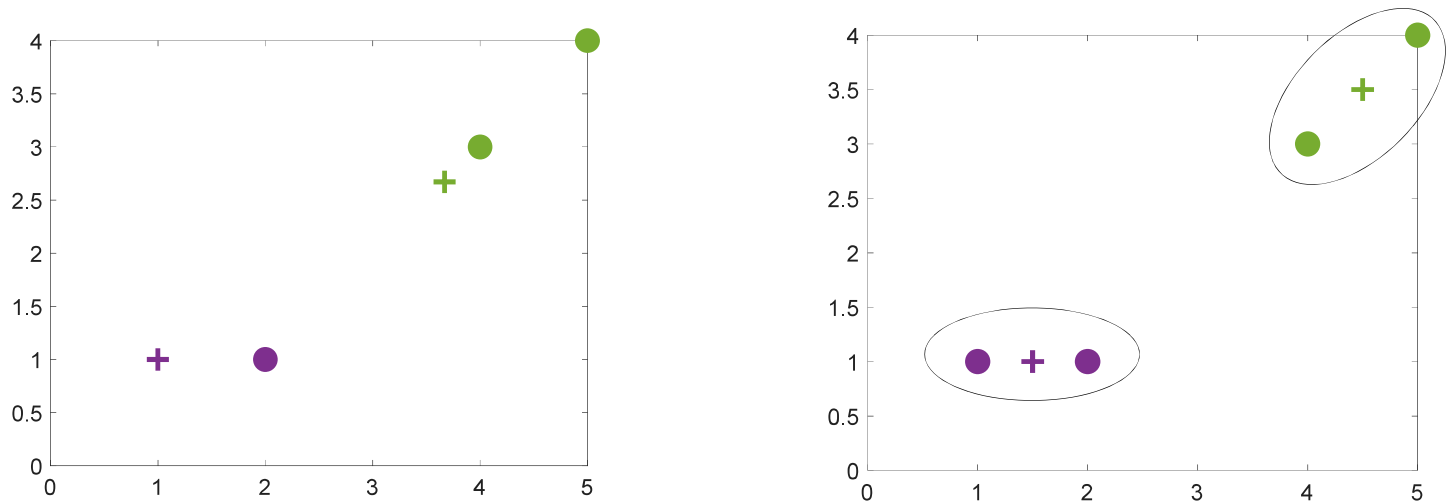
Med B is thus moved to cluster 1.

UNIVERSITY OF BIRMINGHAM DUBAI

Iteration 2: Step 2

2. Update the centroids of the cluster.

- $c_1 = \frac{Med\ A + Med\ B}{2} = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5,1)$
- $c_2 = \frac{Med\ C + Med\ D}{2} = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4.5,3.5)$



#Code

- Repeat the same steps in iteration 3
- Note that cluster assignments do not change
- Algorithm converged.

UNIVERSITY OF BIRMINGHAM DUBAI

Attendance Code:
41096890

K-means clustering: example

- **Histological Analysis:** tissue stained with hemotoxylin and eosin (H&E)

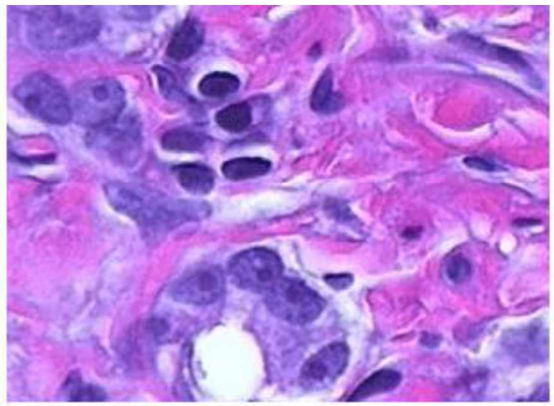
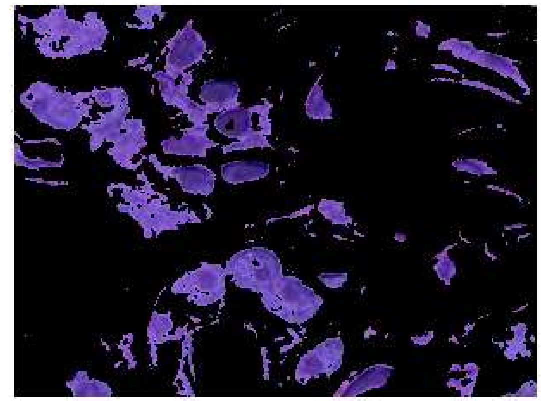
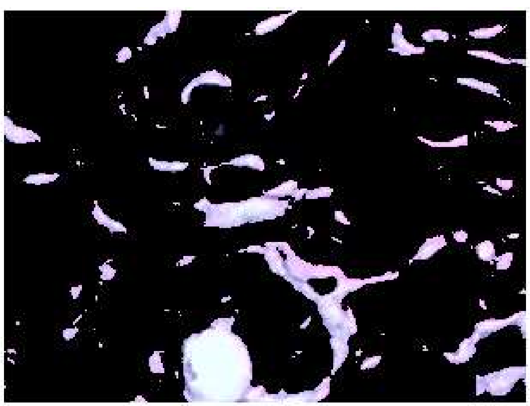


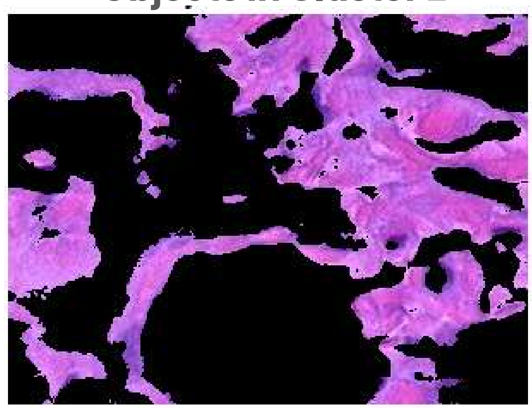
Image courtesy of Alan Partin, Johns Hopkins University



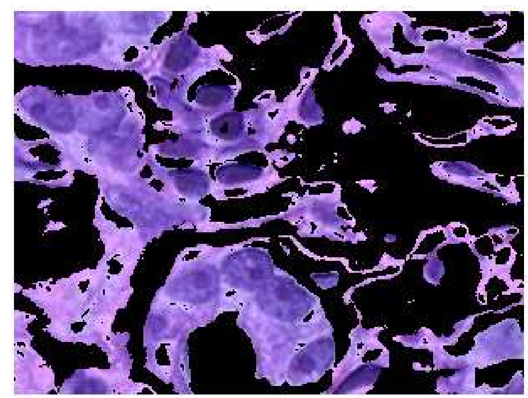
Objects in Cluster 1



Objects in Cluster 2



Objects in Cluster 3



AIML

#Code

Machine learning (ML): overview

- Given some **training data**, ML training finds optimal **model parameters** such that the **prediction error** is minimized; the smaller the error function the better

$$w^{\star} = \arg\min_{w' \in \mathcal{W}} F\left(w'\right)$$

- Then use the trained parameters in the model, to make predictions about new unseen **test data**
- ML algorithms usually categorized according to availability of labelled data: **supervised**, **unsupervised**, **self-supervised**, **transfer learning**

AIML

#Code

Sequential gradient descent (SGD)

- General algorithm for finding a value of the model parameters w such that error function $F(w)$ is minimized, expressed using **multivariable calculus** as $F_w(w) = 0$, where F_w is the (**partial**) **derivative** of F with respect to vector w
- Idea: starting with a guess for w_n, take a "step" in direction of **steepest descent** of the loss function, $-F_w(w_n)$, use this as a better guess w_{n+1}
- Size of the step, $\alpha > 0$ ("learning rate"), determines how quickly the minimum is reached, but can **overshoot** and also **diverge** if α is too large; not guaranteed to find the minimum, unless the error function is **convex** with respect to w

AIML

#Code

SGD: algorithm

- **Step 1**. *Initialization*: Select an initial guess for w_0, a convergence tolerance $\varepsilon > 0$, step size (learning rate) parameter $\alpha > 0$, set iteration number n=0
- **Step 2**. *Gradient descent step*: Compute new model parameters,

$$w_{n+1} = w_n - \alpha\, F_w(w_n)$$

- **Step 3**. *Convergence test*: Compute new loss function value $F(w_{n+1})$, and loss function improvement, $\Delta F = |F(w_{n+1}) - F(w_n)|$ and if $\Delta F < \varepsilon$, exit with solution $w^*=w_{n+1}$
- **Step 4**. *Iteration*: update $n=n+1$ and go to step 2.

AIML

#Code

SGD Example: Regression

(Linear Regression)

- Yield prediction of very early potato cultivars before harvest.
- Potato yielding is associated with the amount of nitrogen fertilizer and average temperature in the season. Here is some data collected from several farms.

- Task: predict how many potatoes will yield in a given farm by measuring the independent variables.

Farm	Fertilizer (10*kg)	Avg. Temp. (°C)	Potato (10*t)
A	12.1	12.5	36.5
B	8.7	11.2	26.4
C	14.0	14.7	42.0
D	9.5	11.8	28.8
E	13.2	13.6	39.7

AIML

#Code

SGD Example: Regression

(Linear Regression)

- Two independent features
  - Fertilizer (x^1)
  - Average Temperature (x^2)

- Regression model

$$f(w,x) = w_1 + w_2x^1 + w_3x^2 = [w_1 \quad w_2 \quad w_3]\, x = w^T x \qquad x = \begin{bmatrix} 1 \\ x^1 \\ x^2 \end{bmatrix}$$

- Sum-of-squares error function

$$F(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2 \qquad F_w(w) = 2\sum_{i=1}^{N} (w^T x_i - y_i) x_i$$

	x^1	x^2	y
$i=1$	12.1	12.5	36.5
$i=2$	8.7	11.2	26.4
$i=3$	14.0	14.7	42.0
$i=4$	9.5	11.8	28.8
$i=5$	13.2	13.6	39.7

AIML

SGD Example: Regression

(Linear Regression)

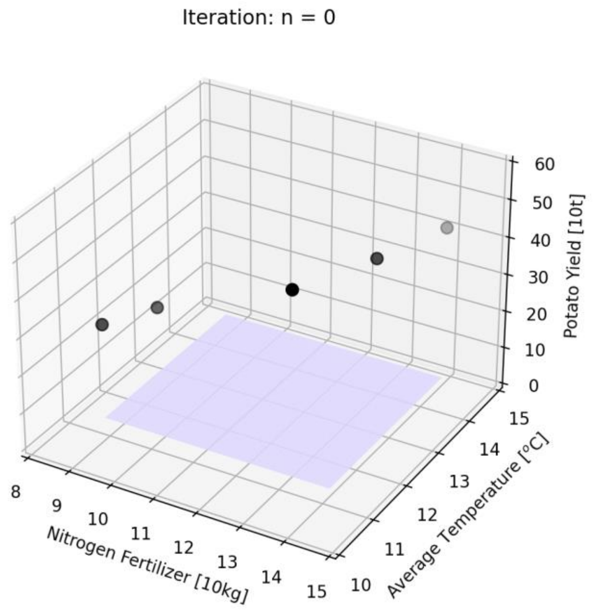
- Initial guess: $w_0^T = [1 \quad 0 \quad 0]$
- Tolerance: $\epsilon = 1$
- Learning rate: $\alpha = 0.0005$

- Error function evaluation:

$$F(w_0) = \sum_{i=1}^{5}\left([1 \quad 0 \quad 0]\begin{bmatrix}1\\ x_i^1\\ x_i^2\end{bmatrix} - y_i\right)^2 = 5856.9$$

- Gradient evaluation

$$F_w(w_0) = 2\sum_{i=1}^{N}(w^T x_i - y_i)\begin{bmatrix}1\\ x_i^1\\ x_i^2\end{bmatrix} = \begin{bmatrix}-336.8\\ -3998.9\\ -4370.58\end{bmatrix}$$



- Gradient descent step

$$w_1 = w_0 - \alpha F_w(w_0) = \begin{bmatrix}1\\ 0\\ 0\end{bmatrix} - 0.0005\begin{bmatrix}-336.8\\ -3998.9\\ -4370.58\end{bmatrix} = \begin{bmatrix}1.168\\ 1.999\\ 2.185\end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_1) - F(w_0)| = |1513.7 - 5856.9| = 4343.2$$

SGD Example: Regression

(Linear Regression)

- Current weight:

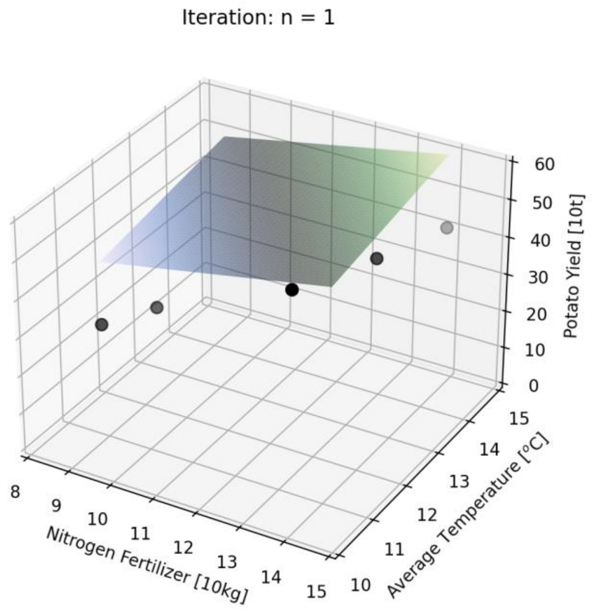
$$w_1^T = [1.168 \quad 1.999 \quad 2.185]$$

- Error function evaluation:

$$F(w_1) = \sum_{i=1}^{5}\left(w_1^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i\right)^2 = 1513.7$$

- Gradient evaluation

$$F_w(w_1) = 2\sum_{i=1}^{N}(w^T x_i - y_i)\begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} 173.7 \\ 2011.0 \\ 2227.1 \end{bmatrix}$$



- Gradient descent step

$$w_2 = w_1 - \alpha F_w(w_1) = \begin{bmatrix} 1.168 \\ 1.999 \\ 2.185 \end{bmatrix} - 0.0005\begin{bmatrix} 173.7 \\ 2011.0 \\ 2227.1 \end{bmatrix} = \begin{bmatrix} 1.082 \\ 0.994 \\ 1.072 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_2) - F(w_1)| = |399.2 - 1513.7| = 1114.5 > \epsilon$$

AIML

SGD Example: Regression

(Linear Regression)

- Current weight:

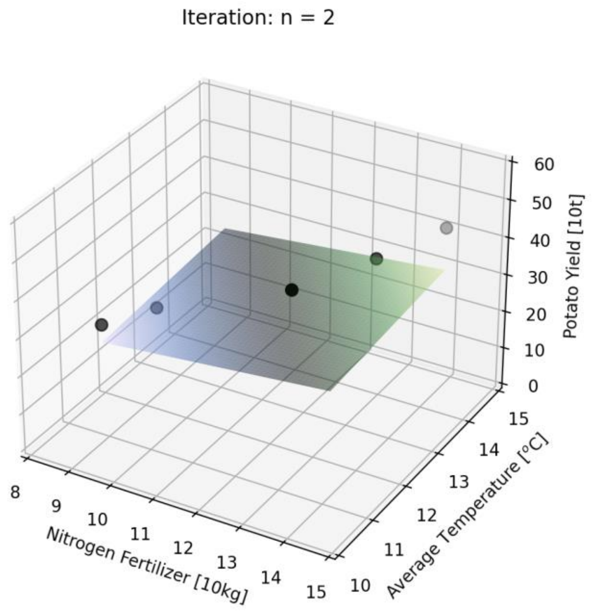
$$w_2^T = [1.082 \quad 0.994 \quad 1.072]$$

- Error function evaluation:

$$F(w_2) = \sum_{i=1}^{5}\left(w_2^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i\right)^2 = 399.2$$

- Gradient evaluation

$$F_w(w_2) = 2\sum_{i=1}^{N}(w^T x_i - y_i)\begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} -84.923 \\ -1033.4 \\ -1115.0 \end{bmatrix}$$



- Gradient descent step

$$w_3 = w_2 - \alpha F_w(w_2) = \begin{bmatrix} 1.082 \\ 0.994 \\ 1.072 \end{bmatrix} - 0.0005\begin{bmatrix} -84.923 \\ -1033.4 \\ -1115.0 \end{bmatrix} = \begin{bmatrix} 1.124 \\ 1.511 \\ 1.629 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_3) - F(w_2)| = |113.1 - 399.2\,| = 286.1 > \epsilon$$

SGD Example: Regression

(Linear Regression)

- Current weight:

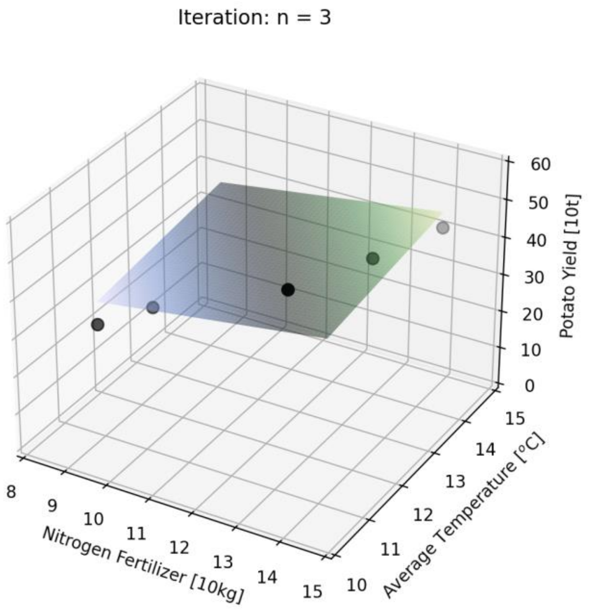
$$w_3^T = [1.124 \quad 1.511 \quad 1.629]$$

- Error function evaluation:

$$F(w_3) = \sum_{i=1}^{5} \left(w_3^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i \right)^2 = 113.1$$

- Gradient evaluation

$$F_w(w_3) = 2 \sum_{i=1}^{N} (w^T x_i - y_i) \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} 46.058 \\ 508.84 \\ 577.9 \end{bmatrix}$$



- Gradient descent step

$$w_4 = w_3 - \alpha F_w(w_3) = \begin{bmatrix} 1.124 \\ 1.511 \\ 1.629 \end{bmatrix} - 0.0005 \begin{bmatrix} 46.058 \\ 508.84 \\ 577.9 \end{bmatrix} = \begin{bmatrix} 1.101 \\ 1.256 \\ 1.340 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_4) - F(w_3)| = |39.661 - 113.1\,| = 73.5 > \epsilon$$

AIML

SGD Example: Regression

(Linear Regression)

- Current weight:

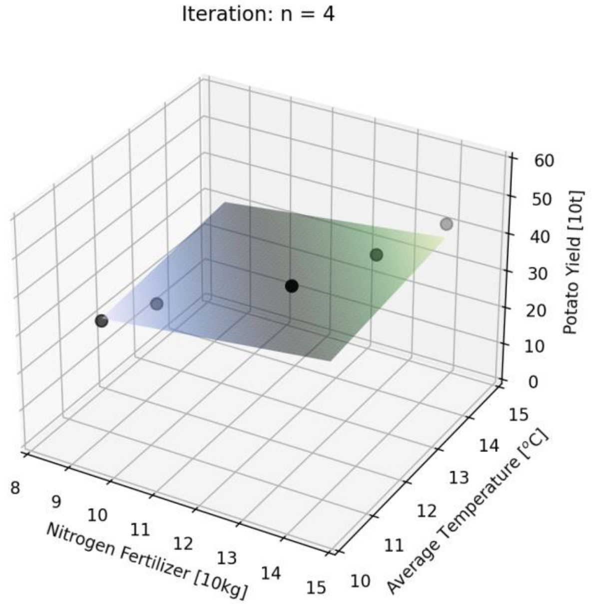
$$w_4^T = [1.101 \quad 1.256 \quad 1.340]$$

- Error function evaluation:

$$F(w_4) = \sum_{i=1}^{5} \left(w_4^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i \right)^2 = 39.661$$

- Gradient evaluation

$$F_w(w_4) = 2 \sum_{i=1}^{N} (w^T x_i - y_i) \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} -20.3 \\ -272.3 \\ -279.7 \end{bmatrix}$$



- Gradient descent step

$$w_5 = w_4 - \alpha F_w(w_4) = \begin{bmatrix} 1.101 \\ 1.256 \\ 1.340 \end{bmatrix} - 0.0005 \begin{bmatrix} -20.3 \\ -272.3 \\ -279.7 \end{bmatrix} = \begin{bmatrix} 1.111 \\ 1.394 \\ 1.480 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_5) - F(w_4)| = |20.747 - 39.661\,| = 18.9 > \epsilon$$

SGD Example: Regression

(Linear Regression)

- Current weight:

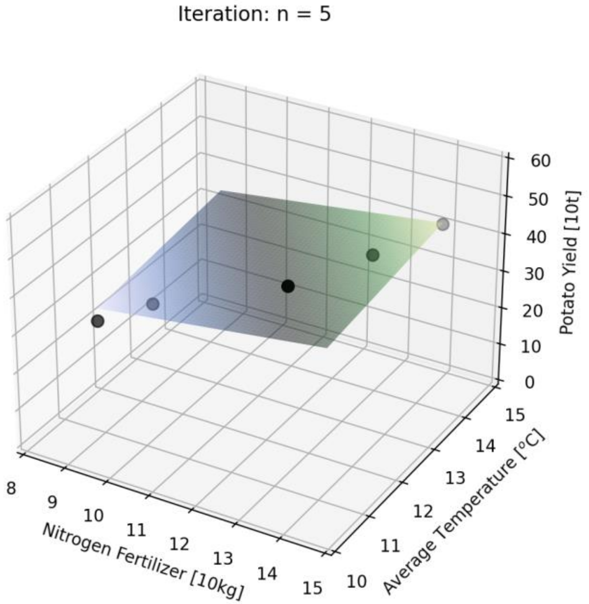
$$w_5^T = [1.111 \quad 1.394 \quad 1.480]$$

- Error function evaluation:

$$F(w_5) = \sum_{i=1}^{5} \left(w_5^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i \right)^2 = 20.747$$

- Gradient evaluation

$$F_w(w_5) = 2 \sum_{i=1}^{N} (w^T x_i - y_i) \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} 13.3 \\ 123.4 \\ 154.7 \end{bmatrix}$$



- Gradient descent step

$$w_6 = w_5 - \alpha F_w(w_5) = \begin{bmatrix} 1.111 \\ 1.394 \\ 1.480 \end{bmatrix} - 0.0005 \begin{bmatrix} 13.3 \\ 123.4 \\ 154.7 \end{bmatrix} = \begin{bmatrix} 1.104 \\ 1.331 \\ 1.403 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_6) - F(w_5)| = |15.830 - 20.747\,| = 4.92 > \epsilon$$

SGD Example: Regression

AIML

(Linear Regression)

- Current weight:

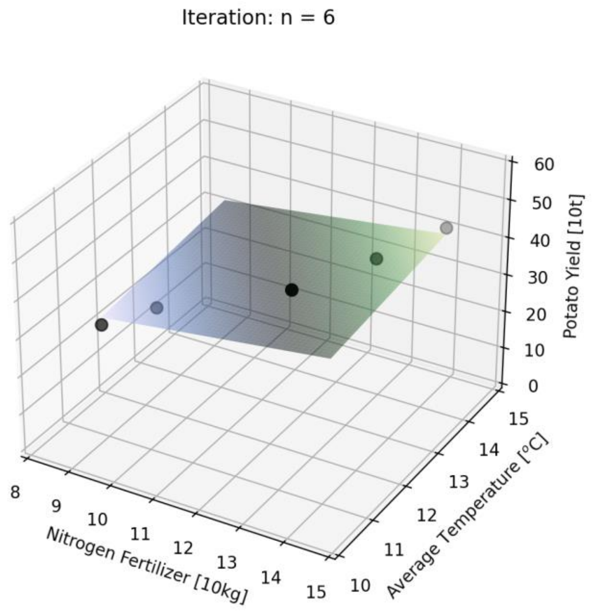
$$w_6^T = [1.104 \quad 1.331 \quad 1.403]$$

- Error function evaluation:

$$F(w_6) = \sum_{i=1}^{5} \left(w_6^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i \right)^2 = 15.830$$

- Gradient evaluation

$$F_w(w_6) = 2 \sum_{i=1}^{N} (w^T x_i - y_i) \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} -3.727 \\ -76.99 \\ -65.40 \end{bmatrix}$$



- Gradient descent step

$$w_7 = w_6 - \alpha F_w(w_6) = \begin{bmatrix} 1.104 \\ 1.331 \\ 1.403 \end{bmatrix} - 0.0005 \begin{bmatrix} -3.727 \\ -76.99 \\ -65.40 \end{bmatrix} = \begin{bmatrix} 1.106 \\ 1.369 \\ 1.436 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_7) - F(w_6)| = |14.505 - 15.830\ | = 1.33 > \epsilon$$

AIML

SGD Example: Regression

(Linear Regression)

- Current weight:

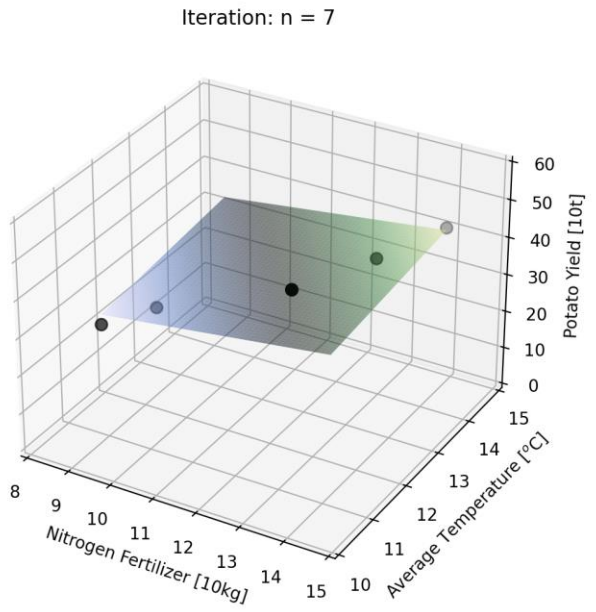
$$w_7^T = [1.106 \quad 1.369 \quad 1.436]$$

- Error function evaluation:

$$F(w_7) = \sum_{i=1}^{5} \left(w_7^T \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} - y_i \right)^2 = 14.505$$

- Gradient evaluation

$$F_w(w_7) = 2 \sum_{i=1}^{N} (w^T x_i - y_i) \begin{bmatrix} 1 \\ x_i^1 \\ x_i^2 \end{bmatrix} = \begin{bmatrix} 4.891 \\ 24.57 \\ 46.04 \end{bmatrix}$$



- Gradient descent step

$$w_8 = w_7 - \alpha F_w(w_7) = \begin{bmatrix} 1.106 \\ 1.369 \\ 1.436 \end{bmatrix} - 0.0005 \begin{bmatrix} 4.891 \\ 24.57 \\ 46.04 \end{bmatrix} = \begin{bmatrix} 1.104 \\ 1.357 \\ 1.413 \end{bmatrix}$$

- Convergence test

$$\Delta F = |F(w_8) - F(w_7)| = |14.103 - 14.505\,| = 0.40 < \epsilon$$

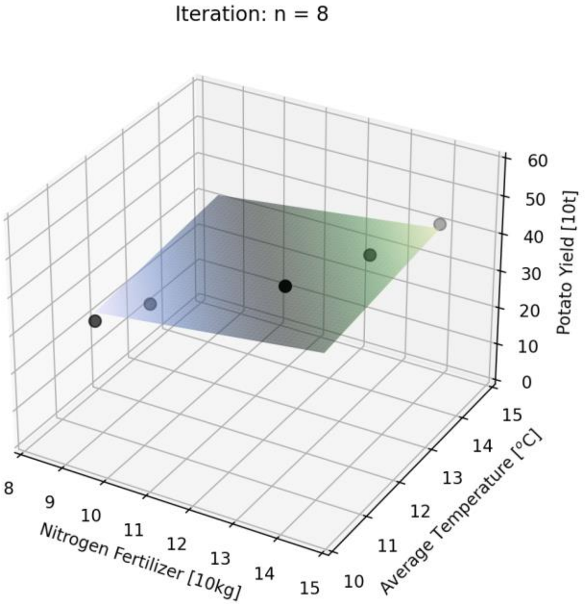
SGD Example: Regression

(Linear Regression)

- Optimal weight:

$$w^\star = w_8 = \begin{bmatrix} 1.104 \\ 1.357 \\ 1.413 \end{bmatrix}$$

- Suppose we have a new farm that used 10 kg of nitrogen fertilizer in a season with average temperature of 15 ^{o}C. How many potatoes would you expect to yield on this farm?



- Optimized model

$$f(w,x) = w^T x = \begin{bmatrix} 1.104 & 1.357 & 1.413 \end{bmatrix} \begin{bmatrix} 1 \\ 10 \\ 15 \end{bmatrix} = 35.9\ t$$

AIML

AIML

Regression: analysis

#Code

- For linear regression, error function is convex so SGD with correct parameters, can converge on the globally optimal solution eventually (N.B. do not need to use SGD, there is a straightforward analytical solution)
- Unlike linear regression, more complex regression can often be made to fit N training data points **exactly** (zero training error), but this model will not **generalize well** to another random sample of data from the same circumstance
- A **trade off** between **model complexity** and **test error**: common feature of most machine learning models: want a model which is **as simple as possible** but **no simpler** (Occam's razor)

To recap

- We reviewed the sequential gradient descent algorithm in the context of regression

- **Next lecture**: Classification in ML