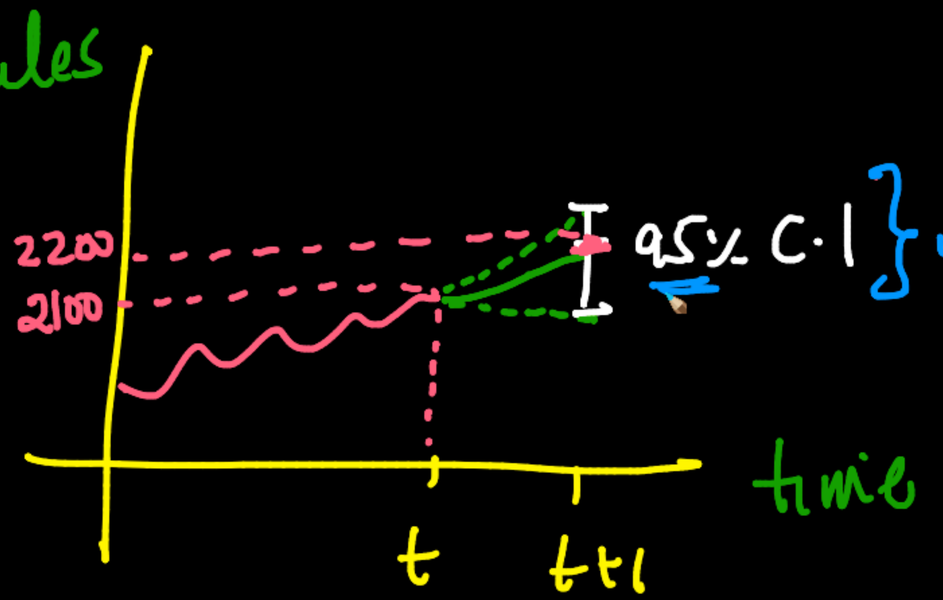
Agenda:

1. Note about NN-module ✓
2. C.I's for time-series ✓
3. WCSS, Silhoutte scores, wards method
   - 3a. Min/Max/Avg linkage outliers
4. Isolation forest
5. Local & global minima in tSNE vs PCA
6. Apriori: antecedent, Consequent
7. clusterability ✓
8. Post read: FB Prophet

C.I for time series (TSF)

ARIMA
SARIMA
SARIMAX



{ 95% C.I on y_{t+1} instead of $\hat{y}_{t+1}$

[2105, 2406]

2200

(Q) How can we predict 95% C.I on y_{t+1} ?

$y_{t+1} \sim$ Some dist^b (params)

from pools-classes { (Let) $y_{t+1} \sim \text{Normal}(\mu, \sigma)$

95% C.I on $y_{t+1} = [\mu - 1.96\sigma, \mu + 1.96\sigma]$

SARIMA : classical TSF

(p, q, d, s, P, D, Q)

$\sum \alpha_{it} \sum \beta_{it} \sum$ { Linear combination of weighted values

- ✓ Linear Regr
- DT-regr
- SVM-regr
- RF-regr
- GBDT-regr

Proof: Maximum likelihood estimation for simple linear regression

Index: The Book of Statistical Proofs ▷ Statistical Models ▷ Univariate normal data ▷ Simple linear regression ▷ Maximum likelihood estimation

Theorem: Given a simple linear regression model with independent observations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \; i = 1, \ldots, n \,, \tag{1}$$

the maximum likelihood estimates of β_0, β_1 and σ^2 are given by

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{s_{xy}}{s_x^2} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the sample means, s_x^2 is the sample variance of x and s_{xy} is the sample covariance between x and y.

Proof: With the probab[illegible] probability under

Proof: Maximum likelihood estimation for simple linear regression

Index: The Book of Statistical Proofs ▷ Statistical Models ▷ Univariate normal data ▷ Simple linear regression ▷ Maximum likelihood estimation

Theorem: Given a simple linear regression model with independent observations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \; i = 1, \ldots, n \,, \tag{1}$$

the maximum likelihood estimates of β_0, β_1 and σ^2 are given by

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{s_{xy}}{s_x^2} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the sample means, s_x^2 is the sample variance of x and s_{xy} is the sample covariance between x and y.

Proof: With the probab[illegible] probability under

Proof: Maximum likelihood estimation for simple linear regression

Index: The Book of Statistical Proofs ▷ Statistical Models ▷ Univariate normal data ▷ Simple linear regression ▷ Maximum likelihood estimation

Theorem: Given a simple linear regression model with independent observations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \; i = 1, \ldots, n \,, \tag{1}$$

the maximum likelihood estimates of β_0, β_1 and σ^2 are given by

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{s_{xy}}{s_x^2} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{2}$$

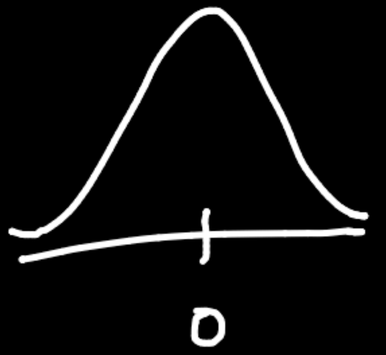
where $\bar{x}$ and $\bar{y}$ are the sample means, s_x^2 is the sample variance of x and s_{xy} is the sample covariance between x and y.

Proof: With the probab[illegible] probability under

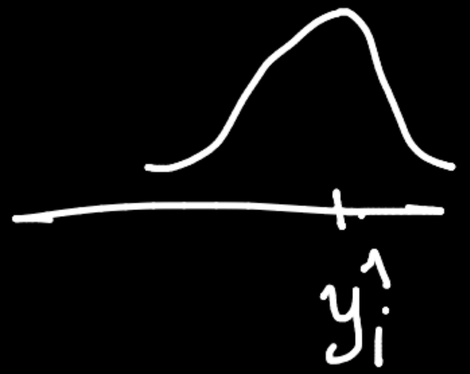
$$y_i = \underbrace{\beta_0 + \beta_1 x_i}_{\hat{y}_i} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

$$y_i = \hat{y}_i + \epsilon_i$$



$$\checkmark \left[y_i \sim N\left(\hat{y}_i, \boxed{\sigma^2}\right) \right]$$



simple linear regression

Theorem: Given a simple linear regression model with independent observations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \; i = 1, \ldots, n \,, \tag{1}$$

the maximum likelihood estimates of β_0, β_1 and σ^2 are given by

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{s_{xy}}{s_x^2} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{2}$$

MSE

where $\bar{x}$ and $\bar{y}$ are the sample means, s_x^2 is the sample variance of x and s_{xy} is the sample covariance between x and y.

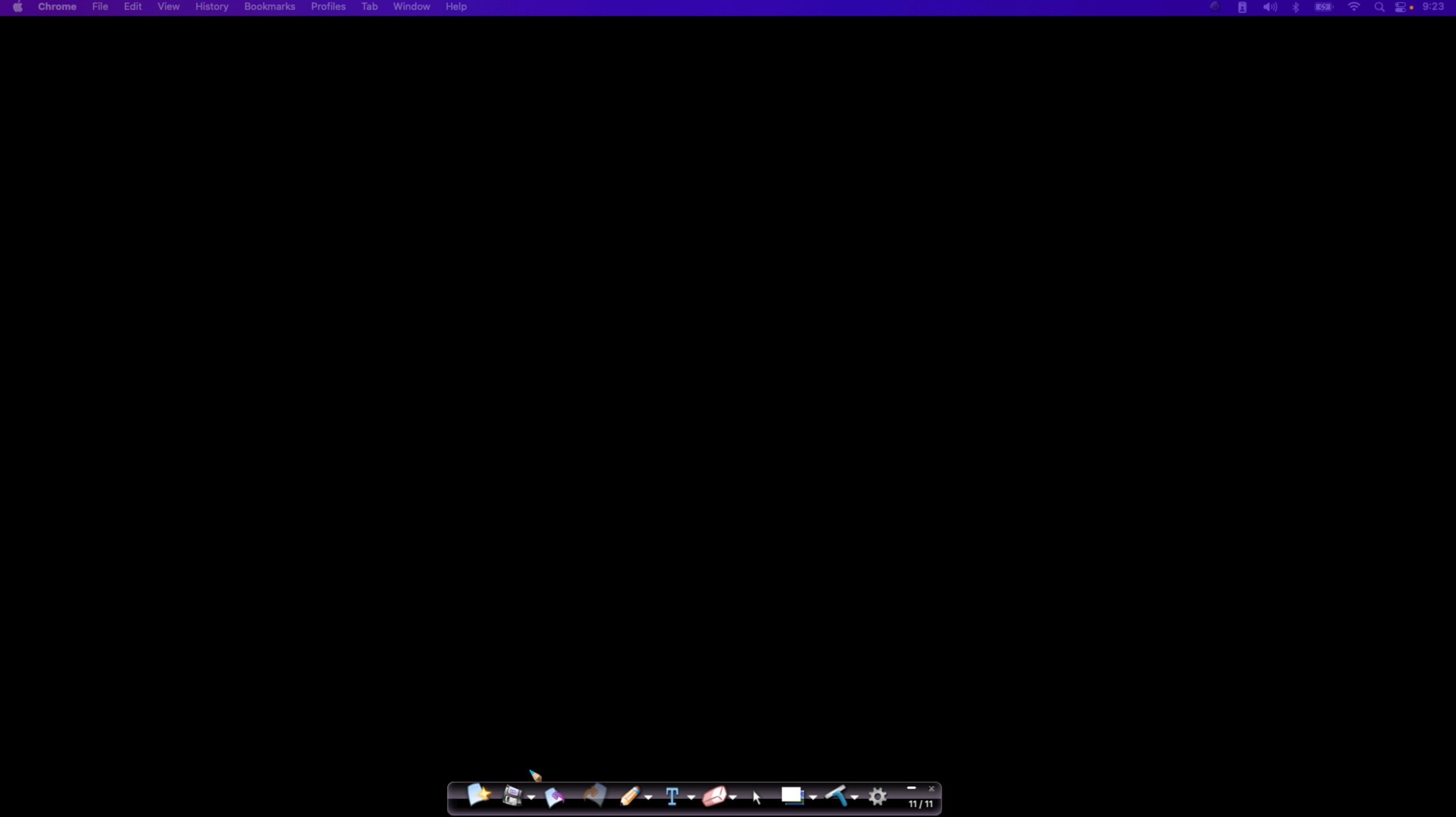
Proof: With the probability density function of the normal distribution and probability under independence, the linear regression equation (1) implies the following likelihood function

{ Linear Regr : $y_i \sim N(\hat{y}_i, MSE = \sigma^2)$

{ Classical TSF : $y_{t+1} \sim N(\hat{y}_{t+1}, \tilde{\sigma}^2 = \text{MSE over all past timestamps})$

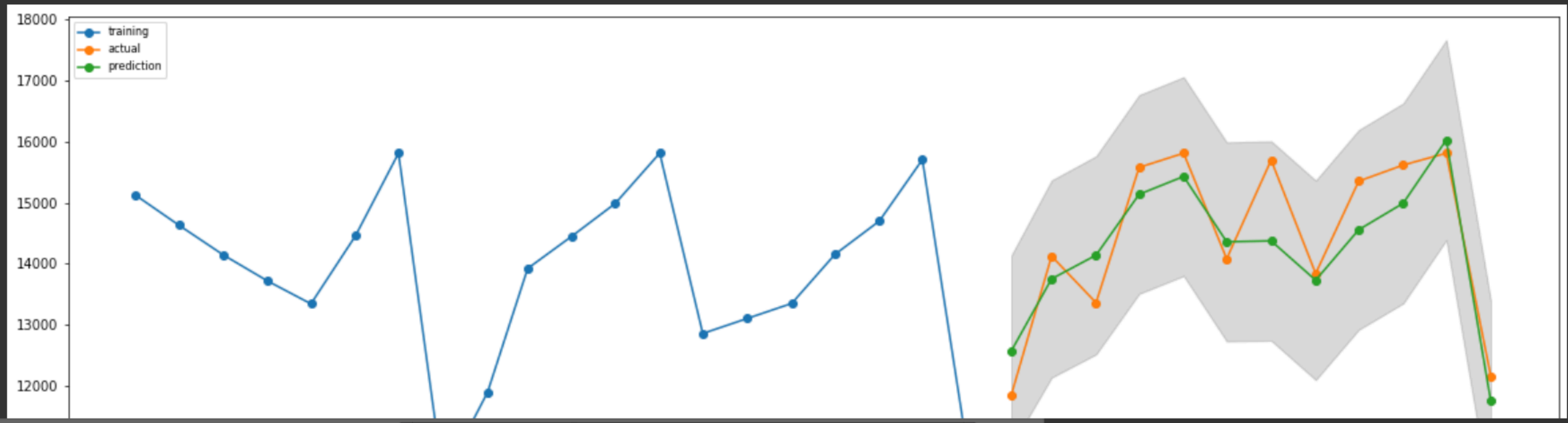
$\Updownarrow$ asy. C.I on y_{t+1}

$\mu \pm 1.96\sigma$


Chrome File Edit View History Bookmarks Profiles Tab Window Help
9:23
11 / 11

Let's use the returned upper and lower bound values of each forecast using the function `conf_int` to create separate series for lower and upper bounds for visualization purpose.

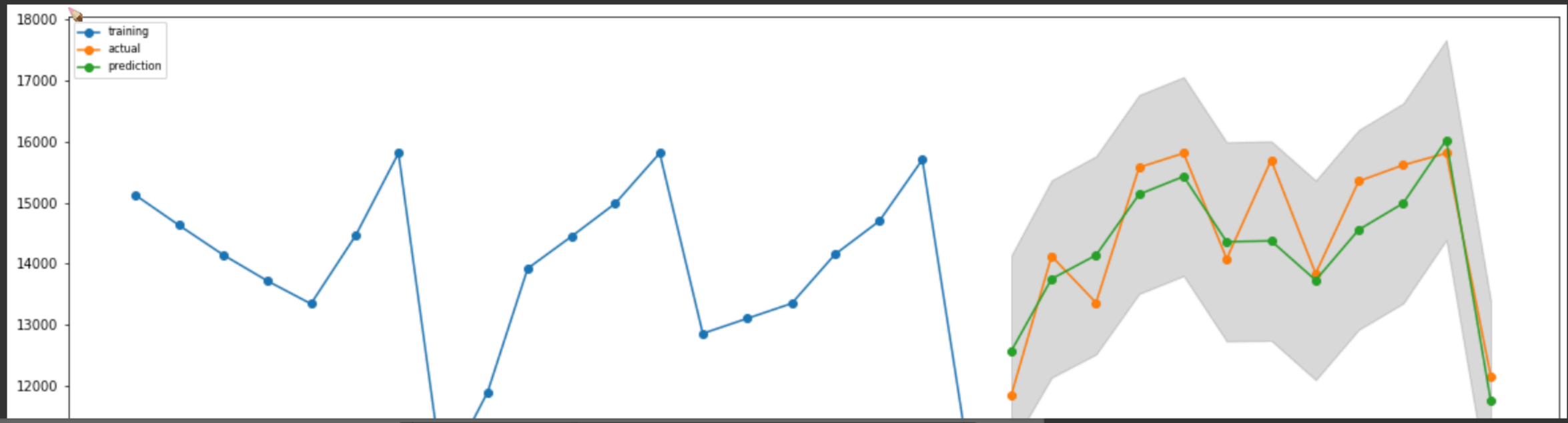
```
test_x[['lower', 'upper']] = model.get_forecast(steps=12).conf_int(0.05).values
plt.plot(train_x['Sales'][-20:], '-o', label='training')
plt.plot(test_x['Sales'], '-o', label='actual')
plt.plot(test_x['pred'], '-o', label='prediction')
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



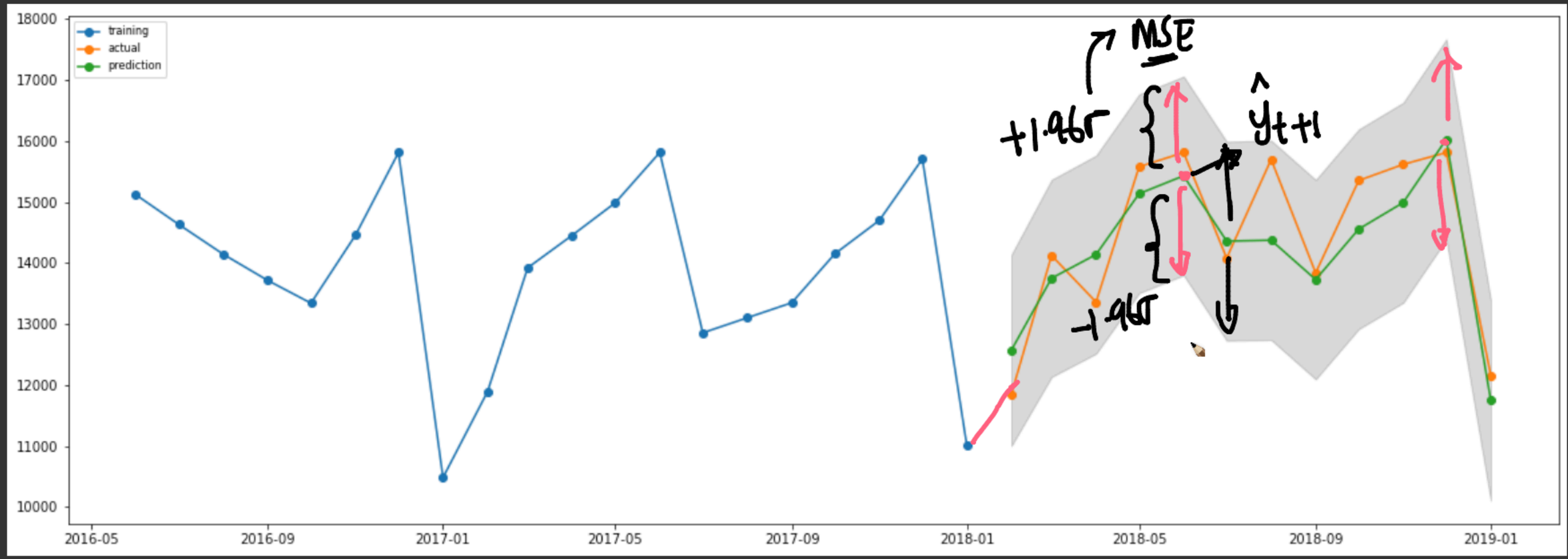
+ Code + Text Cannot save changes Connect

Let's use the returned upper and lower bound values of each forecast using the function `conf_int` to create separate series for lower and upper bounds for visualization purpose.

```
test_x[['lower', 'upper']] = model.get_forecast(steps=12).conf_int(0.05).values
plt.plot(train_x['Sales'][-20:], '-o', label='training')
plt.plot(test_x['Sales'], '-o', label='actual')
plt.plot(test_x['pred'], '-o', label='prediction')
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



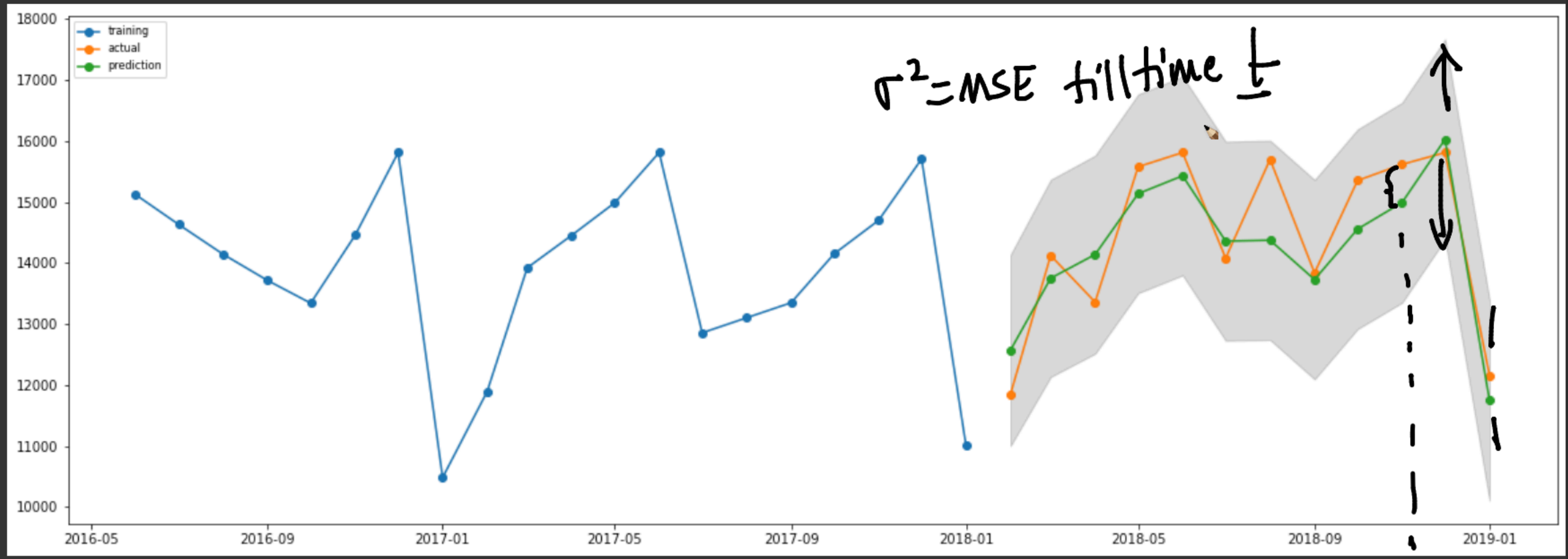
```
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



In the above plot, we can observe

- The actual observed values of test set

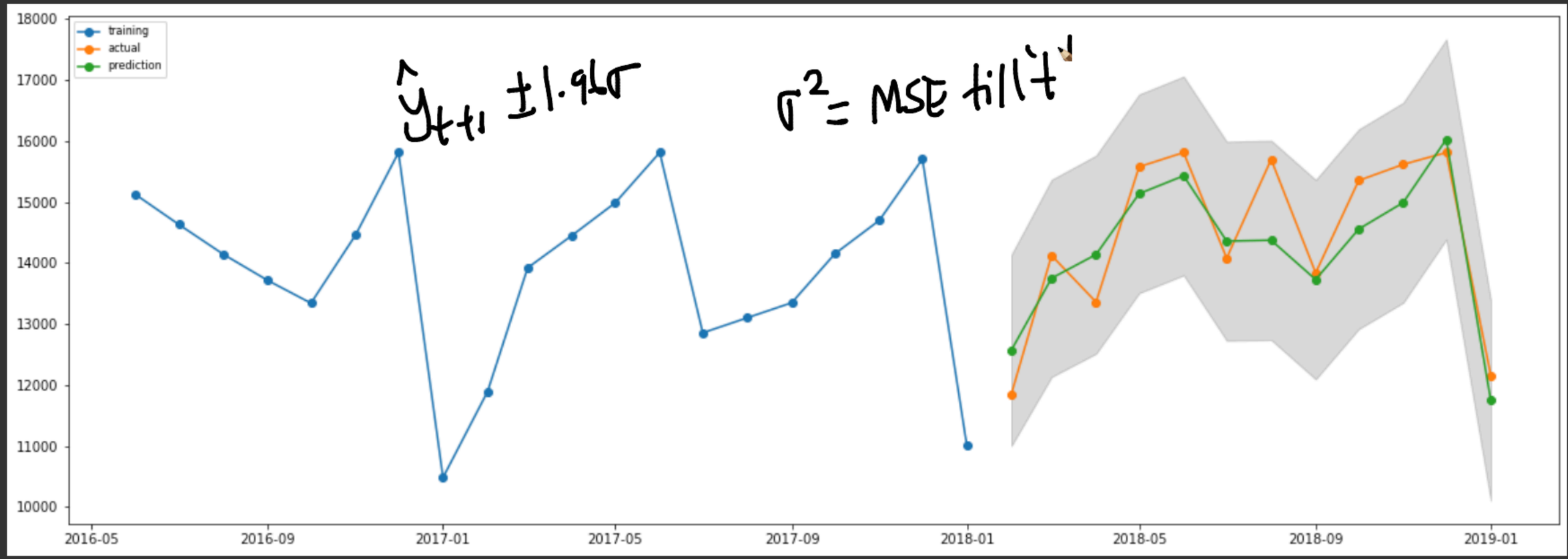
```
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



In the above plot, we can observe

- The actual observed values of test set

```
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



In the above plot, we can observe

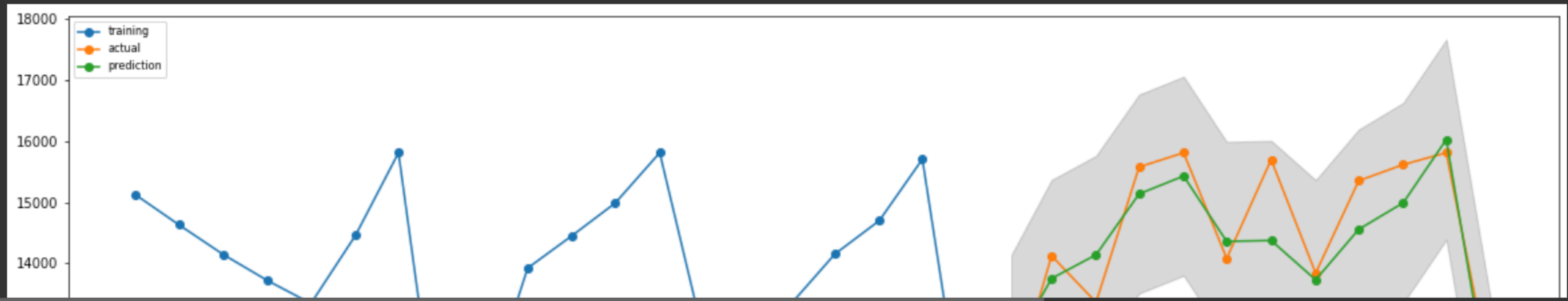
- The actual observed values of test set

It means that the time series model will estimate the upper(n) and lower(m) bound of values around the forecast, where there is only a 5% chance that the real value will **not be** in that range.

That is, 95% that our forecast will fall within the range `(m,n)`.

Let's use the returned upper and lower bound values of each forecast using the function `conf_int` to create separate series for lower and upper bounds for visualization purpose.

```
test_x[['lower', 'upper']] = model.get_forecast(steps=12).conf_int(0.05).values
plt.plot(train_x['Sales'][-20:], '-o', label='training')
plt.plot(test_x['Sales'], '-o', label='actual')
plt.plot(test_x['pred'], '-o', label='prediction')
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



Proof: Maximum likelihood estimation for simple linear regression

Index: The Book of Statistical Proofs ▷ Statistical Models ▷ Univariate normal data ▷ Simple linear regression ▷ Maximum likelihood estimation

Theorem: Given a simple linear regression model with independent observations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \; i = 1, \ldots, n \; , \tag{1}$$

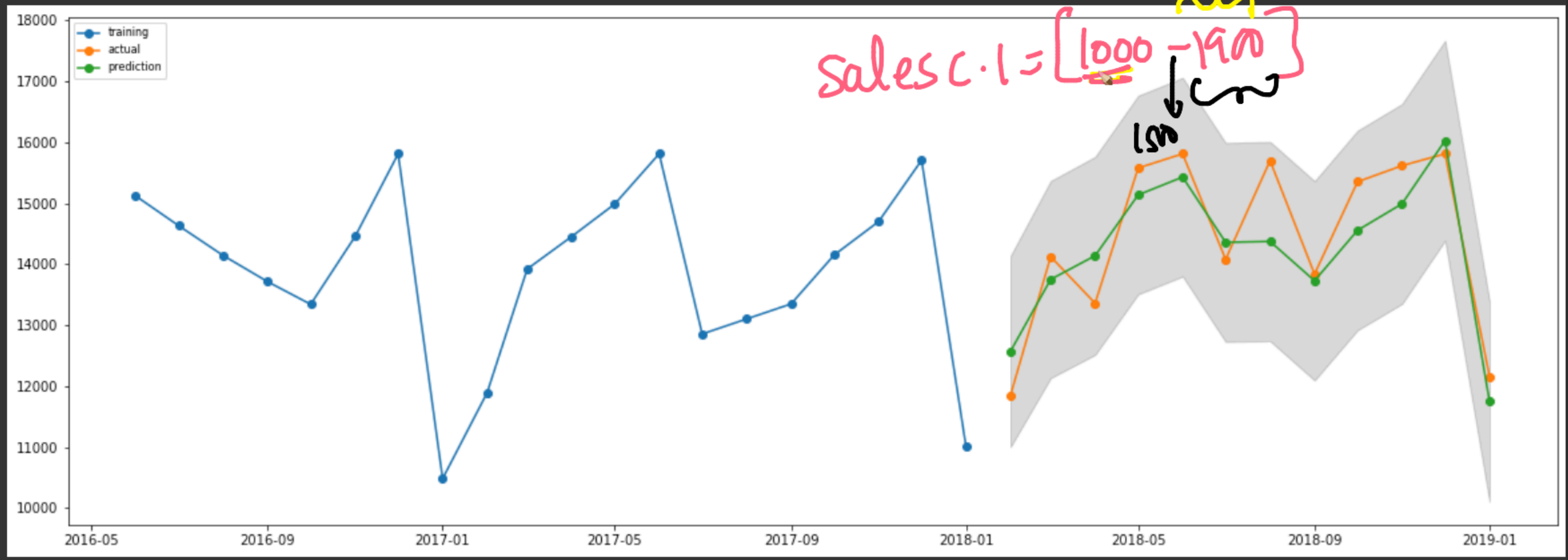
the maximum likelihood estimates of β_0, β_1 and σ^2 are given by

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{s_{xy}}{s_x^2} \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned} \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the sample means, s_x^2 is the sample variance of x and s_{xy} is the sample covariance between x and y.

Proof: With the probab probability under

```
plt.plot(test_x['Sales'], '-o', label='actual')
plt.plot(test_x['pred'], '-o', label='prediction')
plt.fill_between(test_x.index, test_x['lower'], test_x['upper'],
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```



What is K-means doing?

K-means is based on the following intuition: points in a cluster should be close to their cluster mean.

Suppose we have n observations $x_1, \dots, x_n$ we want to cluster in to K subgroups. Let $S_1, \dots, S_K$ be sets containing the indices of observations in each cluster. E.g. if $n = 5$ and $K = 2$ then we might have $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$. Let $\mu_1, \cdots \mu_K$ be the mean of each cluster.

We can measure how close each point is to its cluster mean by summing the squared distance from each point to its cluster center i.e. $\sum_{i \in S_j} ||x_i - \mu_k||^2$. Then we can define the *within cluster sum of squares* (WCSS) as follows

$$WCSS = \sum_{k=1}^{K} \sum_{i \in S_j} ||x_i - \mu_k||^2$$

Note that WCSSS depends only on the cluster assignments $S_1, \dots, S_K$. WCSS is a measure of how well the points cluster. This suggests we select the cluster assignments that minimizes WCSS.

It turns out that finding the cluster assignments $S_1, \dots, S_K$ to minimize WCSS is NP complete meaning we have no hope of actually solving it exactly for anything but small problems. **K-means is a way of approximately solving the WCSS minimization problem**.

In the paragraphs above we motivated minimizing WCSS though a heuristic argument. There is actually a statistical interpretation to minimizing WCSS. Suppose each data point is generated independently from one of K Gaussian distributions where each of the K Gaussian distributions has a different mean but identity covariance matrix. It then turns out that finding the maximum likelihood solution to this problem is exactly minimizing the WCSS. This is a spec

the indices of observations in each cluster. E.g. if $n = 5$ and $K = 2$ then we might have $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$. Let $\mu_1, \cdots \mu_K$ be the mean of each cluster.

We can measure how close each point is to its cluster mean by summing the squared distance from each point to its cluster center i.e. $\sum_{i \in S_j} ||x_i - \mu_k||^2$. Then we can define the *within cluster sum of squares* (WCSS) as follows

Yes ↑ ↓ No

$$WCSS = \sum_{k=1}^{K} \sum_{i \in S_j} ||x_i - \mu_k||^2$$

Note that WCSSS depends only on the cluster assignments $S_1, \dots, S_K$. WCSS is a measure of how well the points cluster. This suggests we select the cluster assignments that minimizes WCSS.

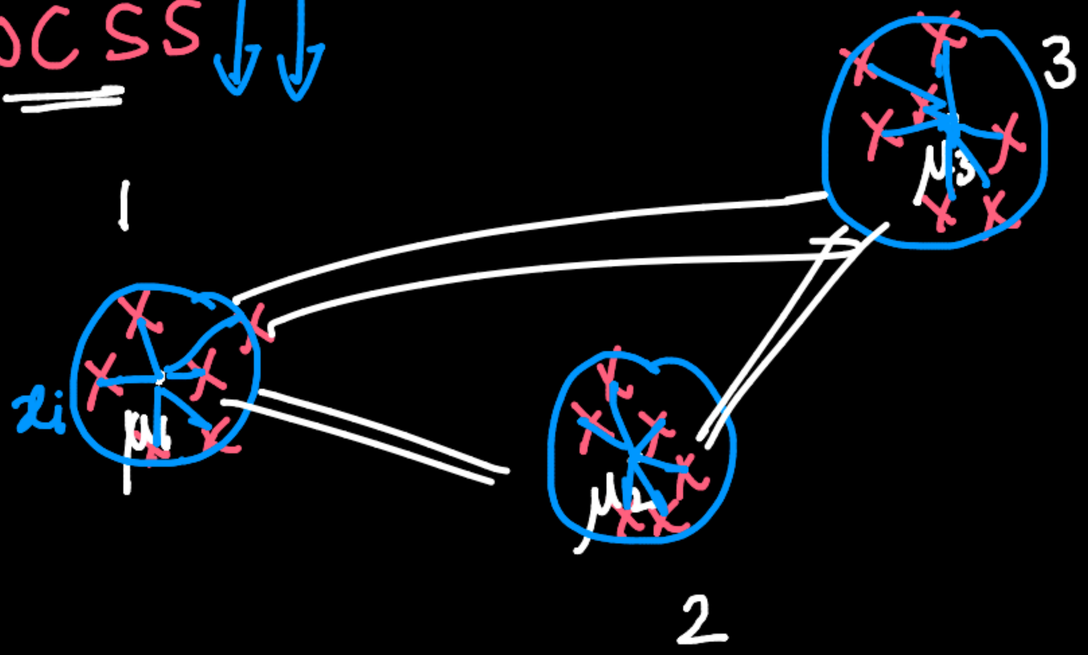
It turns out that finding the cluster assignments $S_1, \dots, S_K$ to minimize WCSS is NP complete meaning we have no hope of actually solving it exactly for anything but small problems. **K-means is a way of approximately solving the WCSS minimization problem**.

In the paragraphs above we motivated minimizing WCSS though a heuristic argument. There is actually a statistical interpretation to minimizing WCSS. Suppose each data point is generated independently from one of K Gaussian distributions where each of the K Gaussian distributions has a different mean but identity covariance matrix. It then turns out that finding the maximum likelihood solution to this problem is exactly minimizing the WCSS. This is a special case of the more general gaussian mixture model.

What happens when you change K?

See this shiny app to see what happens when we change K.

good Clustering : WCSS ↓↓



WC SS

✓ disadv: inter-clust dist ✗

$$\left[\min \sum_{k=1}^{K} \sum_{i \in S_k} \|x_i - \mu_k\|^2 \right] \checkmark$$

the indices of observations in each cluster. E.g. if $n = 5$ and $K = 2$ then we might have $S_1 = \{1, 3, 5\}$ and $S_2 = \{2, 4\}$. Let $\mu_1, \cdots \mu_K$ be the mean of each cluster.

We can measure how close each point is to its cluster mean by summing the squared distance from each point to its cluster center i.e. $\sum_{i \in S_j} ||x_i - \mu_k||^2$. Then we can define the *within cluster sum of squares* (WCSS) as follows

$$WCSS = \sum_{k=1}^{K} \sum_{i \in S_j} ||x_i - \mu_k||^2$$

Note that WCSSS depends only on the cluster assignments $S_1, \dots, S_K$. WCSS is a measure of how well the points cluster. This suggests we select the cluster assignments that minimizes WCSS.

It turns out that finding the cluster assignments $S_1, \dots, S_K$ to minimize WCSS is NP complete meaning we have no hope of actually solving it exactly for anything but small problems. **K-means is a way of approximately solving the WCSS minimization problem**.

In the paragraphs above we motivated minimizing WCSS though a heuristic argument. There is actually a statistical interpretation to minimizing WCSS. Suppose each data point is generated independently from one of K Gaussian distributions where each of the K Gaussian distributions has a different mean but identity covariance matrix. It then turns out that finding the maximum likelihood solution to this problem is exactly minimizing the WCSS. This is a special case of the more general gaussian mixture model.

What happens when you change K?

See this shiny app to see what happens when we change K.

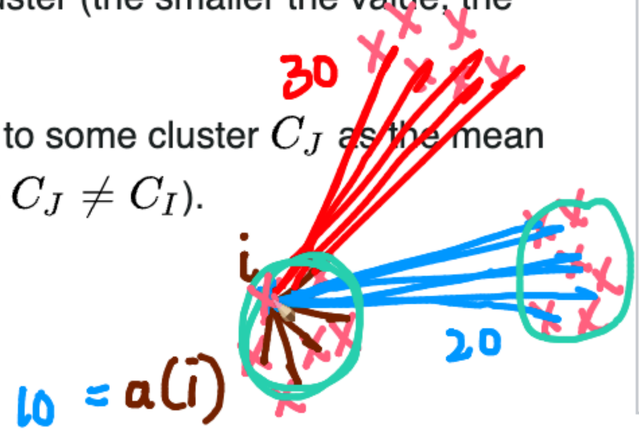
Українська
粵語
中文
Edit links

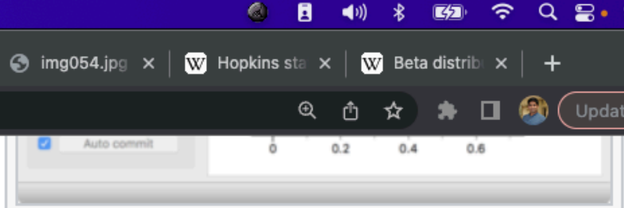
as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

We then define the mean dissimilarity of point i to some cluster C_J as the mean of the distance from i to all points in C_J (where $C_J \neq C_I$).

For each data point $i \in C_I$, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$





A plot showing silhouette scores from three types of animals from the Zoo dataset as rendered by Orange data mining suite. At the bottom of the plot, silhouette identifies dolphin and porpoise as outliers in the group of mammals.

to be the *smallest* (hence the min operator in the formula) mean distance of i to all points in any other cluster (i.e., in any cluster of which i is not a member). The cluster with this smallest mean dissimilarity is said to be the "neighboring cluster" of i because it is the next best fit cluster for point i.

We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

b(i)=20

0.001 20

20

and

$$s(i) = 0, \text{ if } |C_I| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i) & \end{cases}$$

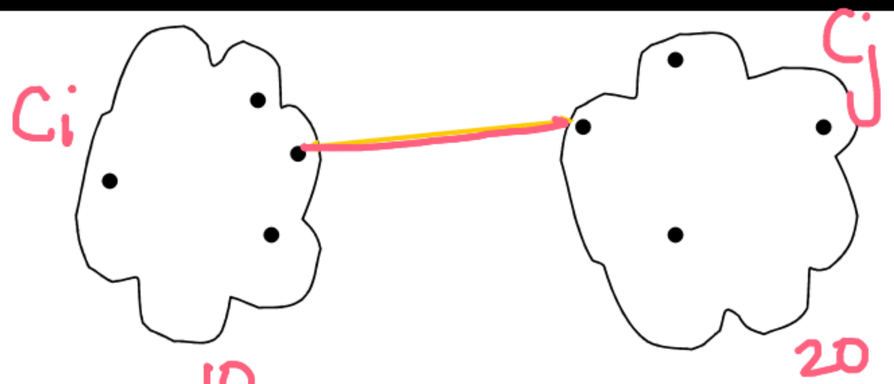
ideal s(i) → 1 ✓

worst s(i) → 0

(?)

s(i) = -1

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

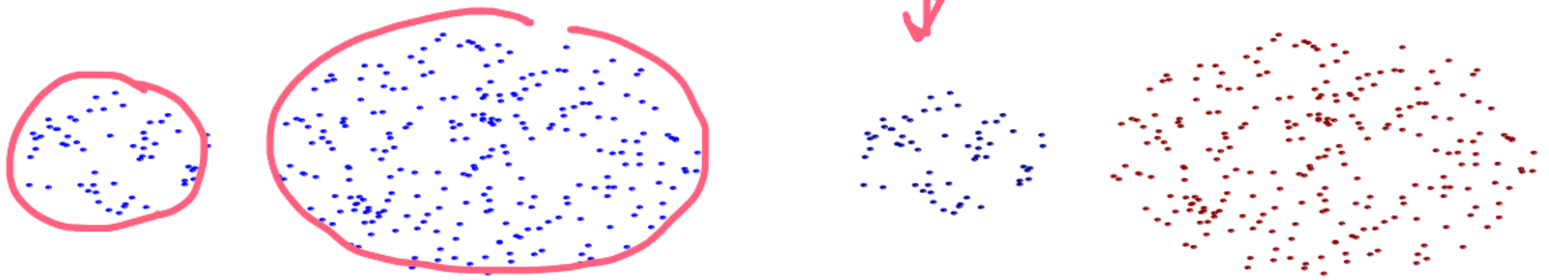
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

Single Link – Complete Link

- Another way to view the processing of the hierarchical algorithm is that we create links between their elements in order of increasing distance
  - The MIN – Single Link, will merge two clusters when a single pair of elements is linked
  - The MAX – Complete Linkage will merge two clusters when all pairs of elements have been linked.

Strength of MIN

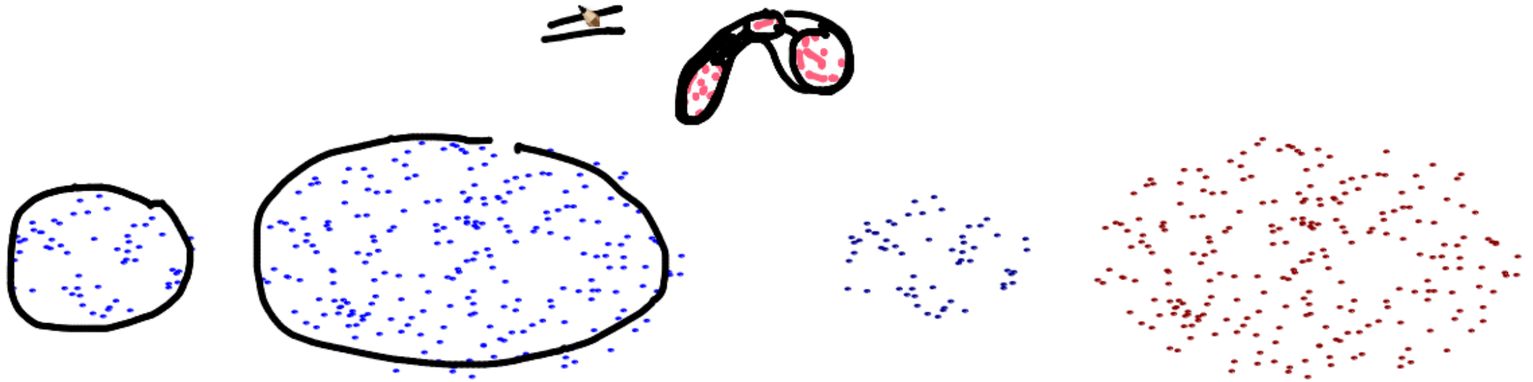


Original Points

Two Clusters

- Can handle non-elliptical shapes

Strength of MIN



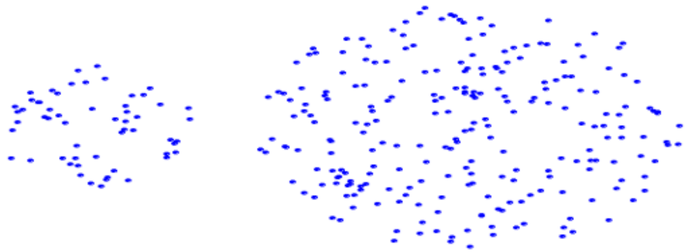
Original Points

Two Clusters

- Can handle non-elliptical shapes

Globular

Strength of MIN

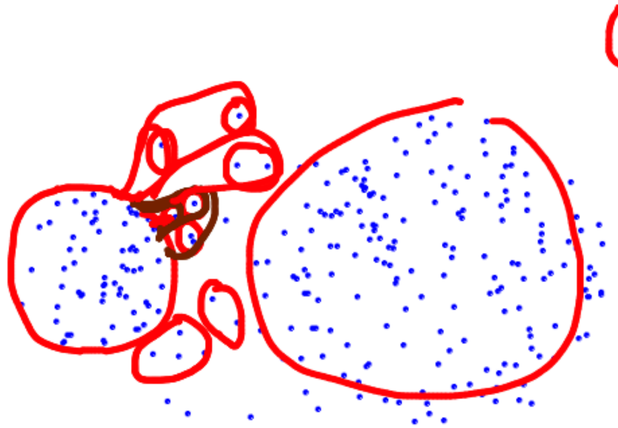


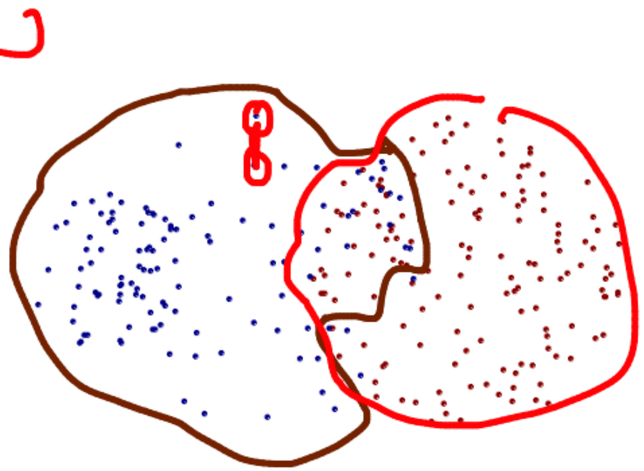
Original Points

Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



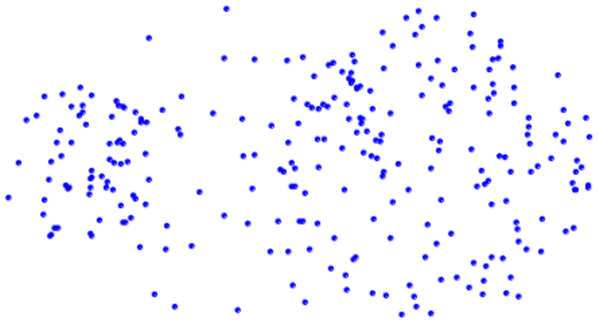


Original Points

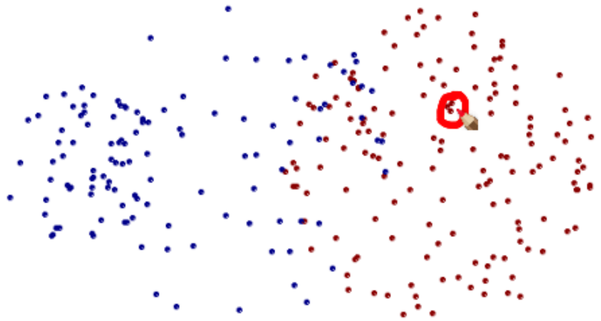
Two Clusters

- Sensitive to noise and outliers

Limitations of MIN



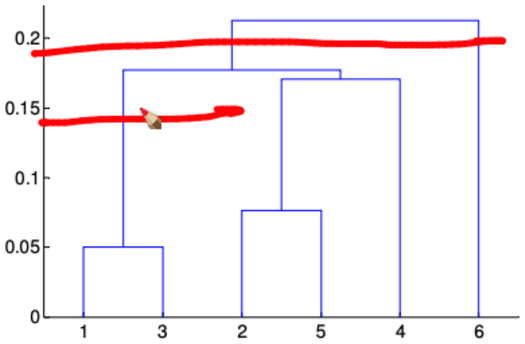
Original Points

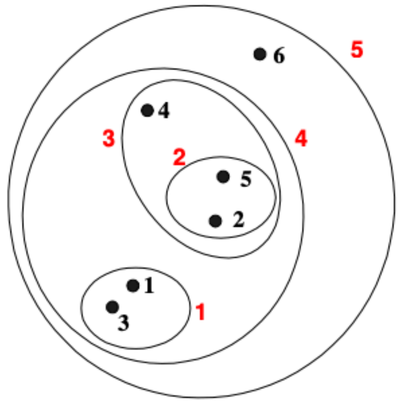


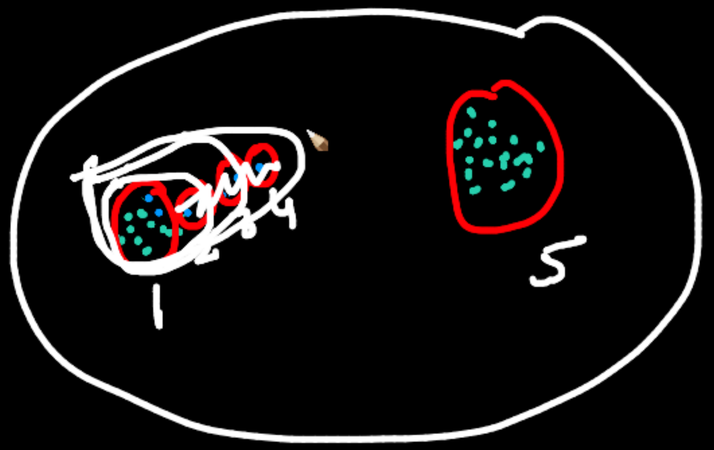
Two Clusters

- Sensitive to noise and outliers

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



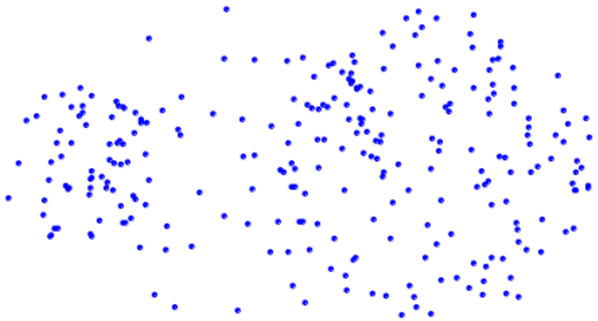




min / single

2 cluster

Limitations of MIN

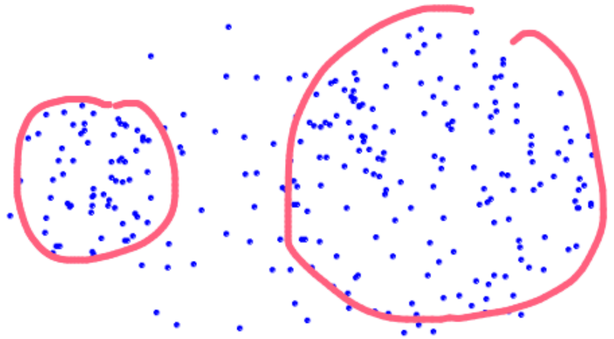


Original Points

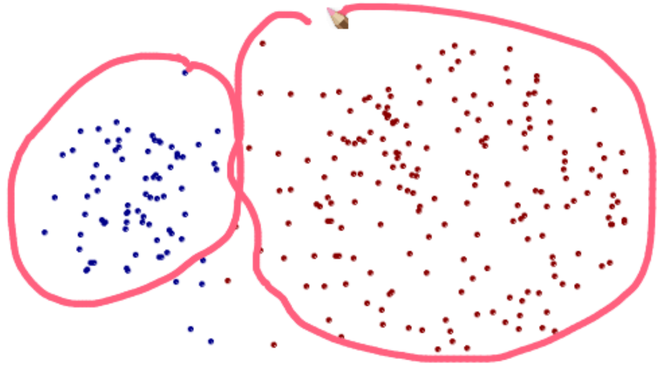
Two Clusters

- Sensitive to noise and outliers

Strength of MAX



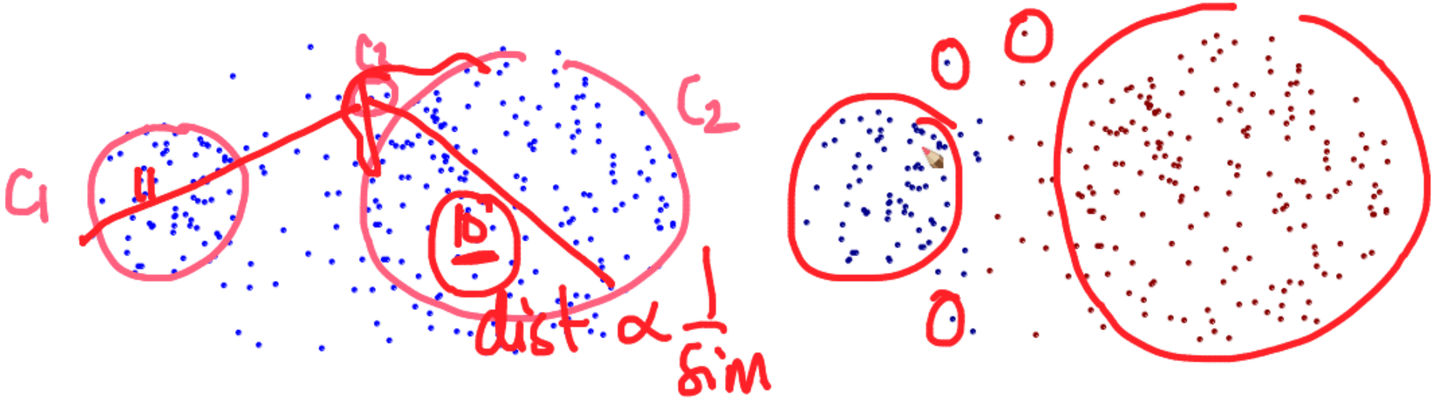
Original Points



Two Clusters

- Less susceptible to noise and outliers

Strength of MAX

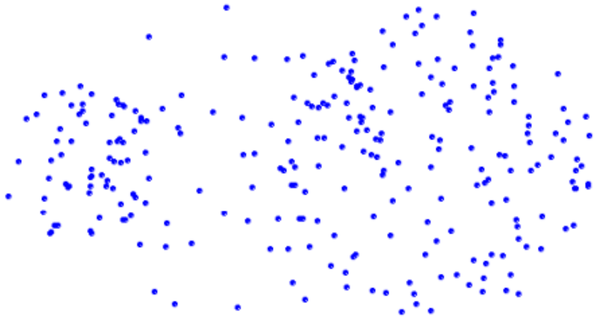


Original Points

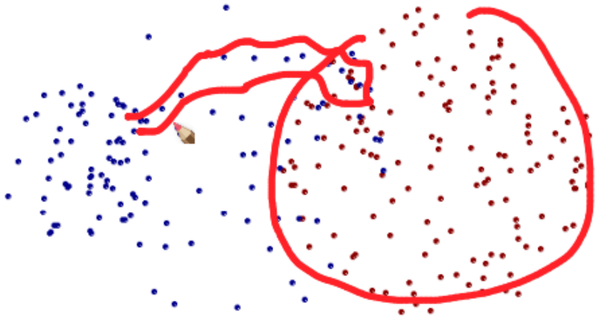
Two Clusters

- Less susceptible to noise and outliers

Limitations of MIN



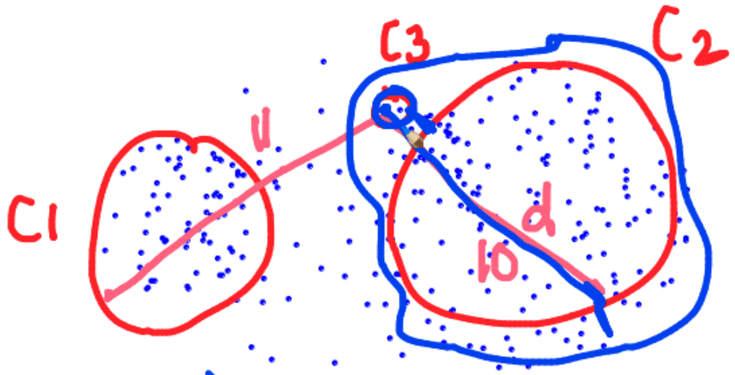
Original Points



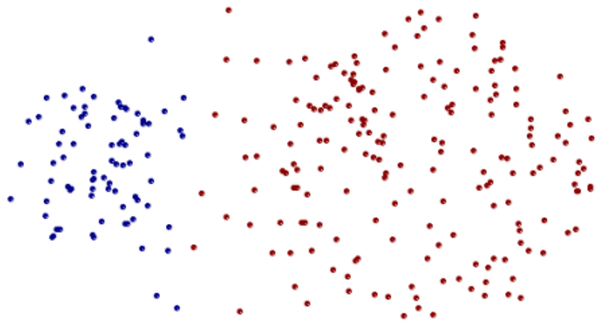
Two Clusters

- Sensitive to noise and outliers

Strength of MAX



Original Points



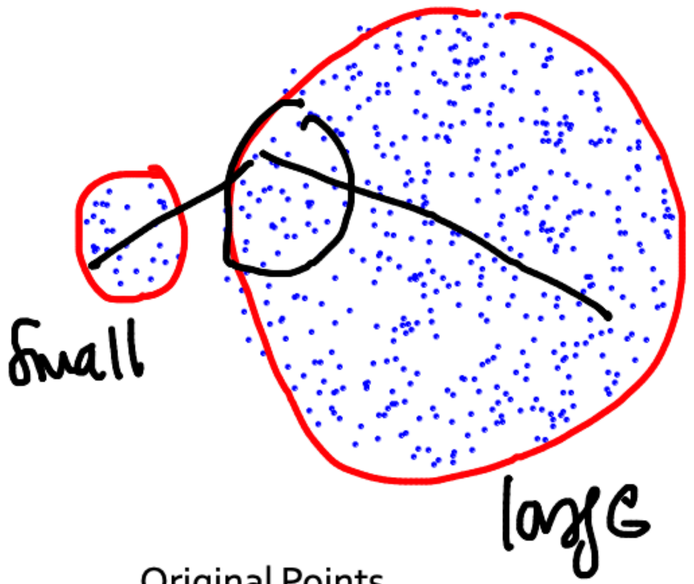
Two Clusters

$\max(1/10, 1/11)$

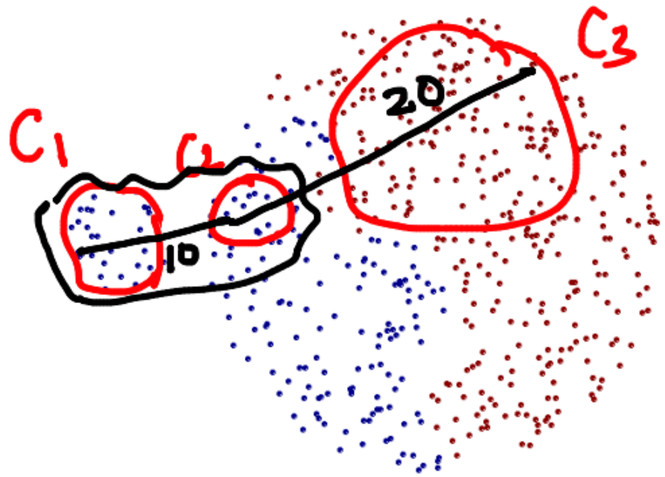
$dist \propto \frac{1}{sim}$

- Less susceptible to noise and outliers

Limitations of MAX

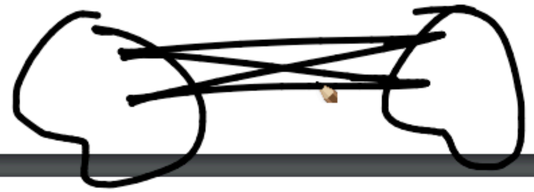


Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters



Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

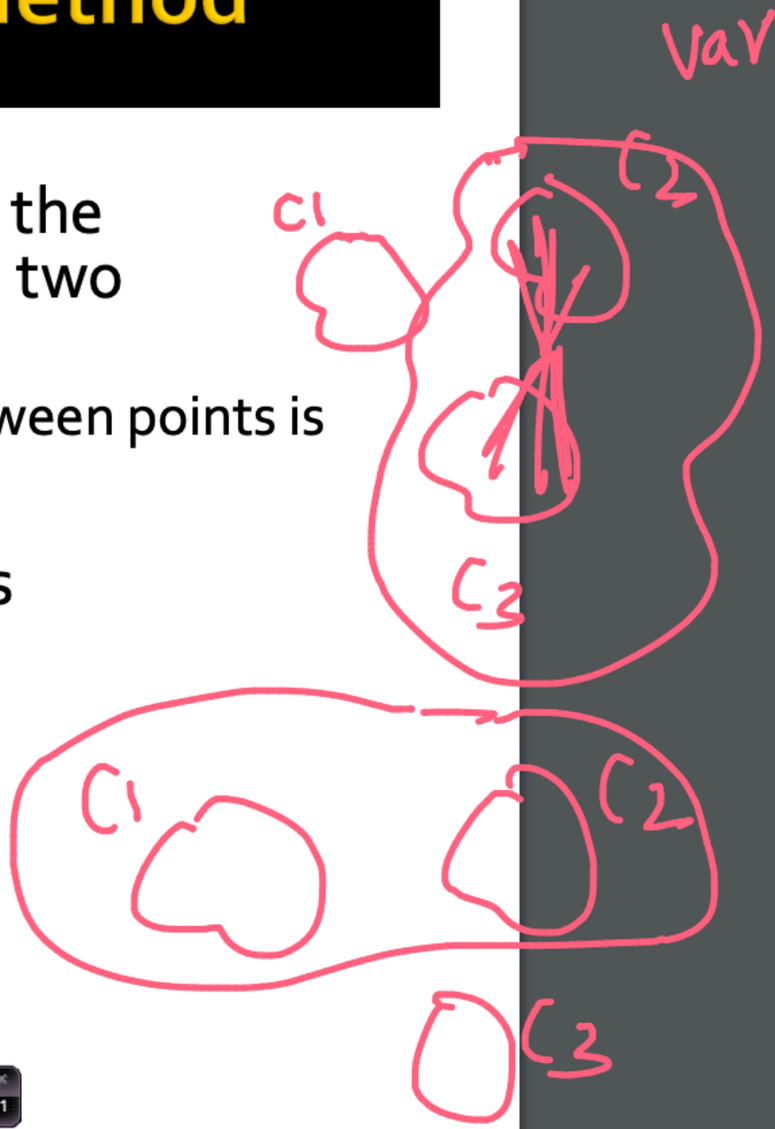
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24					

Cluster Similarity: Ward's Method

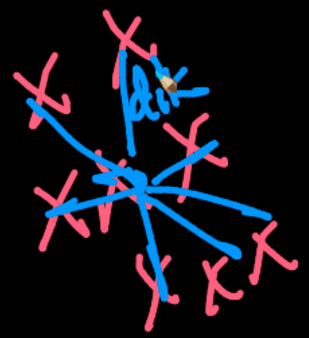
- Similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means



WCSS

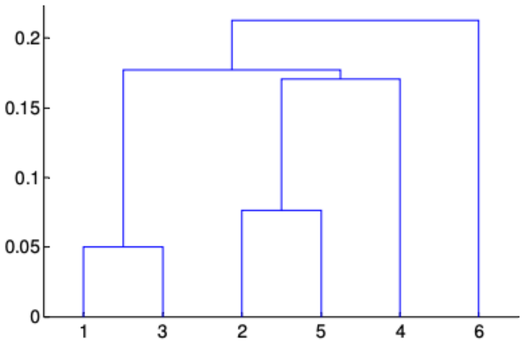
dik

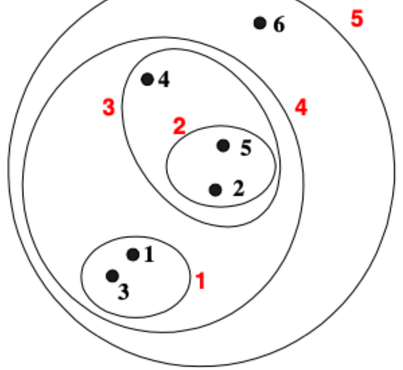
$$SSE = \sum_{K} \sum_{x_i \in C_K} \|(x_i - \mu_K)\|^2$$



as a hierarchical tree

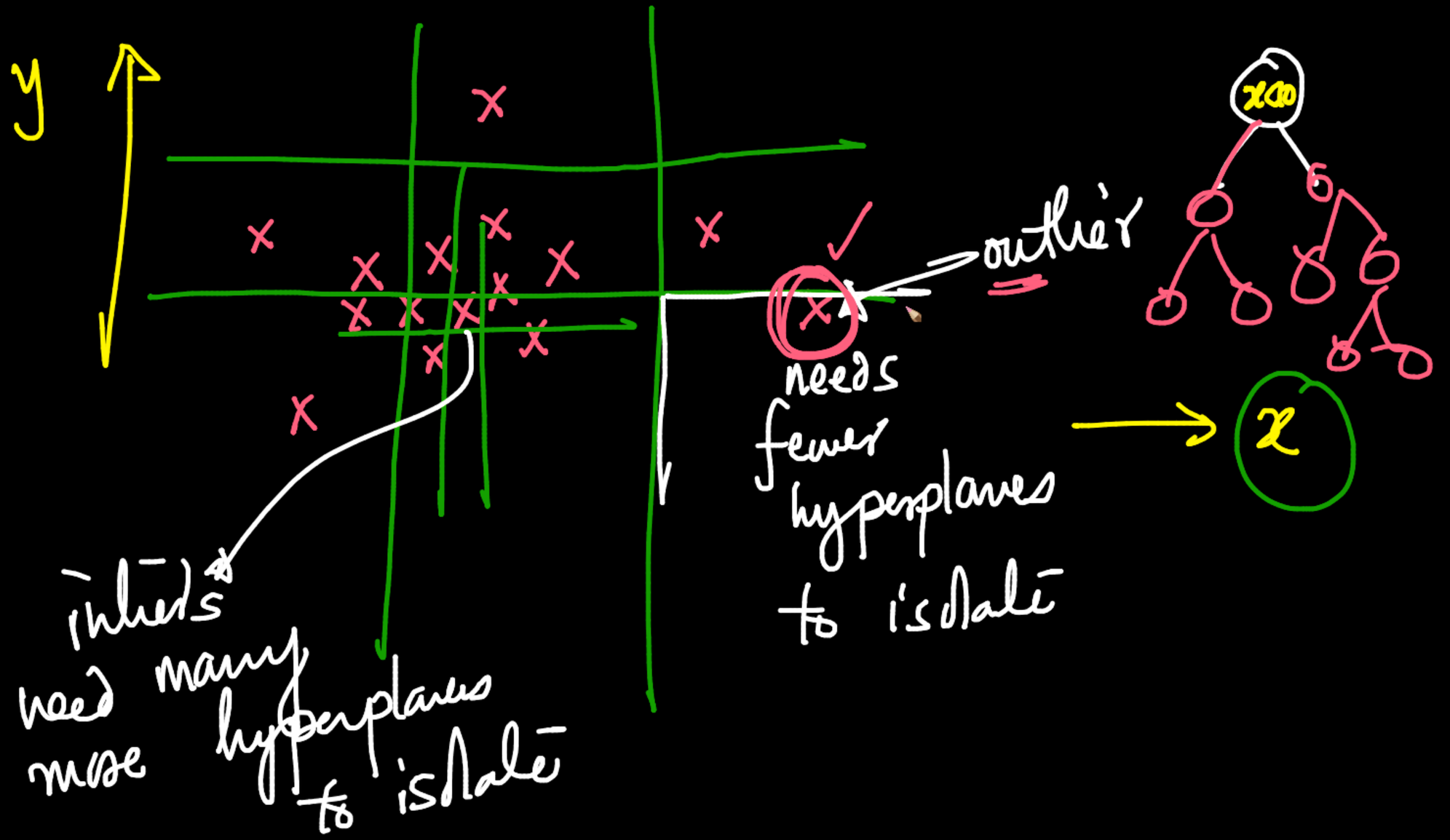
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

Isolation Forest



rest of the sample (isolate), compared to normal points. In order to isolate a data point, the algorithm recursively generates partitions on the sample by randomly selecting an attribute and then randomly selecting a split value for the attribute, between the minimum and maximum values allowed for that attribute.

An example of random partitioning in a 2D dataset of normally distributed points is given in Fig. 2 for a non-anomalous point and Fig. 3 for a point that's more likely to be an anomaly. It is apparent from the pictures how anomalies require fewer random partitions to be isolated, compared to normal points.

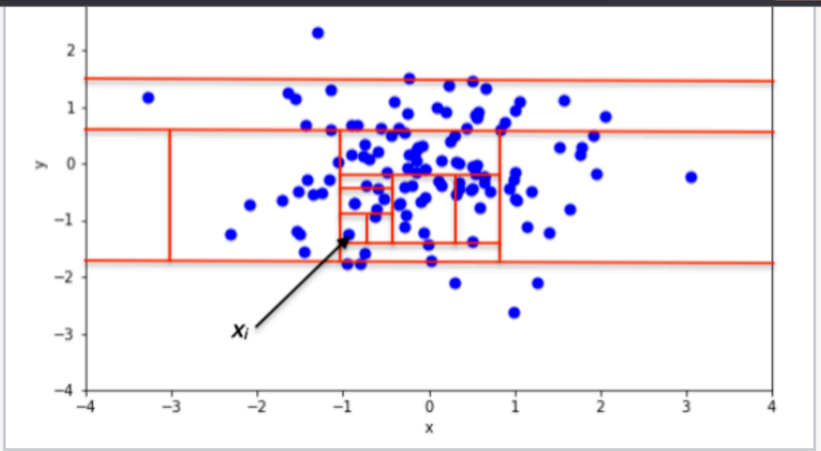


Fig. 2 - an example of isolating a non-anomalous point in a 2D Gaussian distribution.

From a mathematical point of view, recursive partitioning can be represented by a tree structure named *Isolation Tree*, while the number of partitions required to isolate a point can be interpreted as the length of the path, within the tree, to reach a terminating node starting from the root. For example, the path length of point x_i in Fig. 2 is greater than the path length of x_j in Fig. 3.

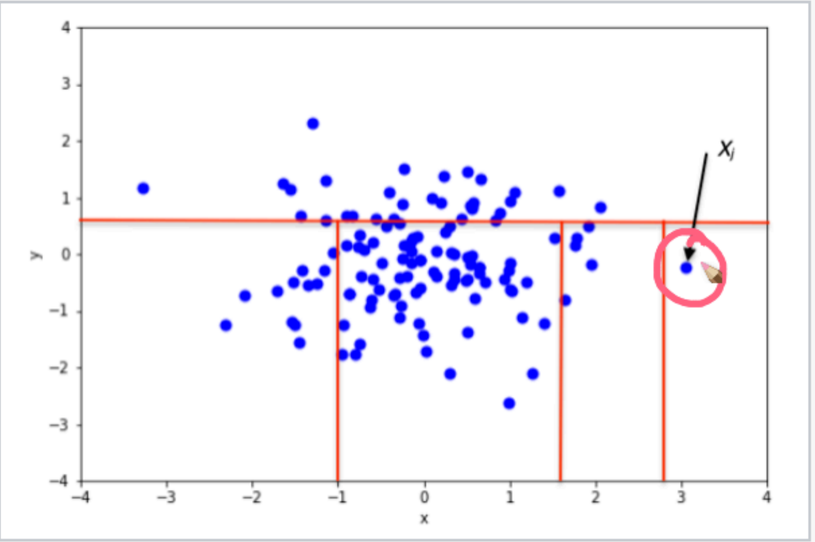


Fig. 3 - an example of isolating an anomalous point in a 2D Gaussian distribution.

More formally, let $X = \{x_1, \ldots, x_n\}$ be a set of d-dimensional points and $X' \subset X$. An Isolation Tree (iTree) is defined as a data structure with the following properties:

1. for each node T in the Tree, T is either an external node with

one global / local optima

optmzn

eig val & vec

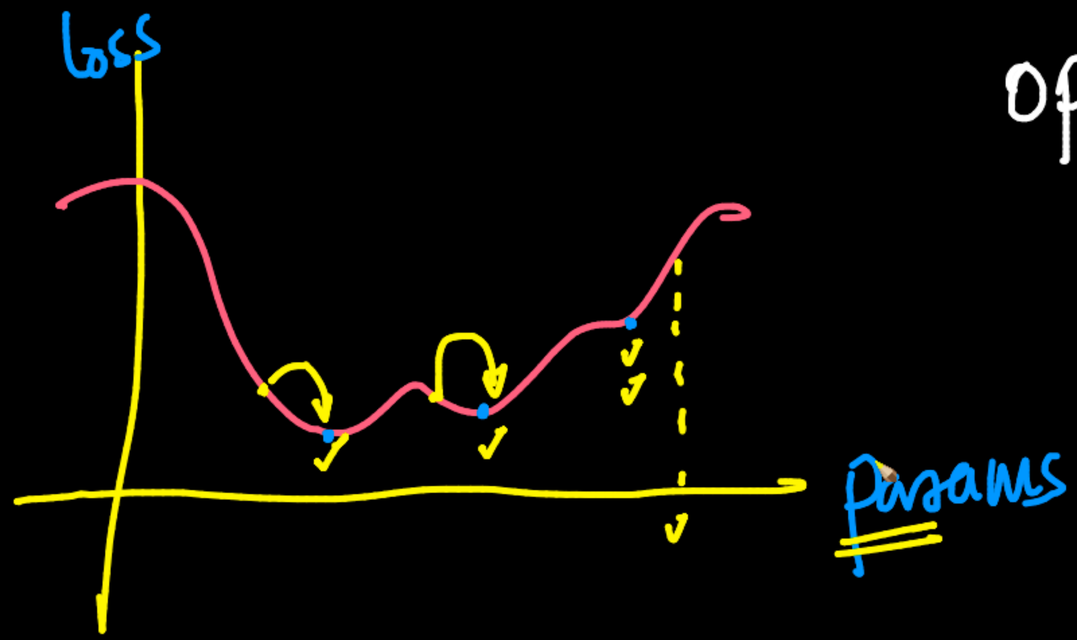
PCA

"tSNE"

optmzn

multiple optmzn

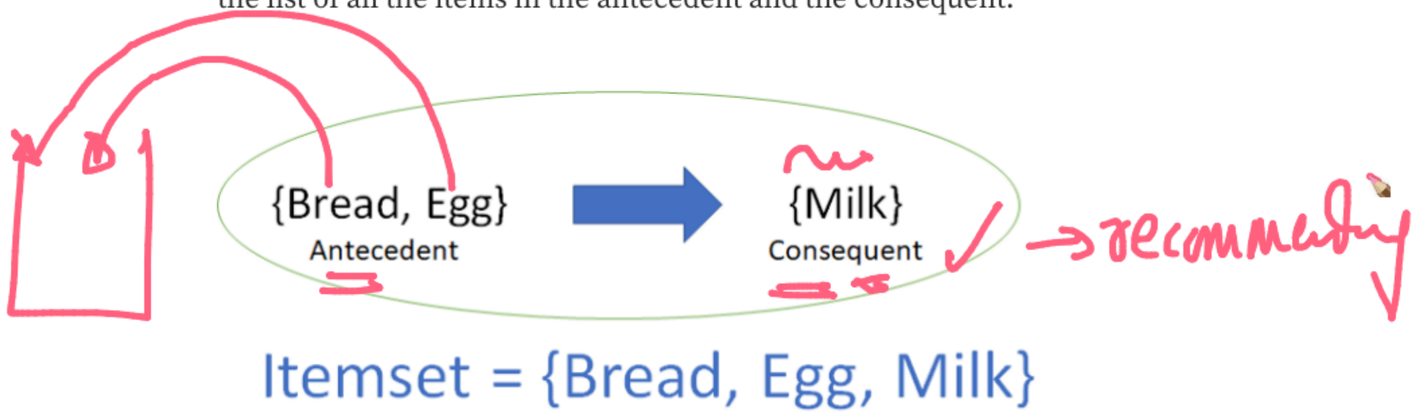
gradient descent



SGD

Adam

implication here is co-occurrence and not causality. For a given rule, *itemset* is the list of all the items in the antecedent and the consequent.



Various metrics are in place to help us understand the strength of association between these two. Let us go through them all.

1. Support

This measure gives an idea of how frequent an *itemset* is in all the transactions. Consider *itemset1* = {bread} and *itemset2* = {shampoo}. There will be far more transactions containing bread than those containing shampoo. So as you rightly guessed, *itemset1* will generally have a higher support than *itemset2*. Now consider *itemset1* = {bread, butter} and *itemset2* = {bread, shampoo}. Many transactions will have both bread and butter on the cart but bread and shampoo? Not so much. So in this case, *itemset1* will generally have a higher support than *itemset2*. Mathematically, support is the fraction of the total number of transactions in which

Anisha Garg
587 Followers
Data Scientist | UT Austin, IIT Bombay alum
Follow

More from Medium

Chi Nguyen in Towards Data Science
Introduction to Simple Association Rules Mining for Market Basket Analysis

Albers Uzila in Towards Data Science
K-means Clustering and Principal Component Analysis in 10 Minutes

Zach ... in Pipeline: A Data Engineer...
3 Data Science Projects That Got Me 12 Interviews. And 1 That Got Me in Trouble.

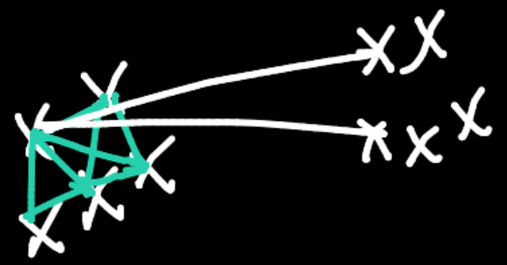
Anna Wu
Google Data Scientist Interview Questions (Step-by-Step Solutions!)

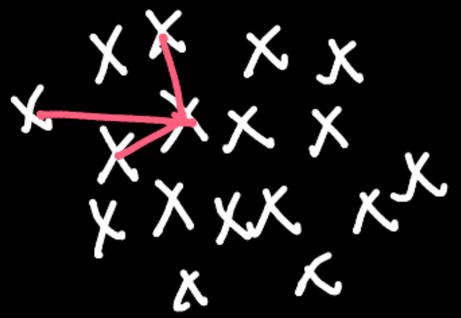
Help Status Writers Blog Careers Privacy Terms About
Text to speech

(Q) Is the data clusterable?

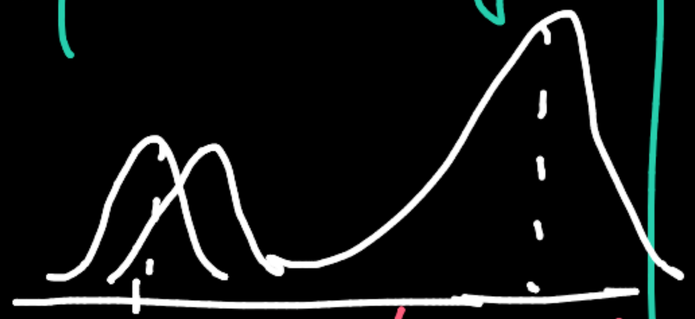
$\{x_i \in \mathbb{R}^d\}_{i=1}^{n}$

✓ { V. good





Compute $d(x_i, x_j)$



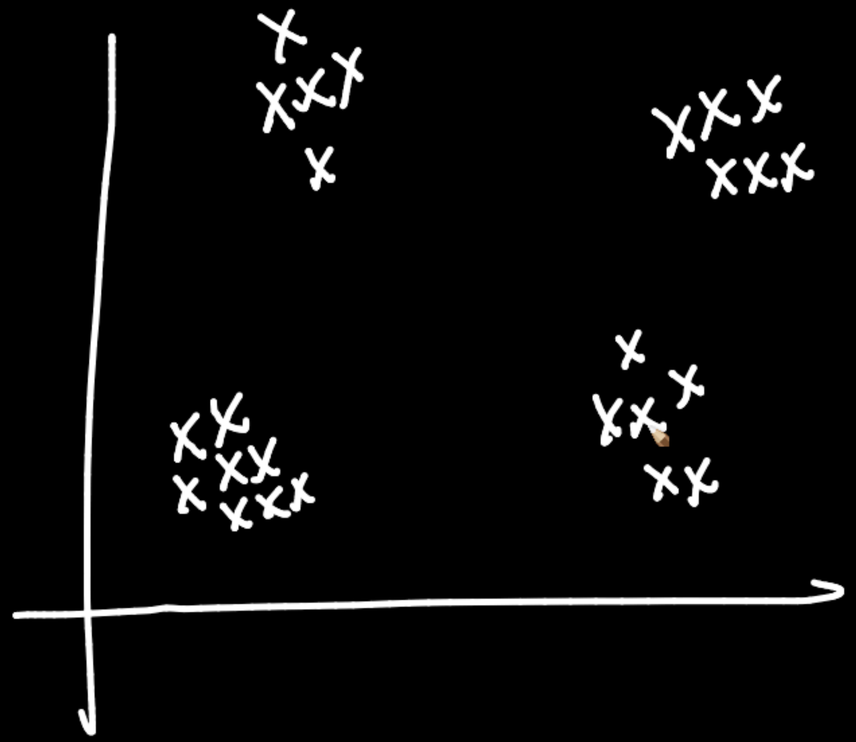
multi-modal

Uni-modal

① $\begin{cases} H_0: \text{no clusters / random pts} \\ H_a: \exists \text{ clusters} \end{cases}$

② $T =$ Sensible

③ disb of T



Recent changes
Upload file

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages
Add links

Preliminaries [edit]

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.

Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.

Notes and references [edit]

1. ^ Hopkins, Bri... rmining the type of distribution of plant individuals".

Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Languages
Add links

Preliminaries [edit]

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

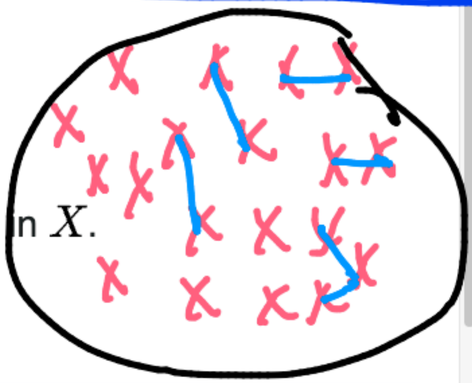
Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.



Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.

Notes and references [edit]

1. ^ Hopkins, Bri... determining the type of distribution of plant individuals".

Recent changes
Upload file

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages

Add links

Preliminaries [edit]

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.

Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.

Notes and references [edit]

1. ^ Hopkins, Bri[illegible]rmining the type of distribution of plant individuals".

Recent changes
Upload file

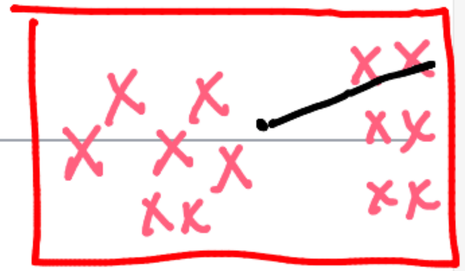
Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages
Add links



Preliminaries [edit]

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

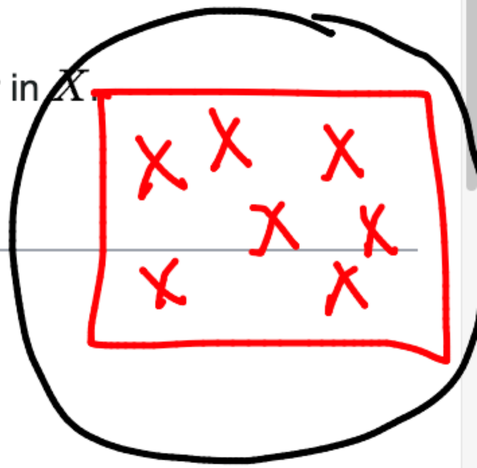
Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.



Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.

Notes and references [edit]

1. ^ Hopkins, Bri... rmining the type of distribution of plant individuals".

Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages
Add links

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.

Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.

Notes and references [edit]

1. ^ Hopkins, Brian; Skellam, John Gordon (1954). "A new method for determining the type of distribution of plant individuals". *Annals of Botany*. Annals Botany Co. **18** (2): 213–227.
2. ^ a b Banerjee, A. (2004). "Validating clusters using the Hopkins statistic". *IEEE International Conference on Fuzzy Systems*: 149–153. doi:10.1109/FUZZY.2004.1375706.
3. ^ Aggarwal, Charu C. (2015). *Data Mining*. Cham: Springer International Publishing. p. 158. doi:10.1007/978-3-319-14142-8. ISBN 978-3-319-14141-1.
4. ^ Cross, G.R.; Jain, A.K. (1982). "Measurement of clustering tendency". *Theory and Application of Digital Control*: 315-320. doi:10.1016/B978-0-08-027618-2.50054-1.

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages
Add links

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

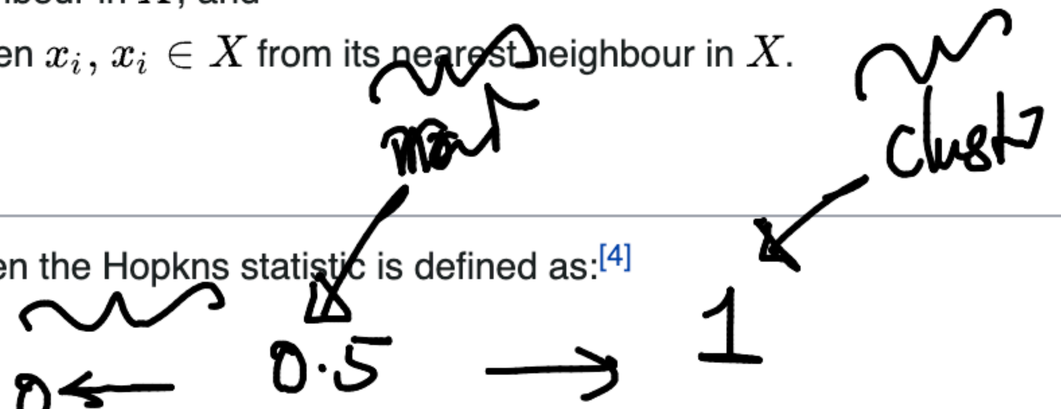
w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.

Definition [edit]

With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

Under the null hypotheses, this statistic has a Beta(m,m) distribution.



Notes and references [edit]

1. ^ Hopkins, Brian; Skellam, John Gordon (1954). "A new method for determining the type of distribution of plant individuals". *Annals of Botany*. Annals Botany Co. **18** (2): 213–227.
2. ^ a b Banerjee, A. (2004). "Validating clusters using the Hopkins statistic". *IEEE International Conference on Fuzzy Systems*: 149–153. doi:

Upload file

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export

Download as PDF
Printable version

Languages

Add links

Preliminaries [edit]

A typical formulation of the Hopkins statistic follows.[2]

Let X be the set of n data points.

Consider a random sample (without replacement) of $m \ll n$ data points with members x_i.

Generate a set Y of m uniformly randomly distributed data points.

Define two distance measures,

u_i, the distance of $y_i \in Y$ from its nearest neighbour in X, and

w_i, the distance of m number of randomly chosen x_i, $x_i \in X$ from its nearest neighbour in X.

Definition [edit]

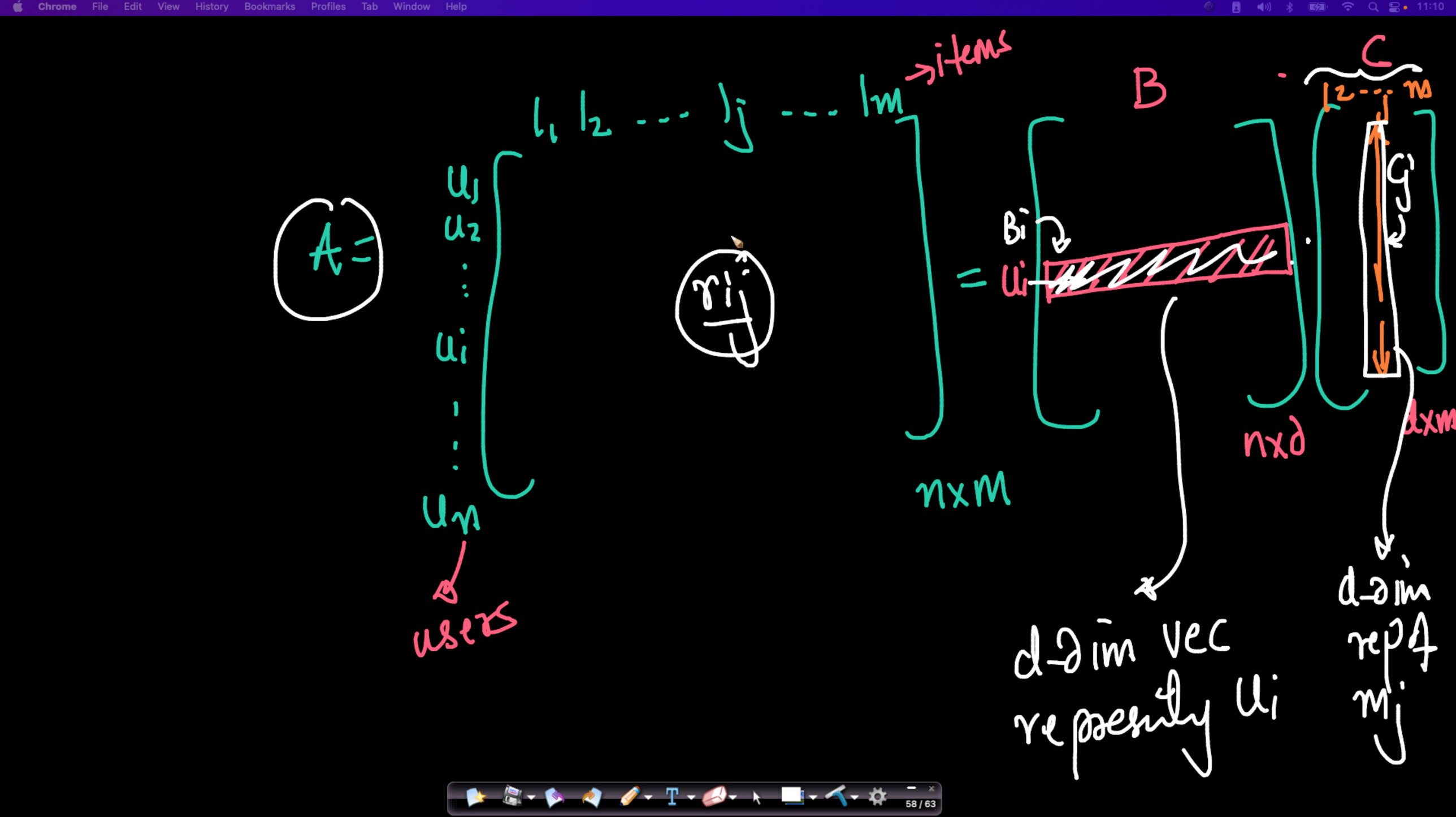
With the above notation, if the data is d dimensional, then the Hopkns statistic is defined as:[4]

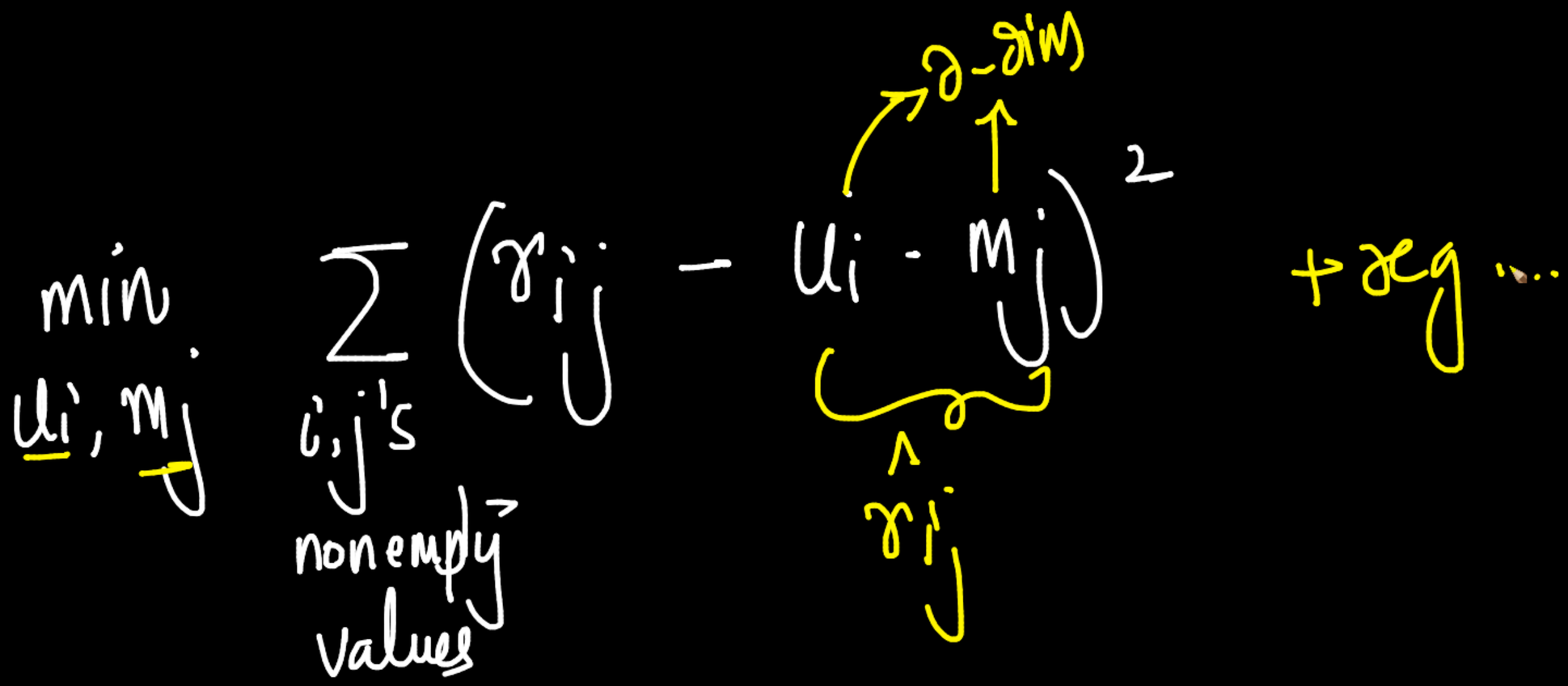
$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

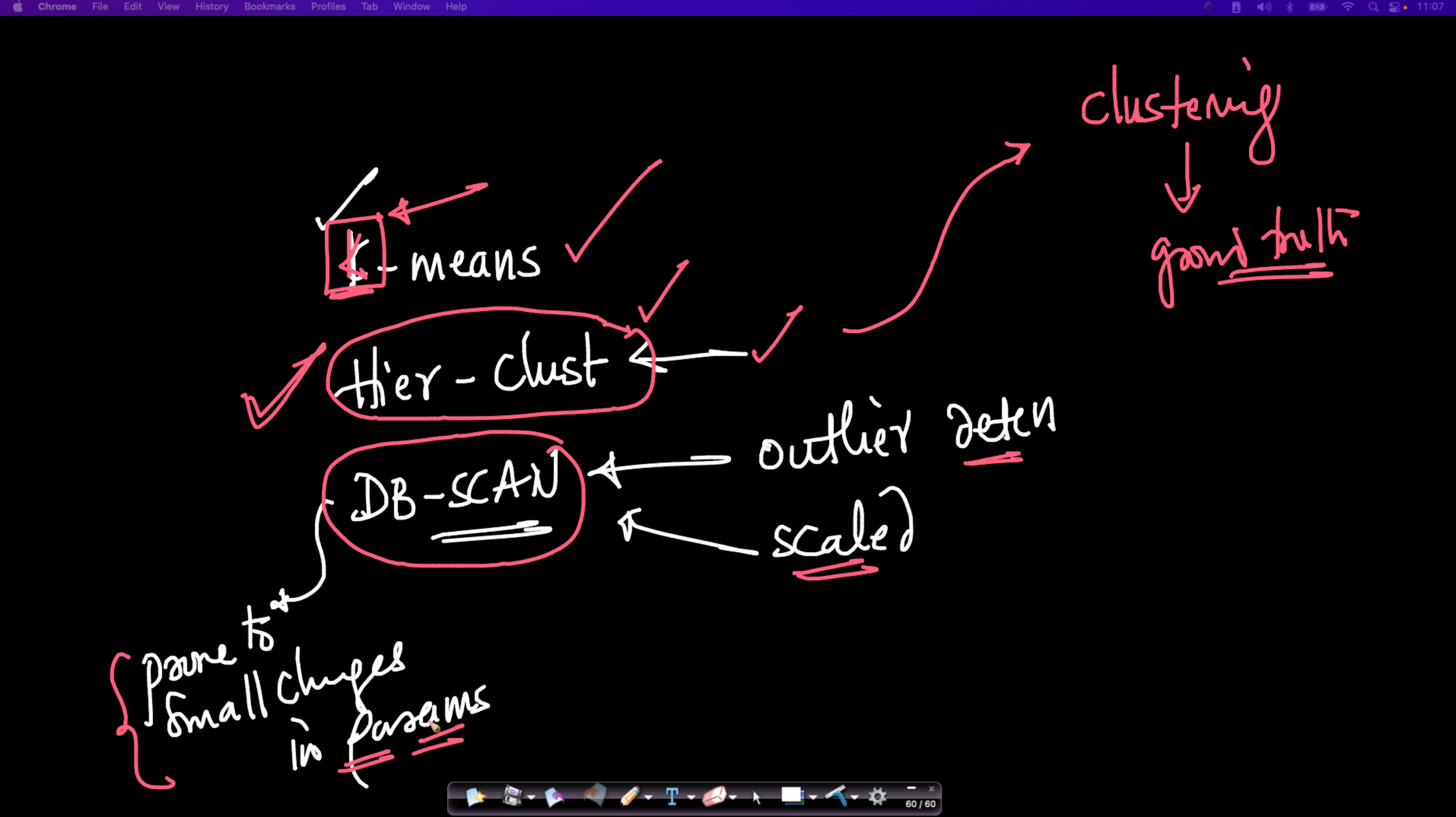
Under the null hypotheses, this statistic has a Beta(m,m) distribution.

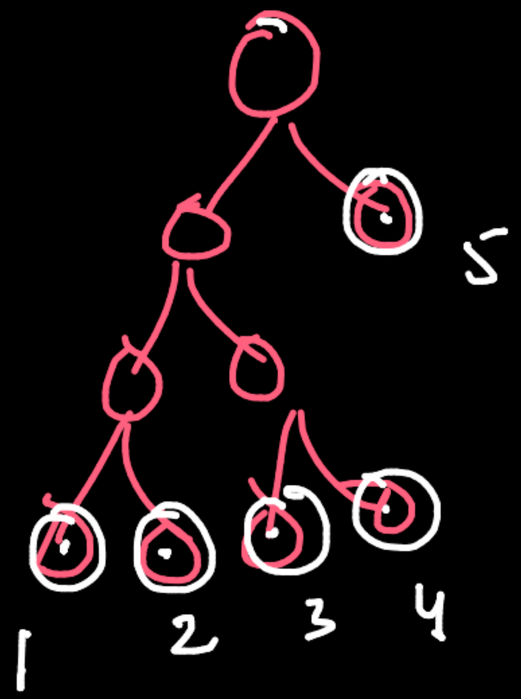
Notes and references [edit]

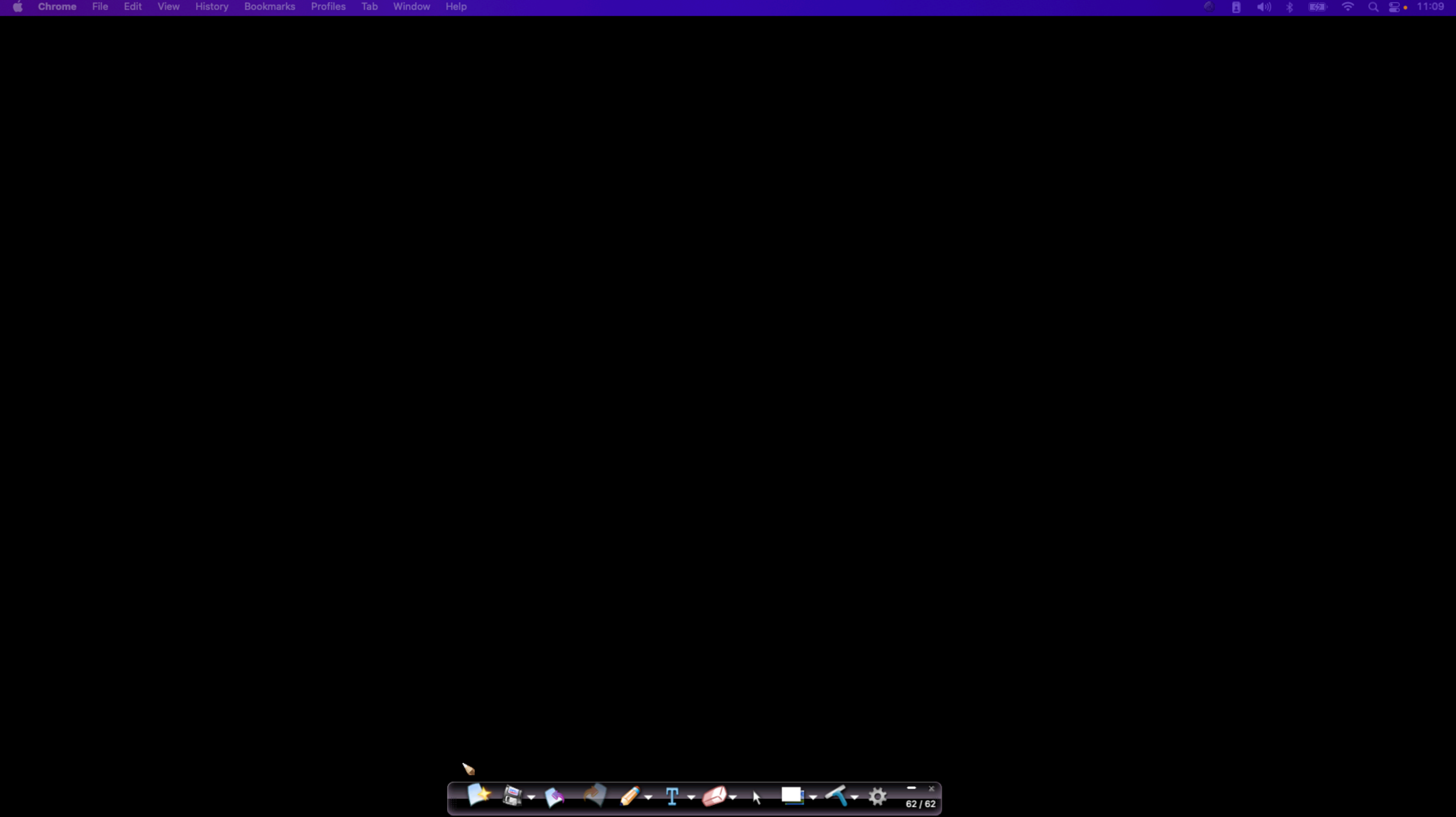
1. ^ Hopkins, Brian; Skellam, John Gordon (1954). "A new method for determining the type of distribution of plant individuals".
Annals of Bota


A =
l_1 l_2 ... l_j ... l_m
items
u_1
u_2
u_i
u_n
users
r_i
= B
Bi
u_i
nxm
nxd
C
l_2 ... l_m
Cj
dxm
d dim vec
represents u_i
d dim rep of m_j


min
u_i, m_j
$\sum$
i,j's
nonempty
values
$(r_{ij} - u_i \cdot m_j)^2$
d-dim
r_{ij}
+ reg ...


clustering
good truth
K - means
Hier - clust
DB-SCAN
outlier data
scaled
prone to small changes in params
60 / 60




Chrome File Edit View History Bookmarks Profiles Tab Window Help
11:09
62 / 62

multi objective problems

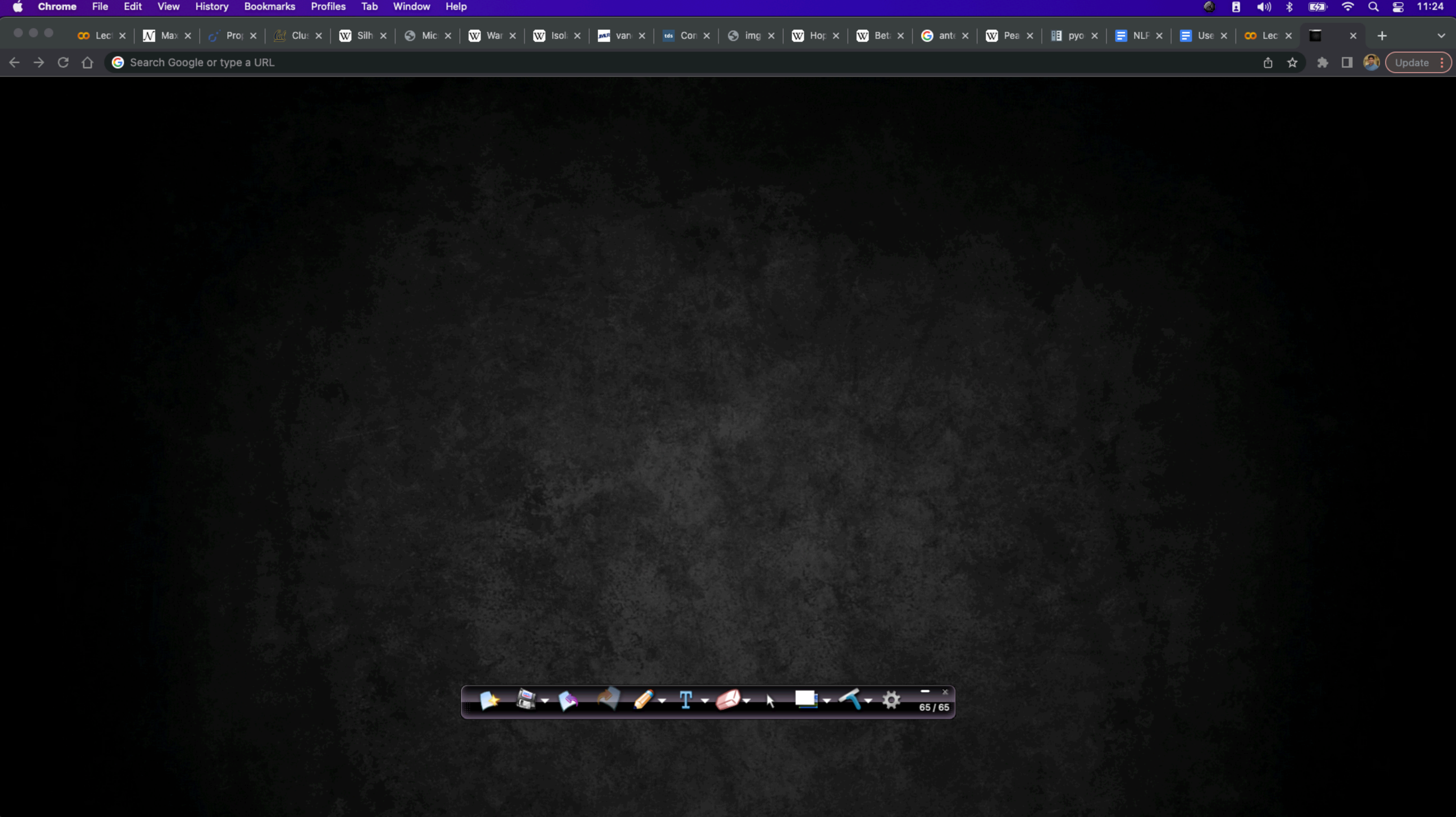
$$\min_{\text{assign}} \; \alpha \cdot \text{Intra clust dist} + \beta \cdot \left(\text{Inter clust dist}\right)^{-1}$$

$$\{\text{lr.reg} : \min_{w,b} \; \text{mse} + \lambda \text{reg}$$

$$\max_{\theta} \quad \text{Sales}(\theta)$$

$$\text{s.t} \quad \text{Spend} \leq \text{th}$$

Lagrange mult


Chrome File Edit View History Bookmarks Profiles Tab Window Help
11:24
Search Google or type a URL
Update
65 / 65