Abdul Qayyum

Lecturer at University of Burgundy, France

- Postdoc in Electrical and Informatics Engineering
- PhD in Electrical & Electronics Engineering
- Masters in Electronics Engineering
- Bachelor in Computer Engineering

Collaborations & Expertise:

Topic: Introduction of Machine Learning

Instructor: Abdul Qayyum, PhD

Class: MSCV

University of Burgundy, France

Outlines of Lecture:

In this lecture we will consider

- What is Artificial intelligence, Machine Learning ?
- Comparison of ML algorithms
- Feature Extraction and Scaling
- Applications of ML
- Introduction and background of statically model
- Demo on Image Segmentation using handcrafted features
- Practical use case studies on segmentation and classification

Introduction

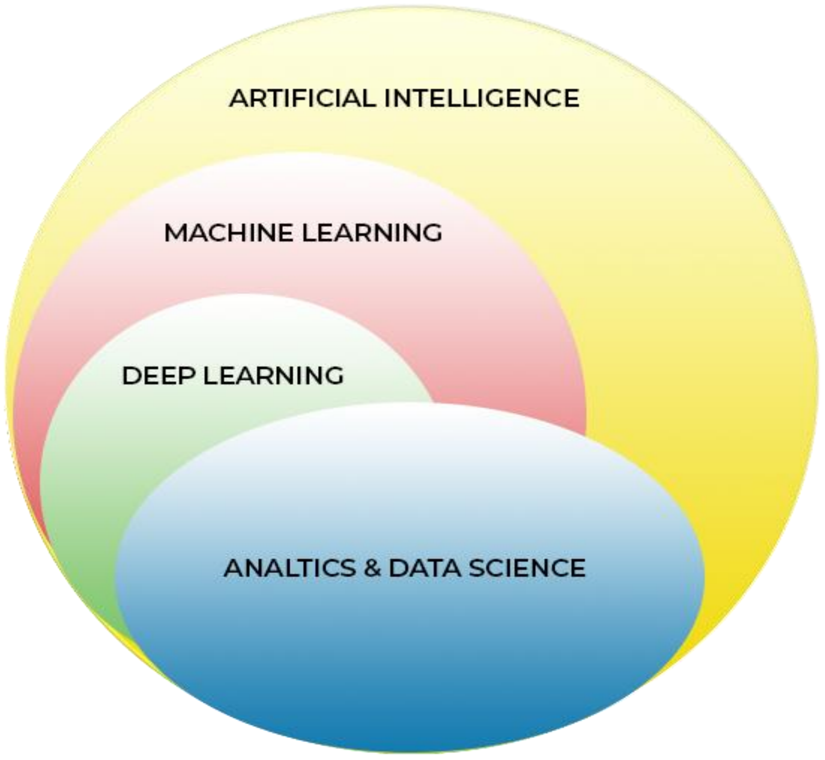
- AI and machine learning are often used interchangeably, especially in the realm of big data.

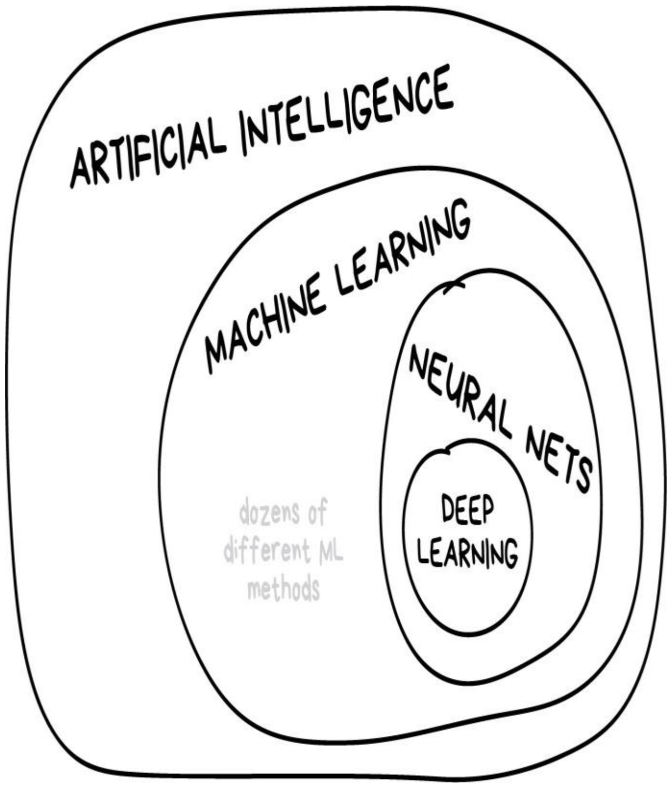
- **Artificial intelligence** is a broader concept than machine learning, which addresses the
use of computers to mimic the cognitive functions of humans.

- When machines carry out tasks based on algorithms in an "intelligent" manner, that is AI.

- **Machine learning** is a subset of AI and focuses on the ability of machines to receive a set
of data and learn for themselves, changing algorithms as they learn more about the information
they are processing.

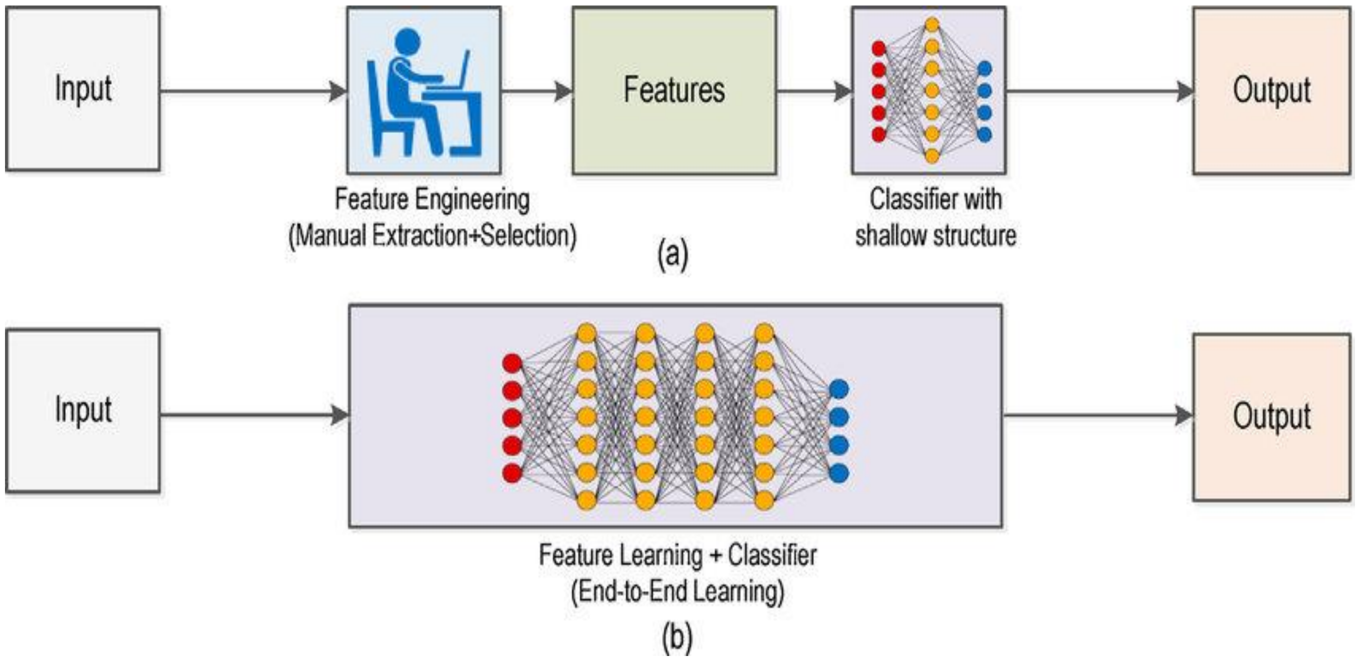
- **Deep learning** goes yet another level deeper and can be considered a subset of machine learning.
The concept of deep learning is sometimes just referred to as "deep neural networks,"
referring to the many layers involved. A neural network may only have a single layer of data,
while a deep neural network has two or more.





Source: https://datascience.stackexchange.com/questions/16422/machine-learning-vs-deep-learning

Difference ML and DL



Machine Learning

- Good results with small data sets
- Quick to train a model
- Need to try different features and classifiers to achieve best results
- Accuracy plateaus

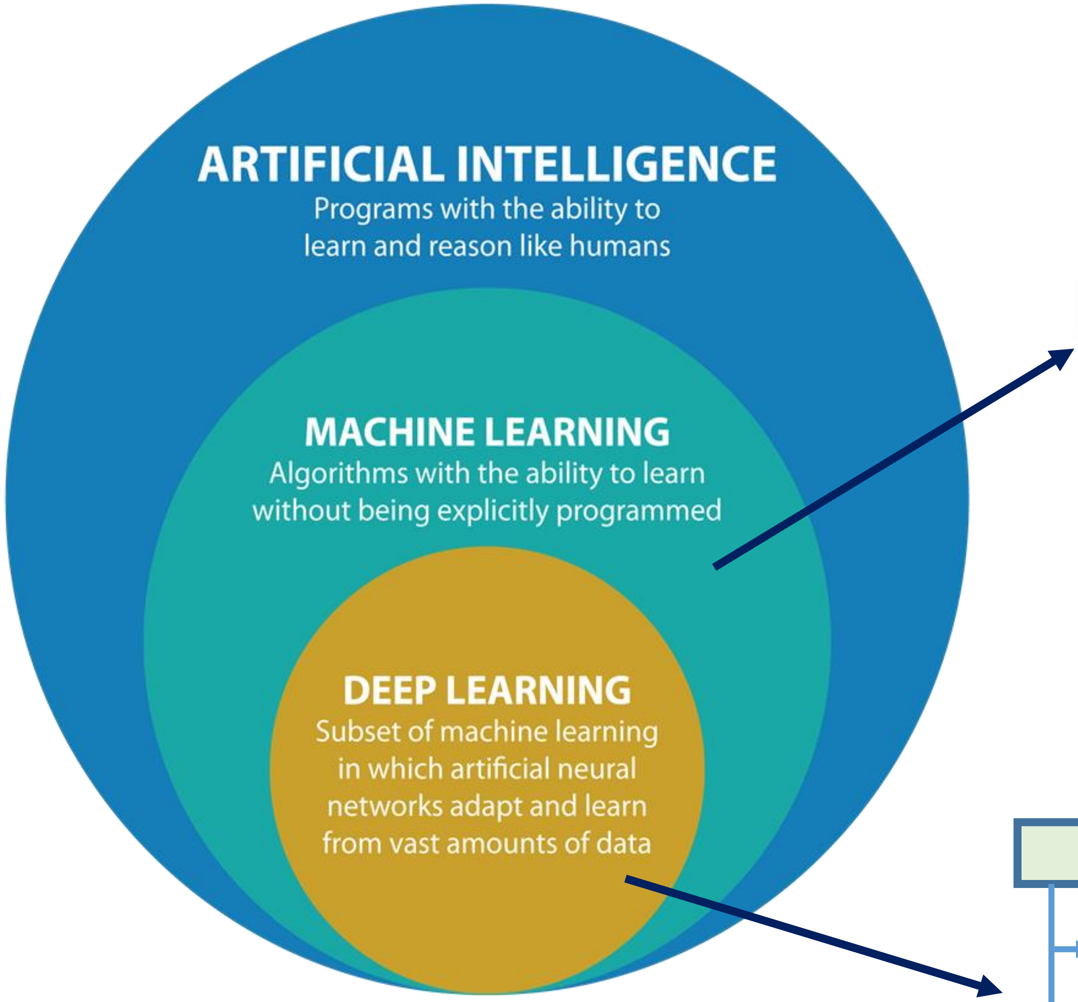
Deep Learning

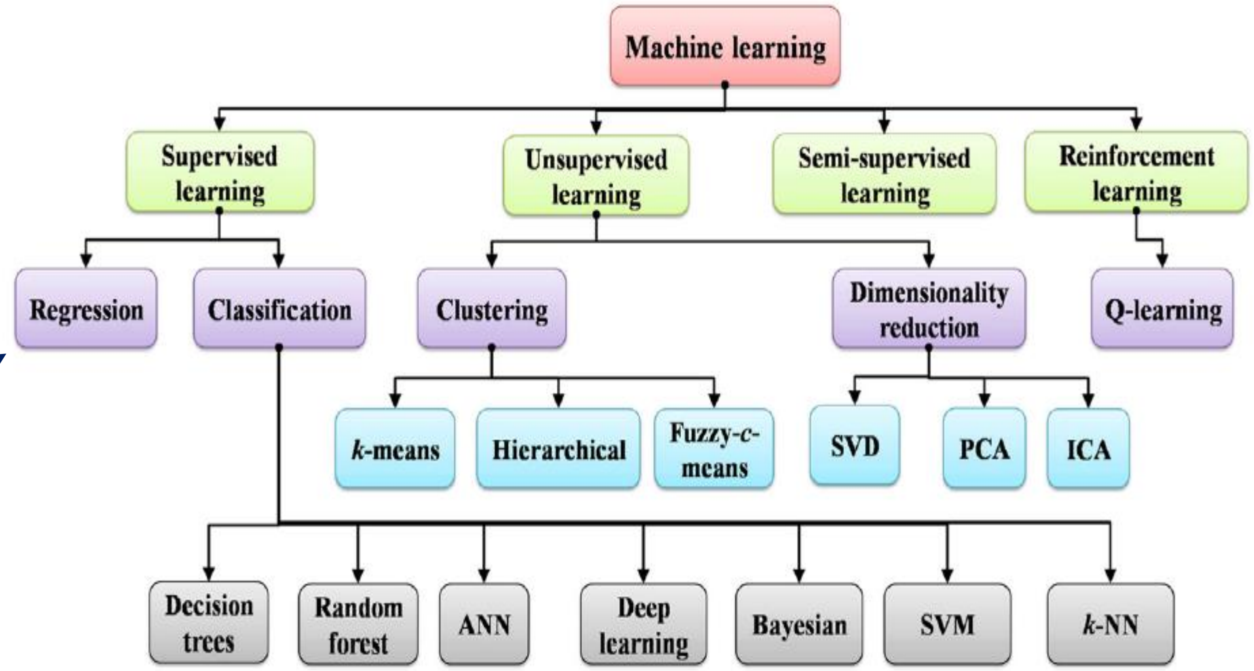
- Requires very large data sets
- Computationally intensive
- Learns features and classifiers automatically
- Accuracy is unlimited

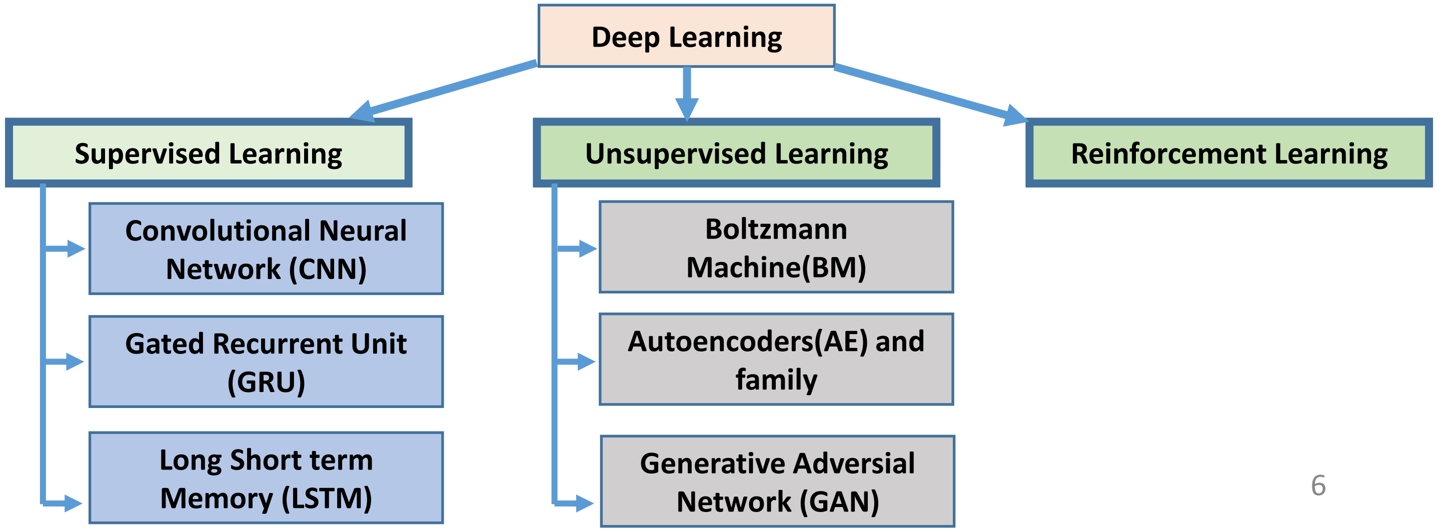
Deep Learning Vs Machine Learning

Factors	Deep Learning	Machine Learning
Data Requirement	Requires large data	Can train on lesser data
Accuracy	Provides high accuracy	Gives lesser accuracy
Training Time	Takes longer to train	Takes less time to train
Hardware Dependency	Requires GPU to train properly	Trains on CPU
Hyperparameter Tuning	Can be tuned in various different ways.	Limited tuning capabilities

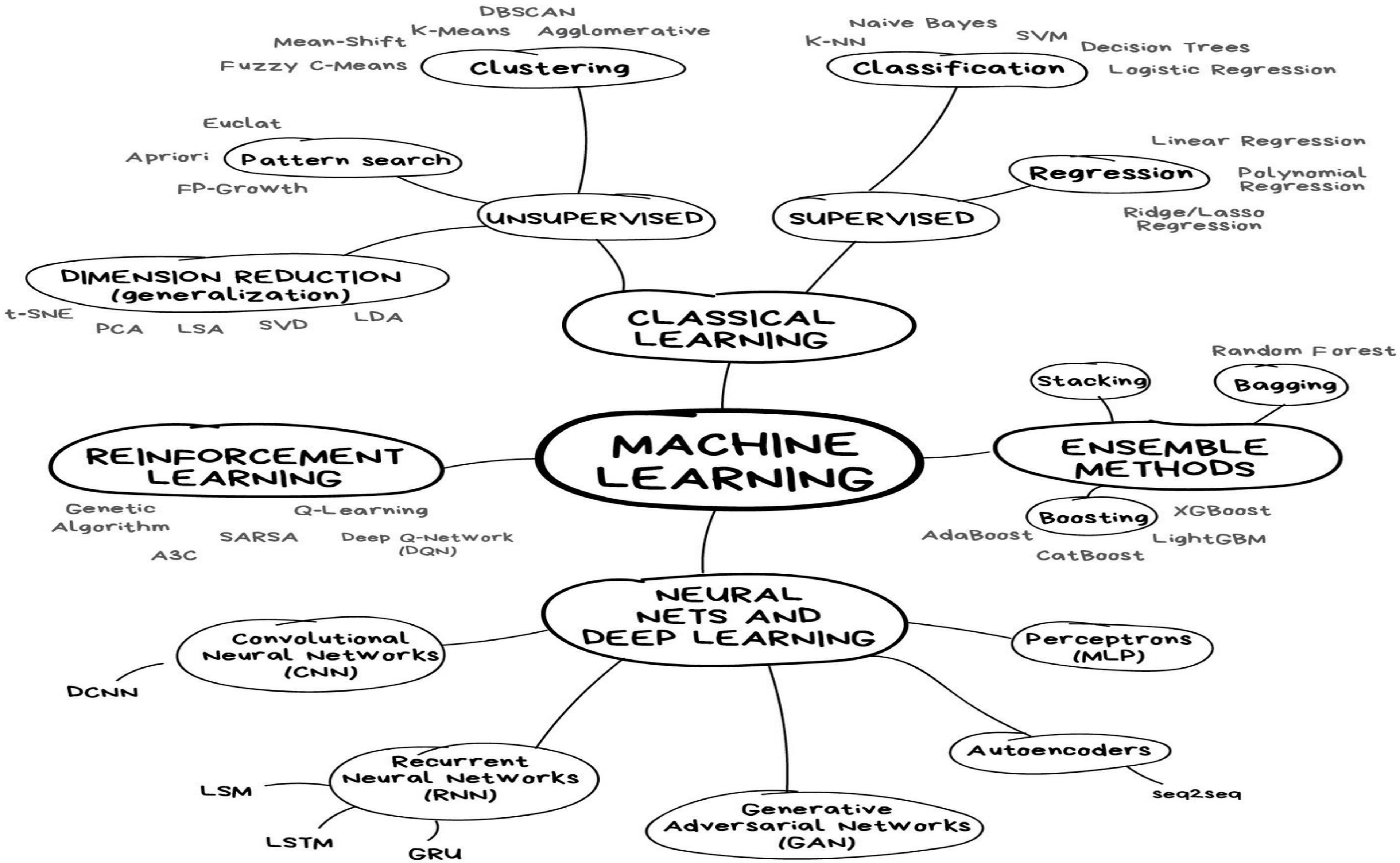
Algorithms of ML and DL



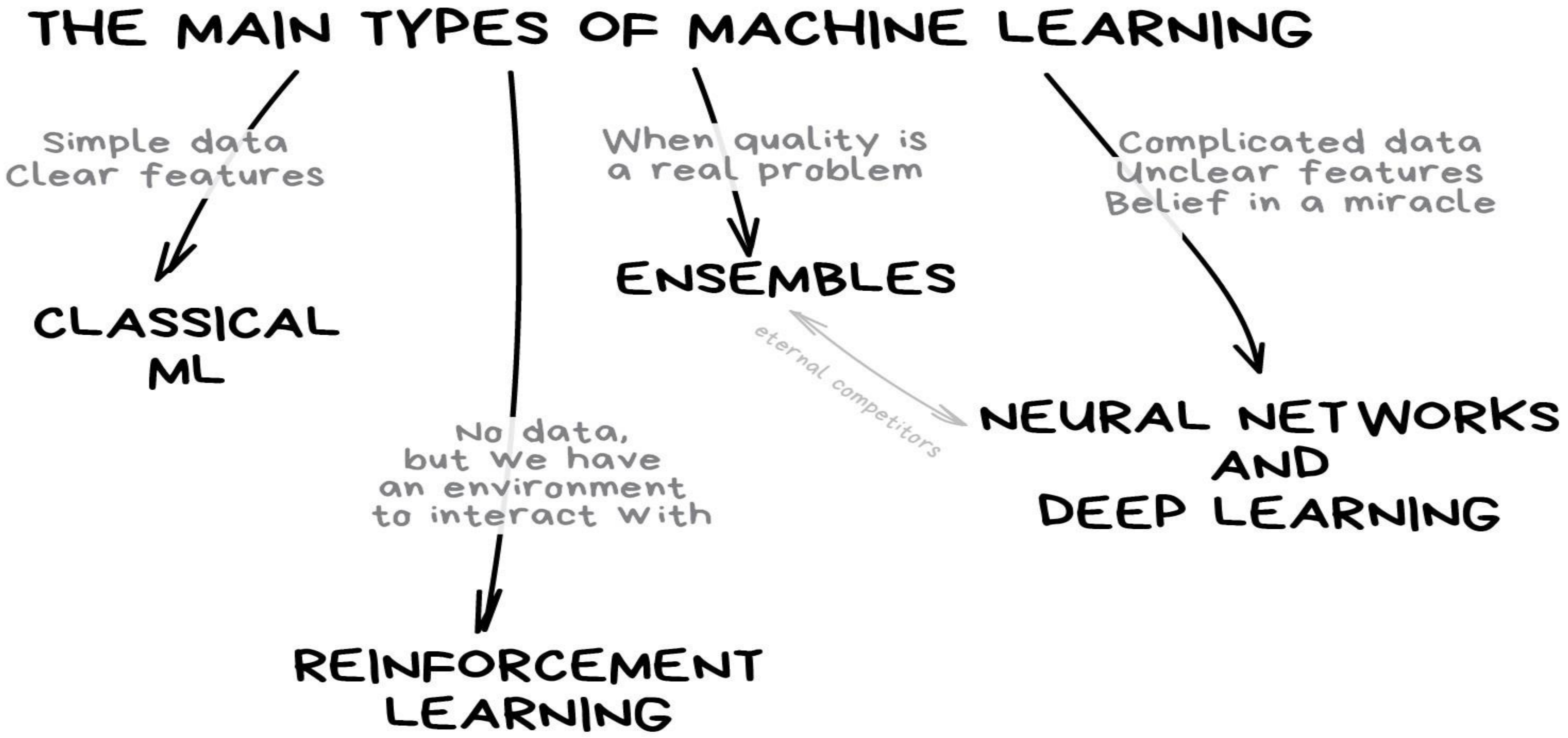




Machine Learning Tree

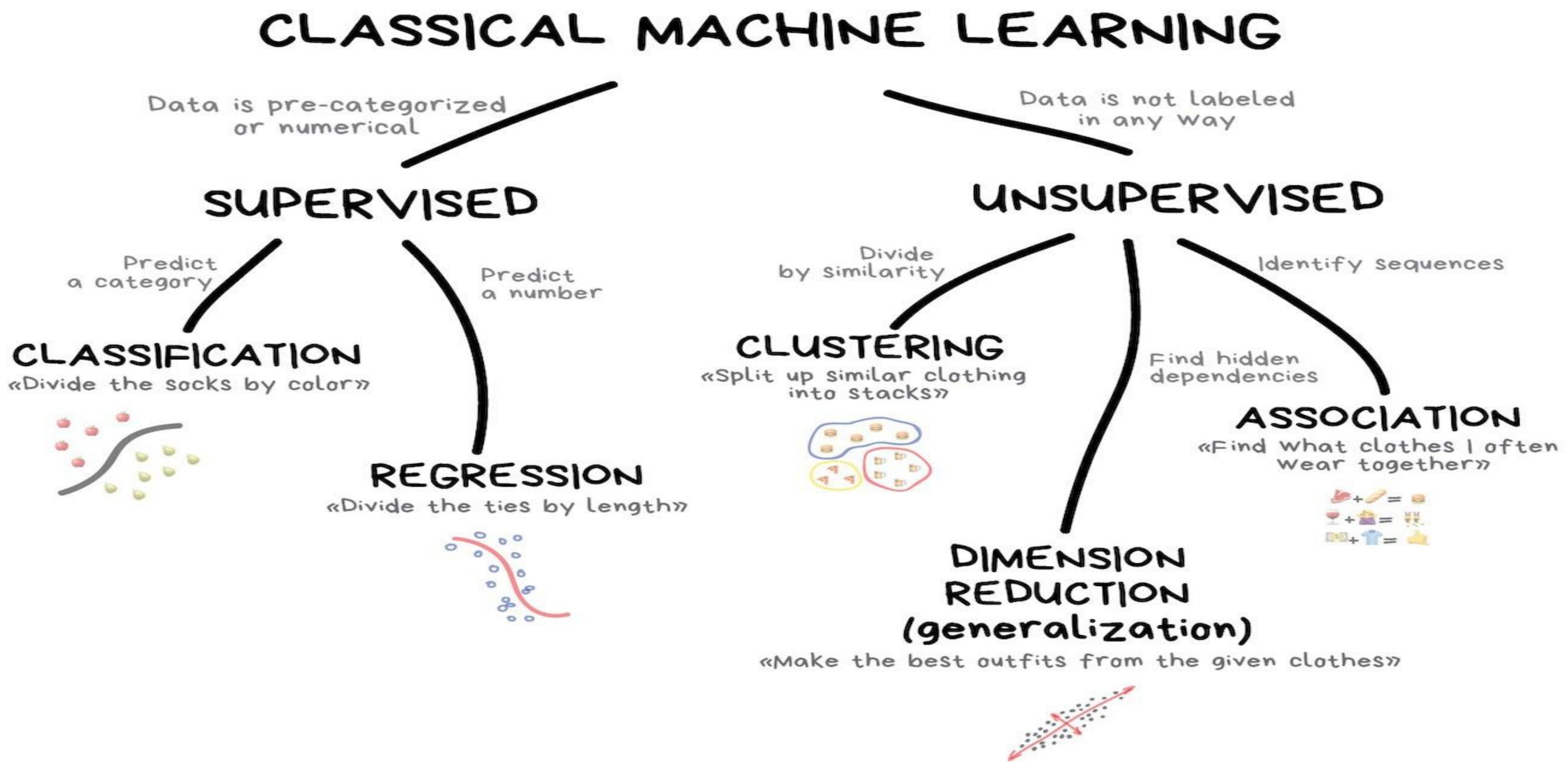


Types of Machine Learning



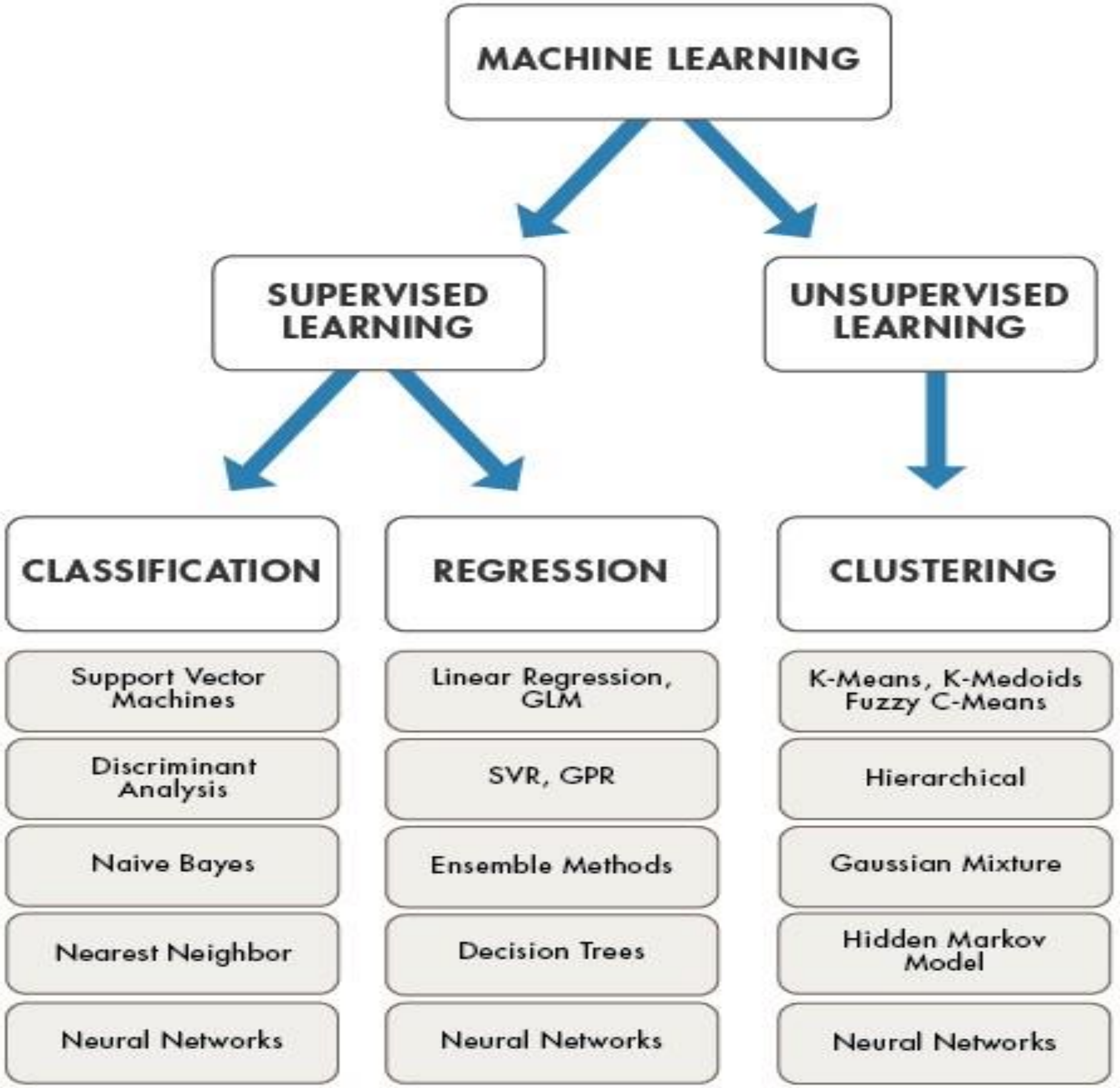
https://noeliagorod.com/2019/05/21/machine-learning-for-everyone-in-simple-words-with-real-world-examples-yes-again/

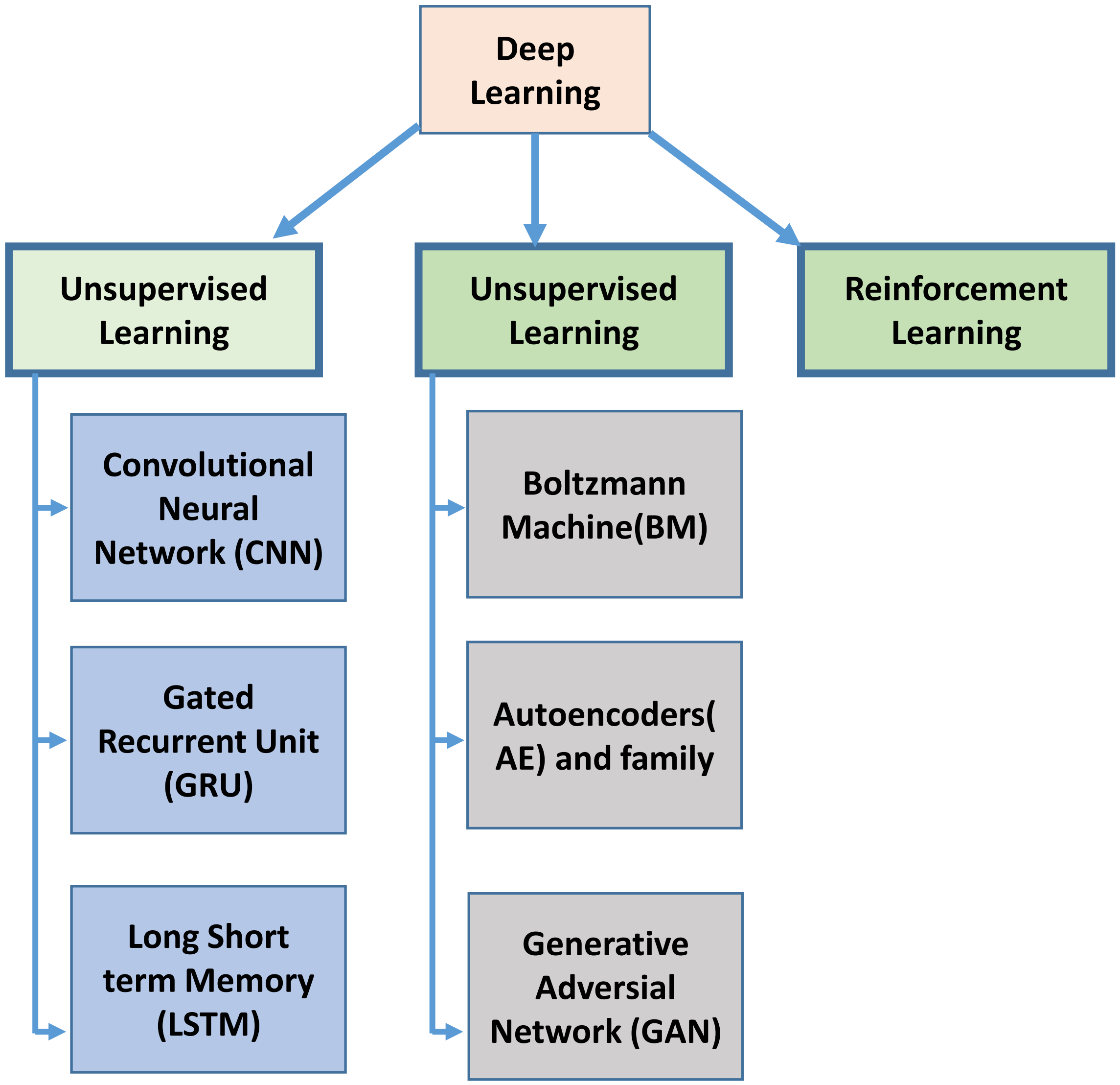
Classical Machine Learning



https://noeliagorod.com/2019/05/21/machine-learning-for-everyone-in-simple-words-with-real-world-examples-yes-again/

Supervised and Unsupervised Learning



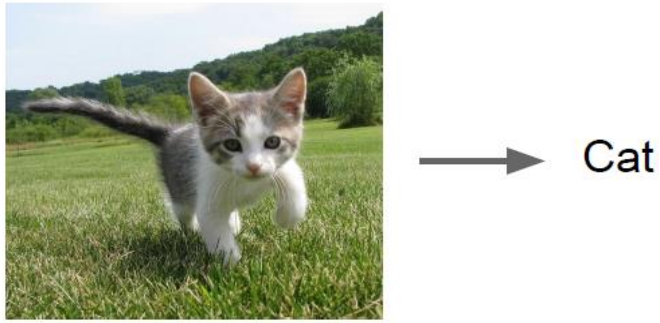


Supervised Learning

Data: (x,y)
x is data, y is label

Goal: Learn a function to map x→ y

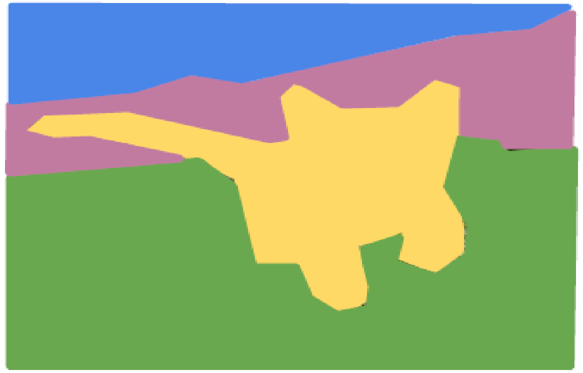
Examples: Classification, Regression, Object detection, Semantic Segmentation, Image captioning



Classification

DOG, DOG, CAT

Object Detection



GRASS, CAT, TREE, SKY

Semantic Segmentation

A cat sitting on a suitcase on the floor

Image captioning

Source: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture12.pdf

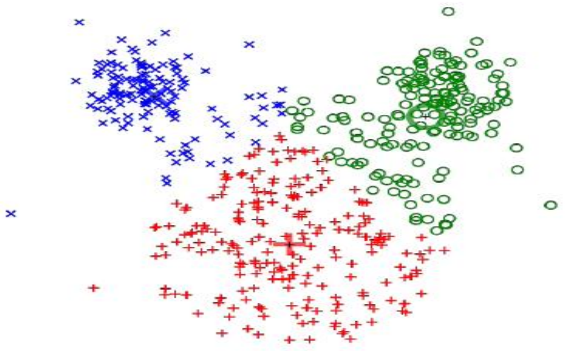
Unsupervised Learning

Data: x

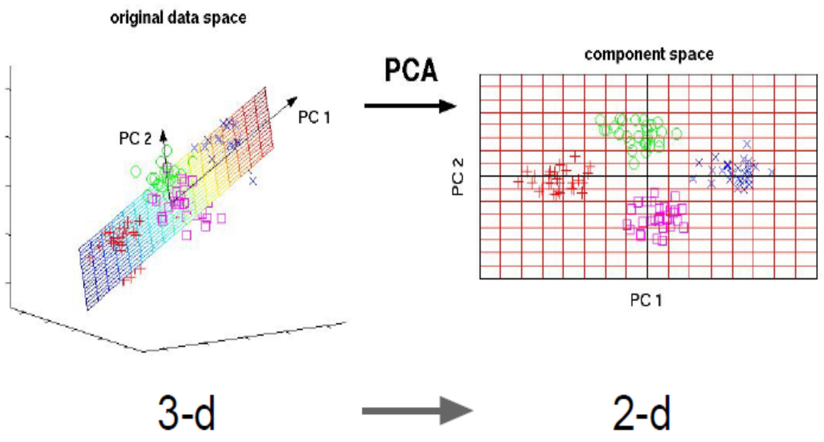
x is data, no label

Goal: Learn some underlying Hidden structure of the data

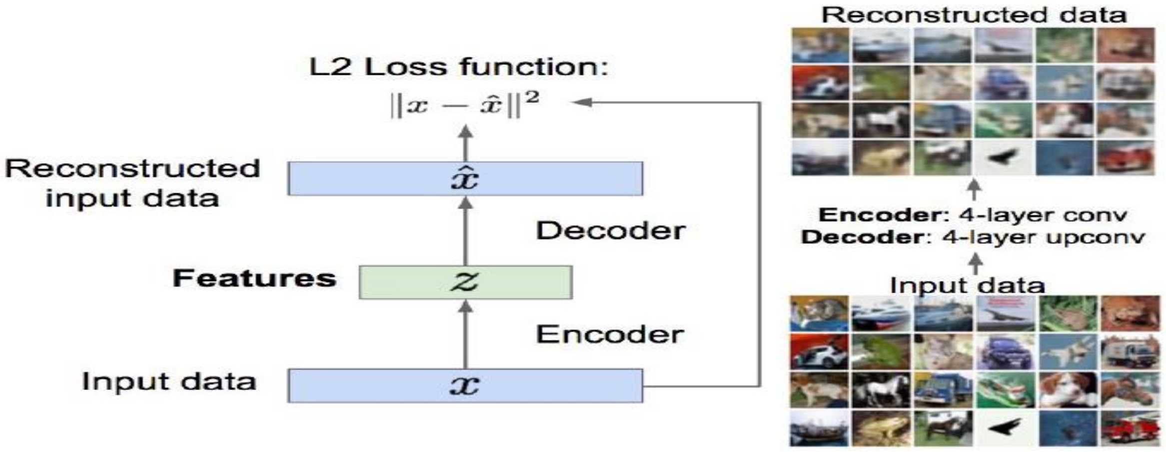
Examples: Clustering, Dimensionality reduction, Feature Learning, density estimation



K-means clustering



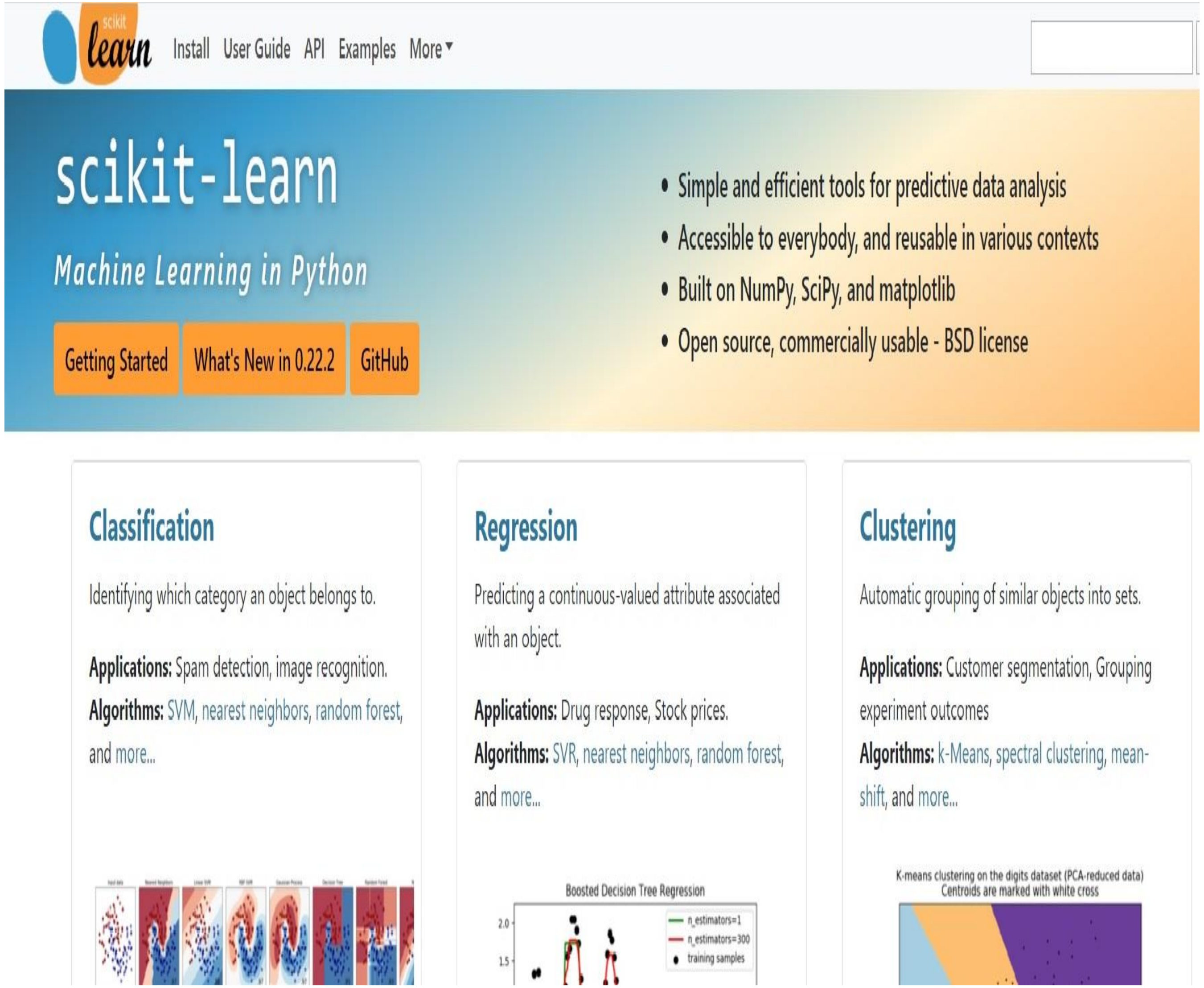
Principal Component Analysis
(Dimensionality reduction)

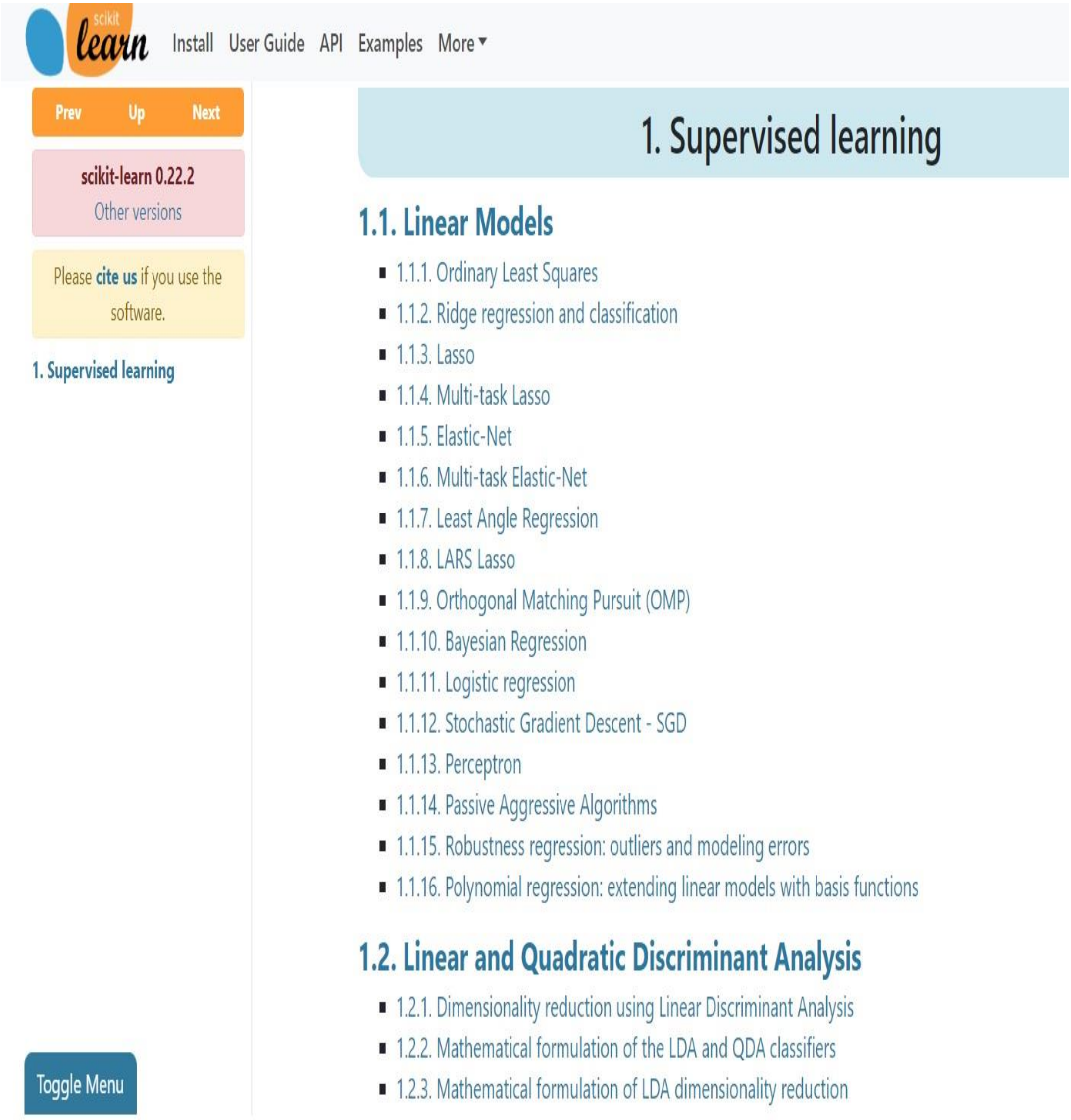


Autoencoders
(Feature learning)

Source: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture12.pdf

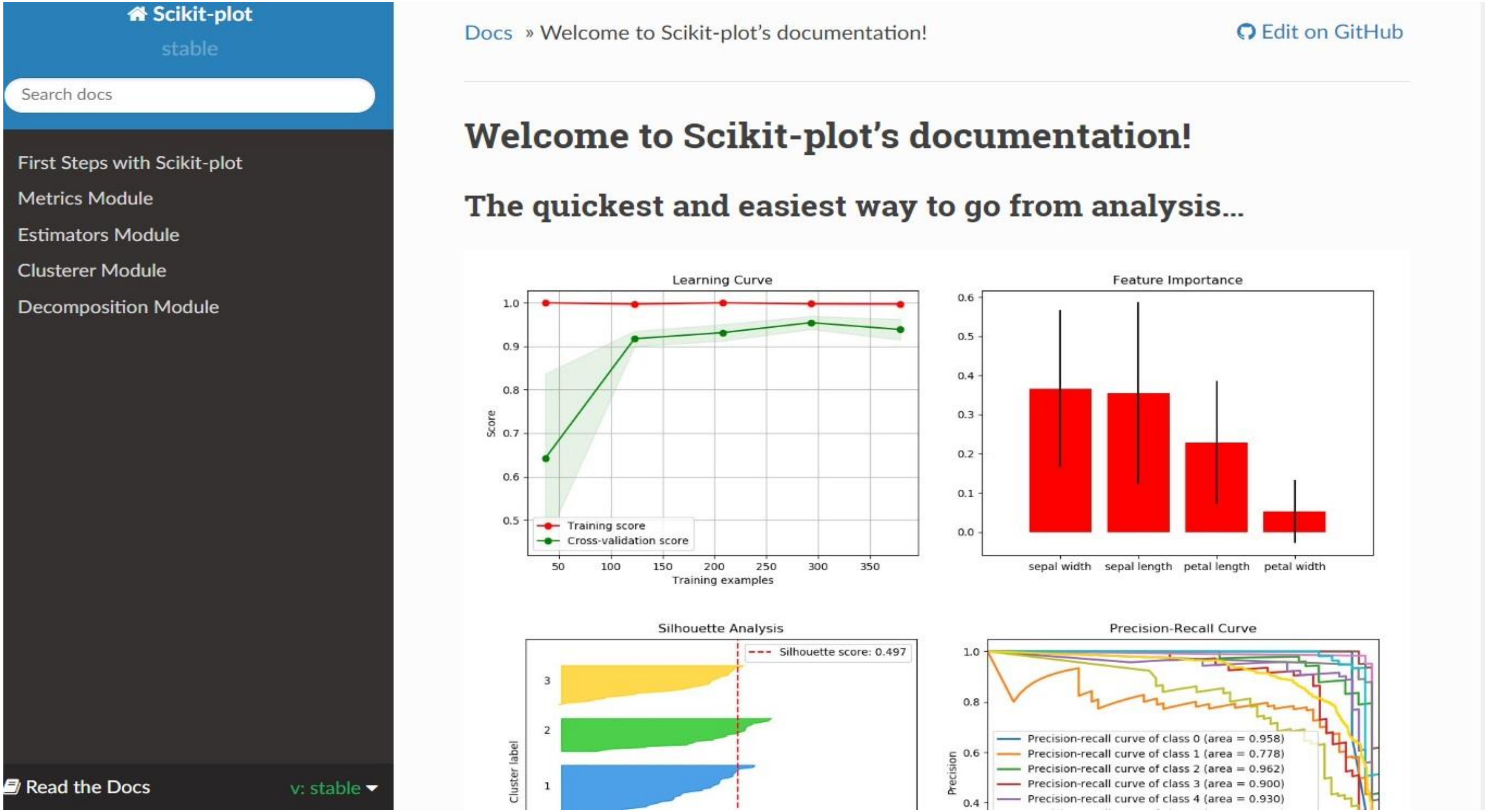
Scikit-learn





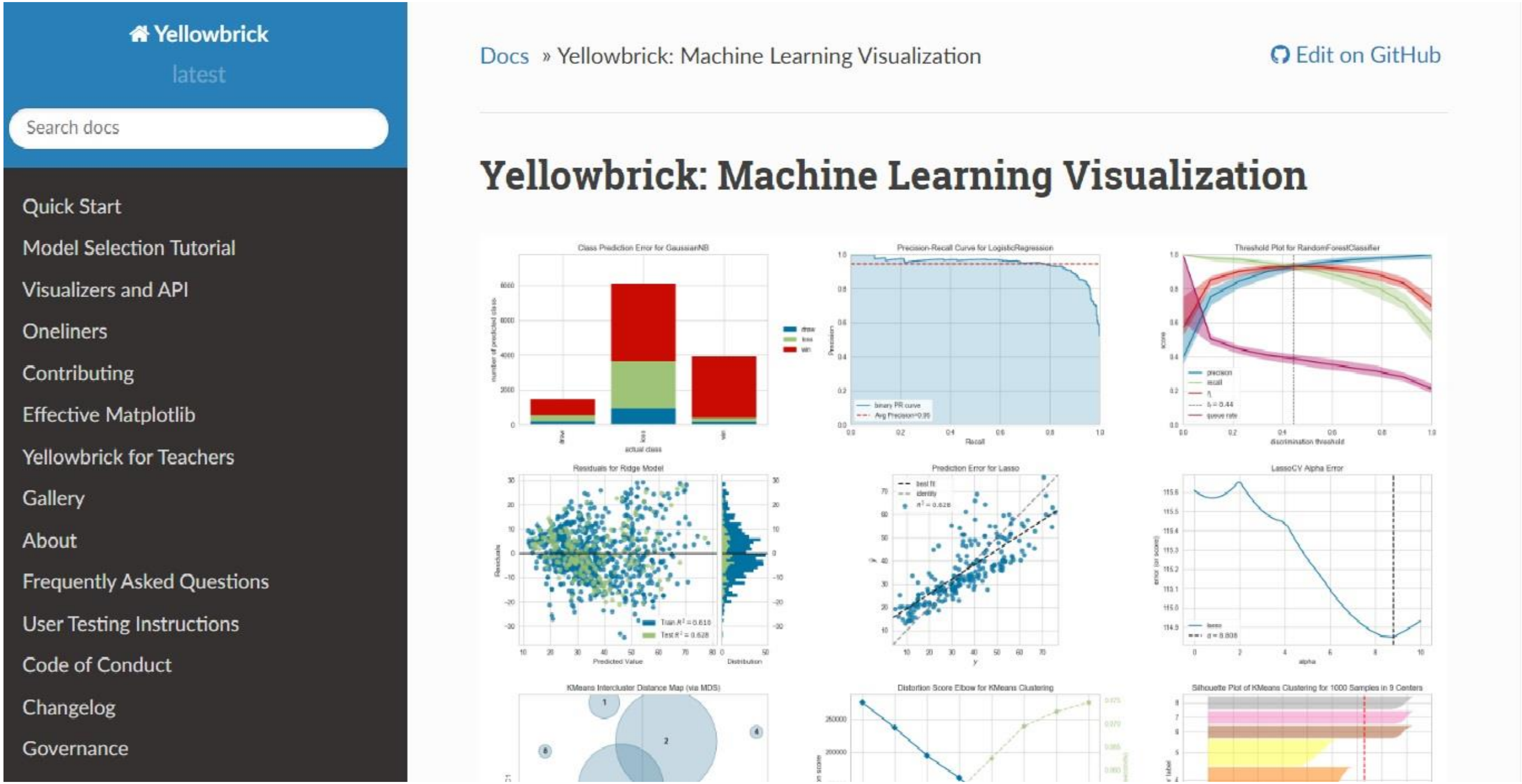
https://scikit-learn.org/stable/

Scikit-plot



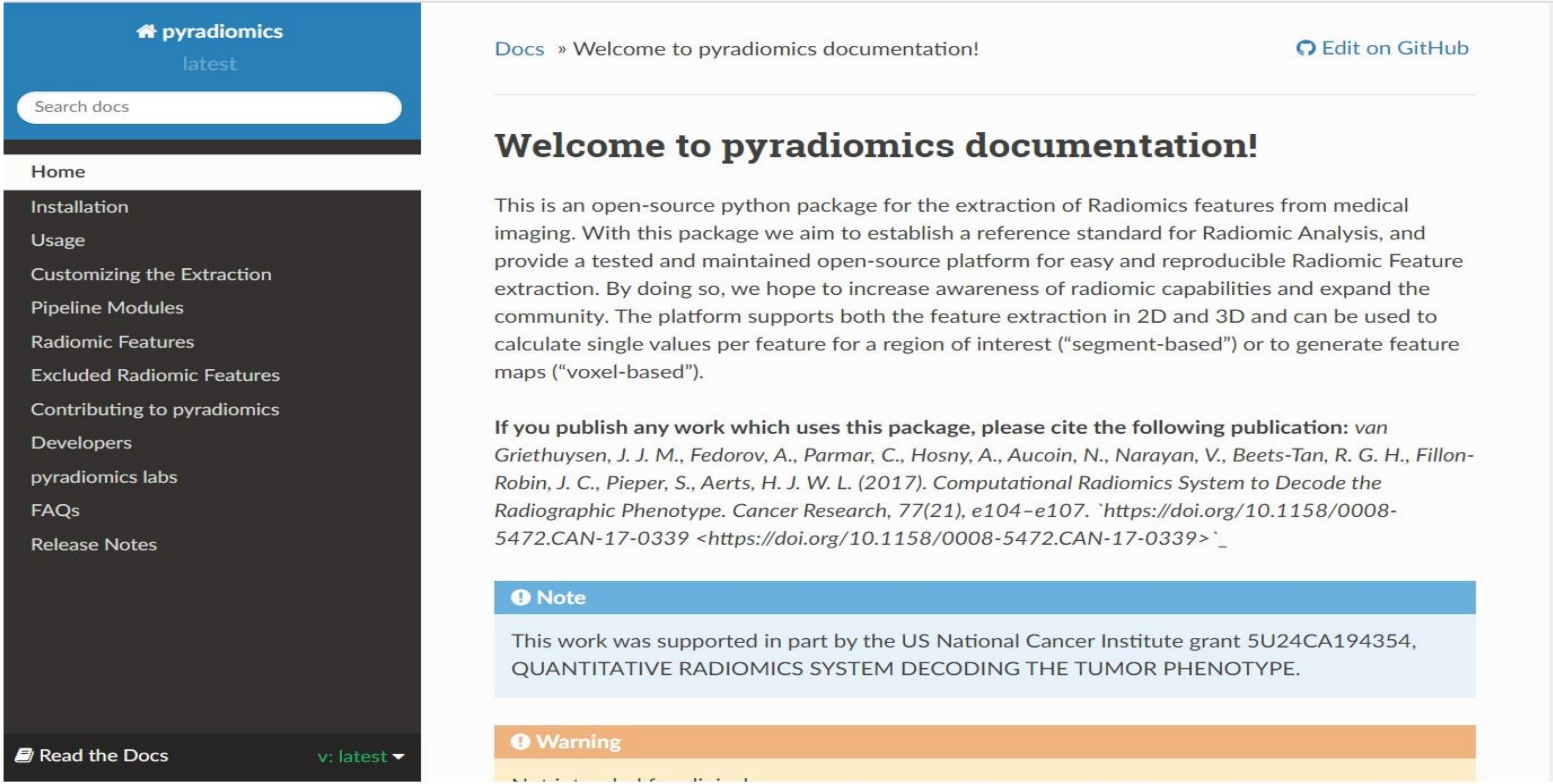
https://scikit-plot.readthedocs.io/en/stable/

Yellowbrick



https://www.scikit-yb.org/en/latest/

Pyradiomics



pyradiomics

latest

Search docs

Home

Installation

Usage

Customizing the Extraction

Pipeline Modules

Radiomic Features

Excluded Radiomic Features

Contributing to pyradiomics

Developers

pyradiomics labs

FAQs

Release Notes

Read the Docs v: latest

Docs » Welcome to pyradiomics documentation!

Edit on GitHub

Welcome to pyradiomics documentation!

This is an open-source python package for the extraction of Radiomics features from medical imaging. With this package we aim to establish a reference standard for Radiomic Analysis, and provide a tested and maintained open-source platform for easy and reproducible Radiomic Feature extraction. By doing so, we hope to increase awareness of radiomic capabilities and expand the community. The platform supports both the feature extraction in 2D and 3D and can be used to calculate single values per feature for a region of interest ("segment-based") or to generate feature maps ("voxel-based").

If you publish any work which uses this package, please cite the following publication: *van Griethuysen, J. J. M., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G. H., Fillon-Robin, J. C., Pieper, S., Aerts, H. J. W. L. (2017). Computational Radiomics System to Decode the Radiographic Phenotype. Cancer Research, 77(21), e104–e107. `https://doi.org/10.1158/0008-5472.CAN-17-0339 <https://doi.org/10.1158/0008-5472.CAN-17-0339>`_*

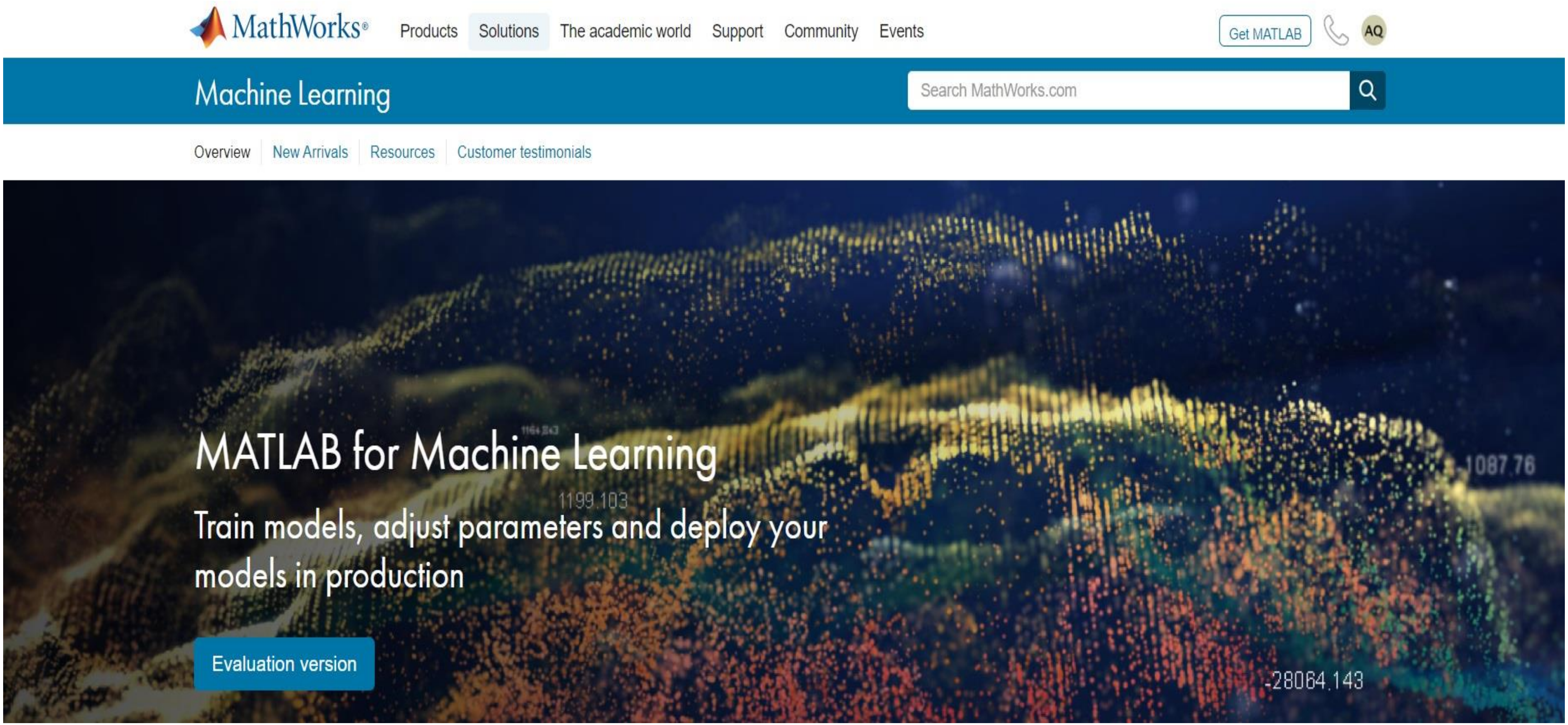
Note

This work was supported in part by the US National Cancer Institute grant 5U24CA194354, QUANTITATIVE RADIOMICS SYSTEM DECODING THE TUMOR PHENOTYPE.

Warning

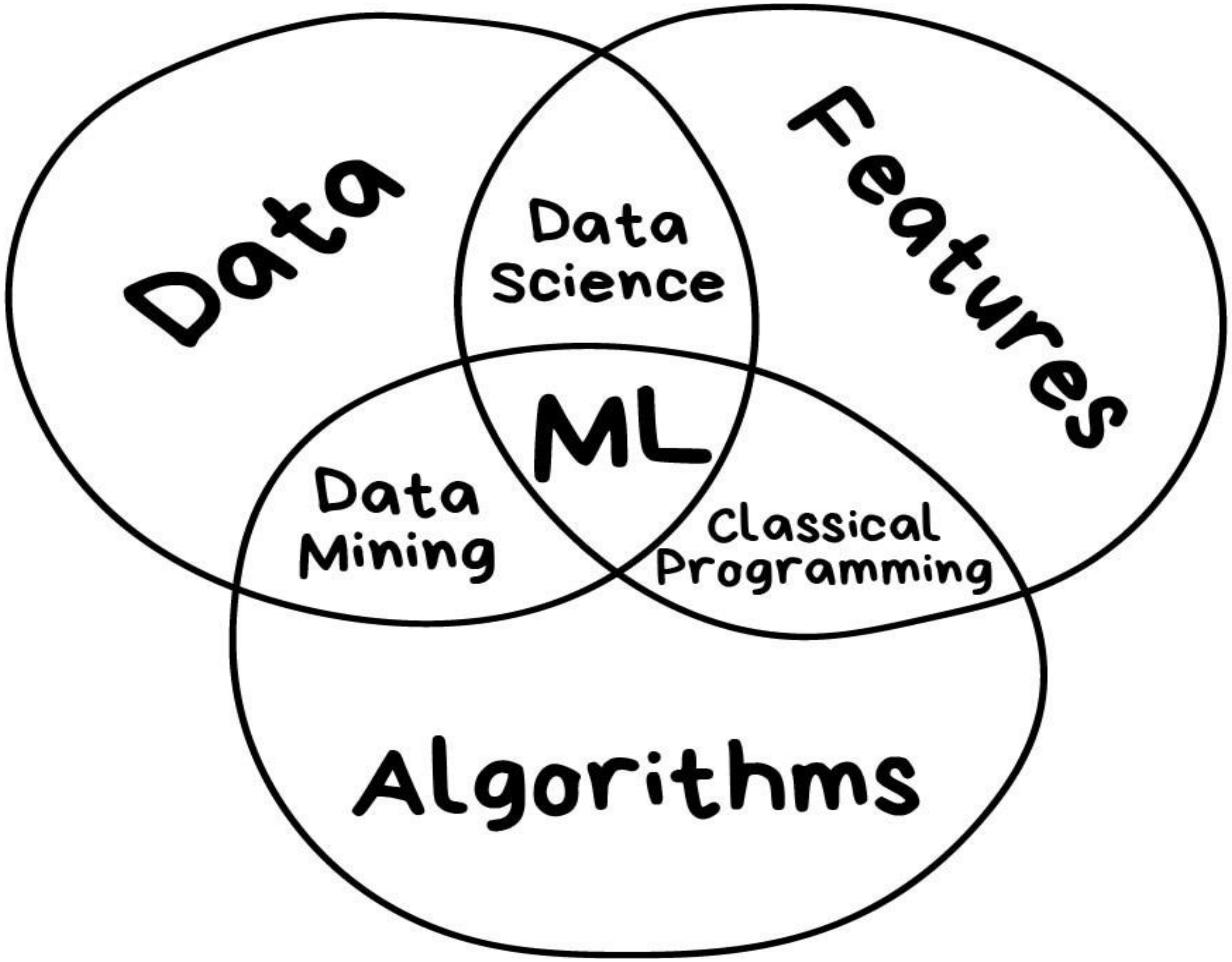
https://pyradiomics.readthedocs.io/en/latest/index.html

MATLAB



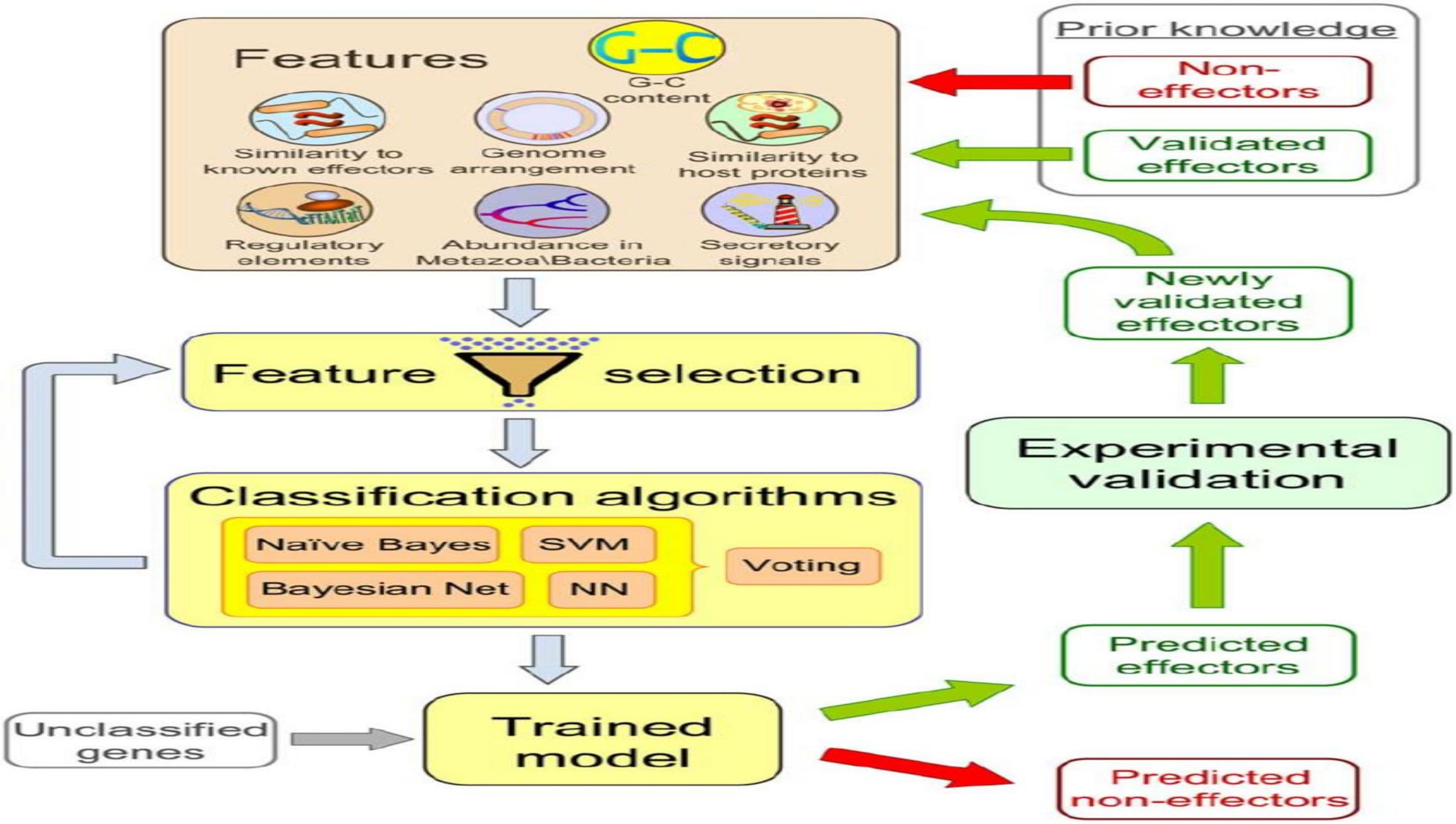
https://fr.mathworks.com/solutions/machine-learning.html

Components of ML



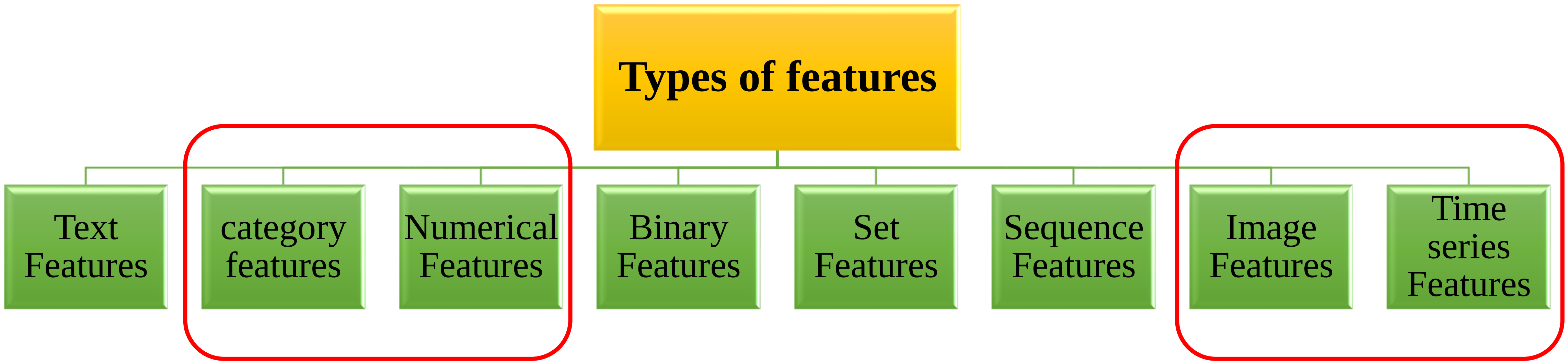
https://noeliagorod.com/2019/05/21/machine-learning-for-everyone-in-simple-words-with-real-world-examples-yes-again/

Design Steps of ML



https://noeliagorod.com/2019/05/21/machine-learning-for-everyone-in-simple-words-with-real-world-examples-yes-again/

Types of Features



Feature Methods

Feature

- Characteristic, attribute, quality, aspect of something

Feature detection

- Selects regions of an image that have unique content, such as **corners or blobs**

Feature extraction

- Transformation of an input data into a set of features involves computing a descriptor, which is typically done on regions centered around detected features.

Feature selection

- Selection of a subset of features out of the extracted features

Feature Selection/Extraction

Feature Selection / Extraction

Curse of Dimensionality:

- samples data needed to train classifier
- function grows exponentially with dimensions
- Overfitting and Generalization performance

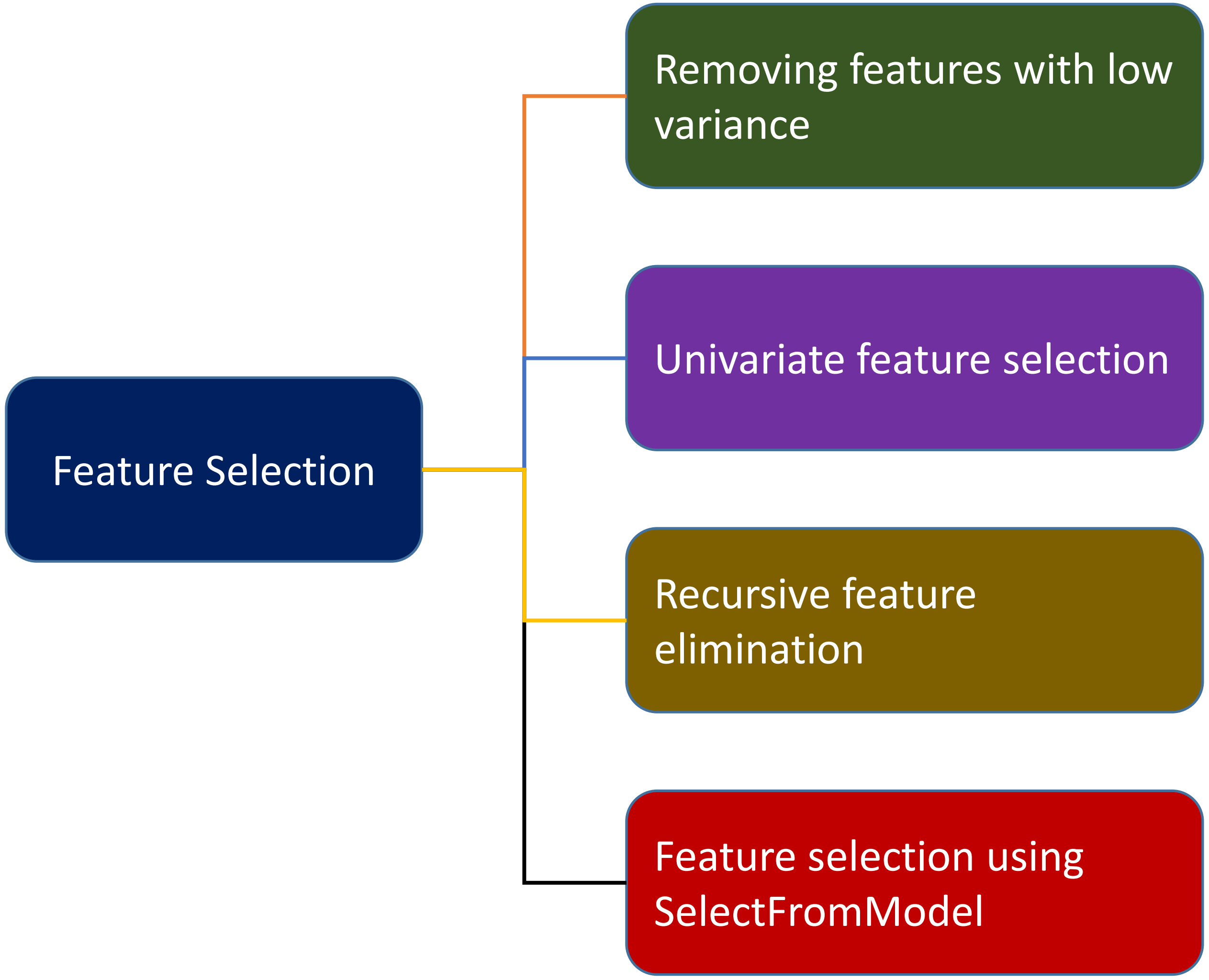
What features best characterize class?

- What words best characterize a document class
- Shape / texture / color

What features critical for performance?

- Subregions characterize protein function?

Feature Selection methods



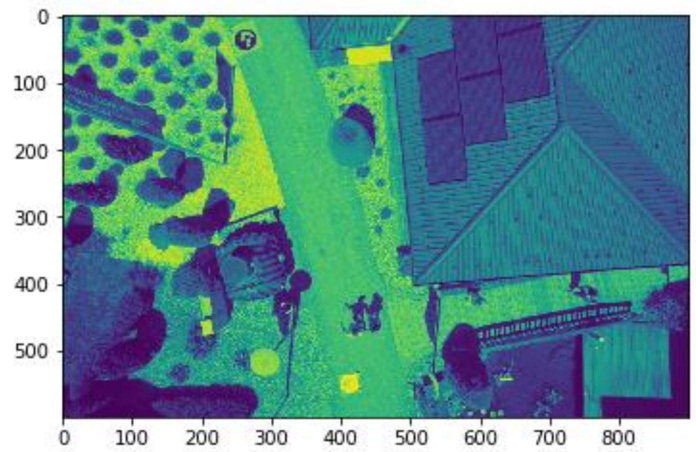
https://scikit-learn.org/stable/modules/feature_selection.html

First Order Statistical Features

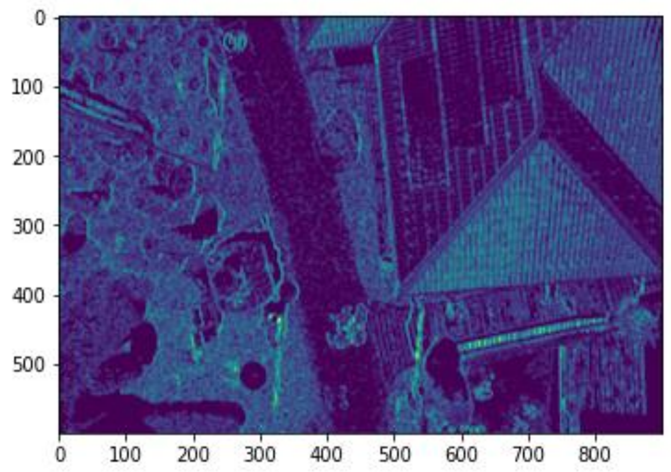
First Order Statistical Features	Mean(mean is the average value within the dataset)
	Variance(is a measure of the histogram width.it represents the deviation of gray levels from the means)
	Skewness(is a measure of the degree of histogram asymmetry around the mean)
	Kurtosis(is the measure of the histogram sharpness)

https://scikit-learn.org/stable/modules/feature_selection.html

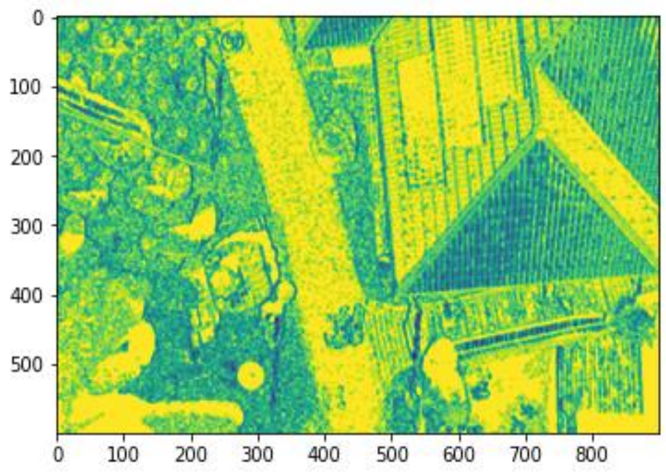
GLCM(grey level co-occurence matrix)



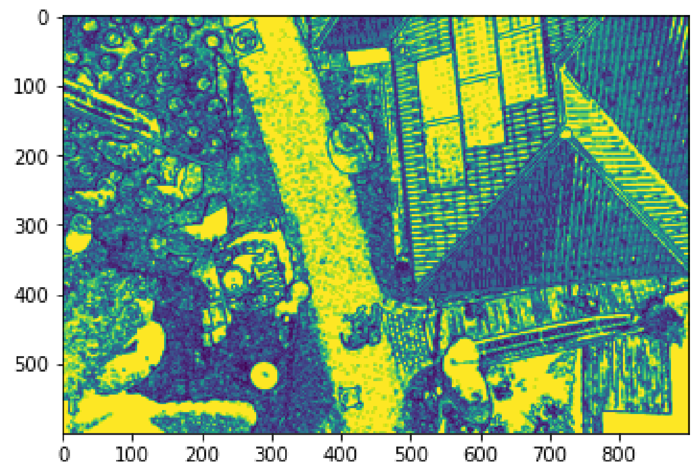
Main image



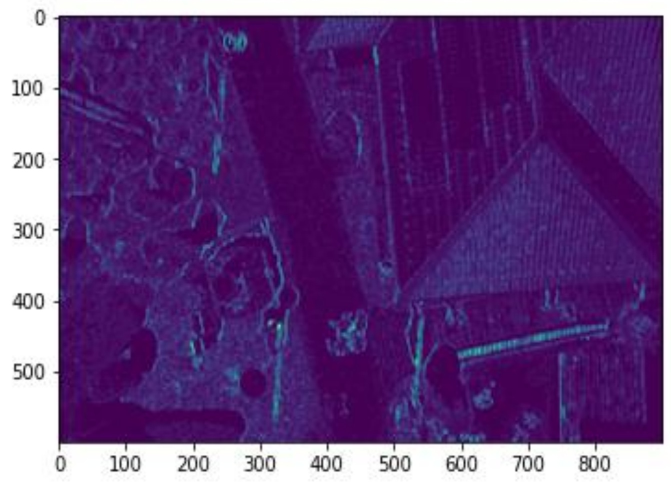
glcm_dissimilarity



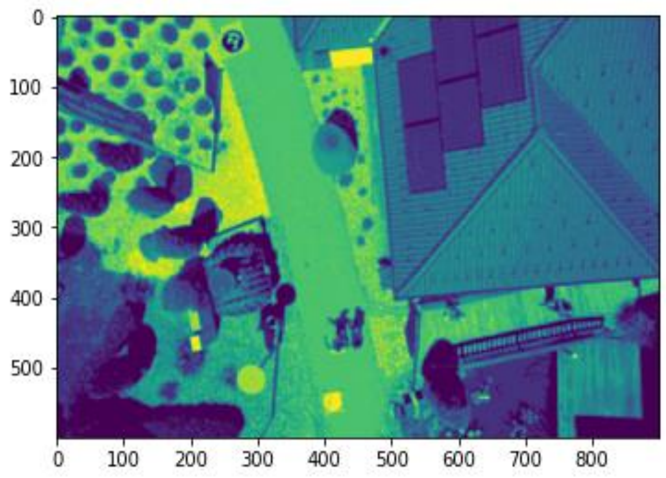
glcm_homogeneity



glcm_max



glcm_contrast

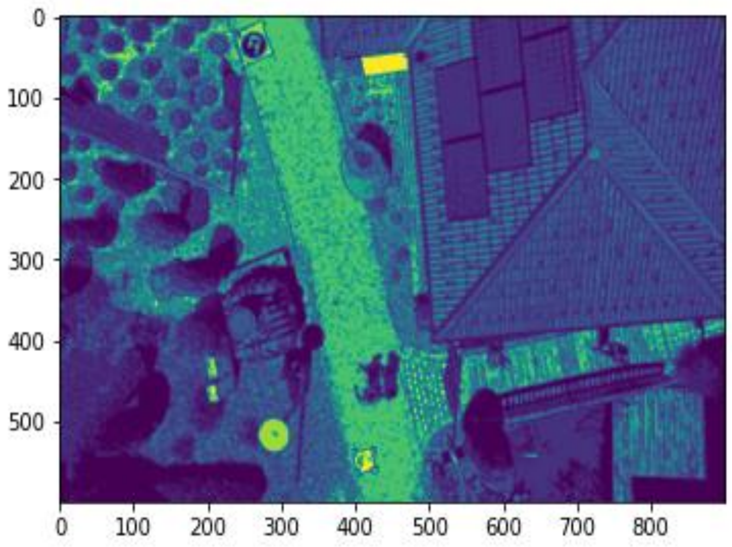


glcm_mean

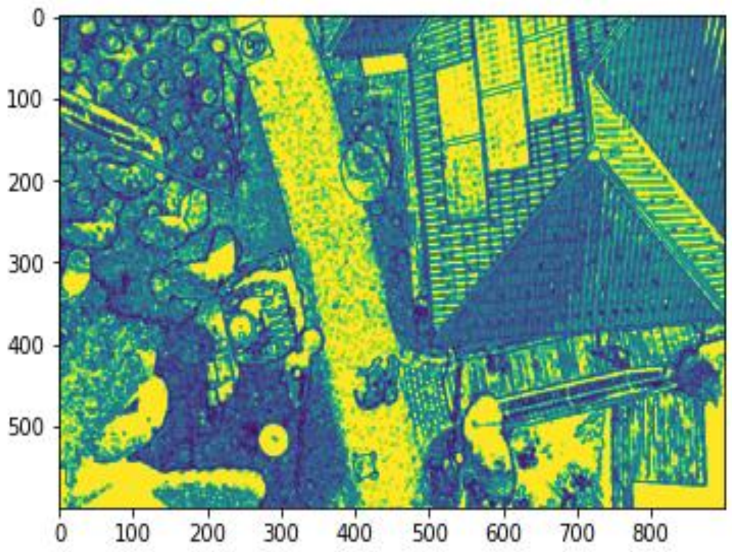
GLCM(grey level co-occurence matrix)

GLCM features

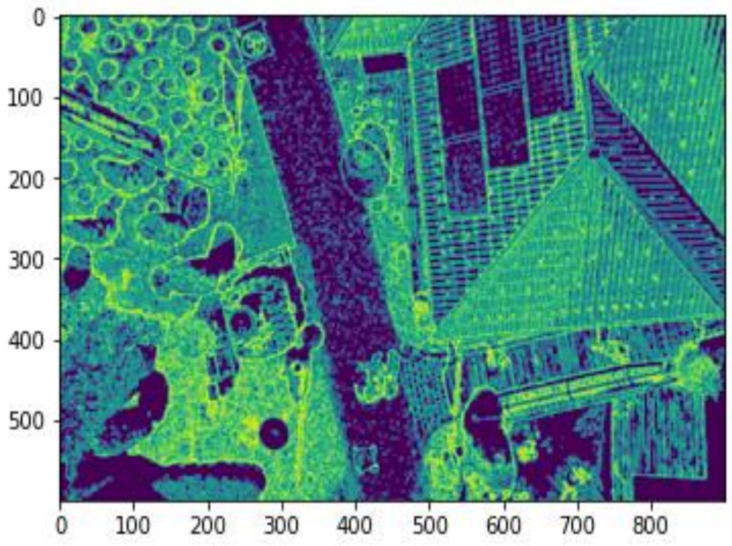
- Contrast
- Homogeneity
- Dissimilarity
- Std
- Mean
- Max
- ASM
- Entropy



glcm _std



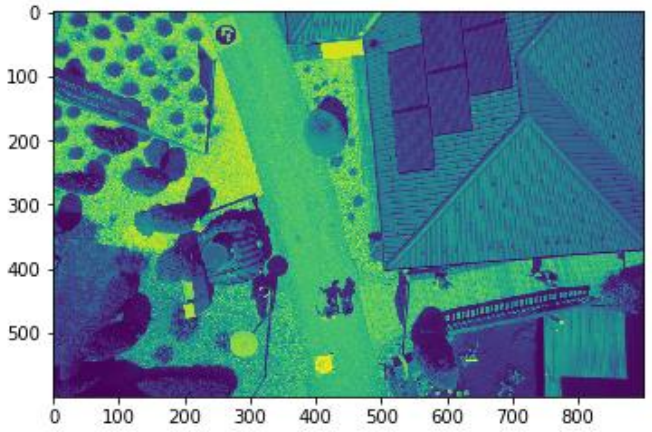
glcm_ASM



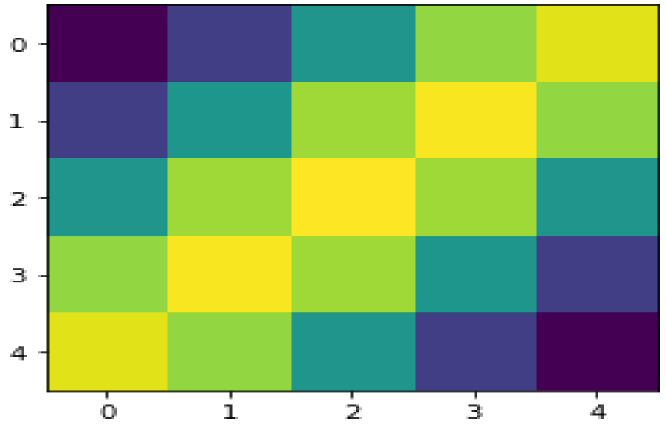
glcm _Entropy

Gabor Filter (Feature Extraction)

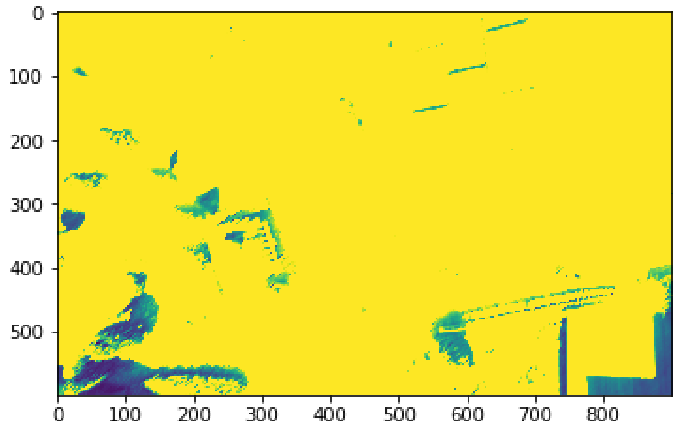
For image processing and computer vision, Gabor filters are generally used in texture analysis, edge detection, feature extraction, etc.
Gabor filters are special classes of bandpass filters, i.e., they allow a certain 'band' of frequencies and reject the others.



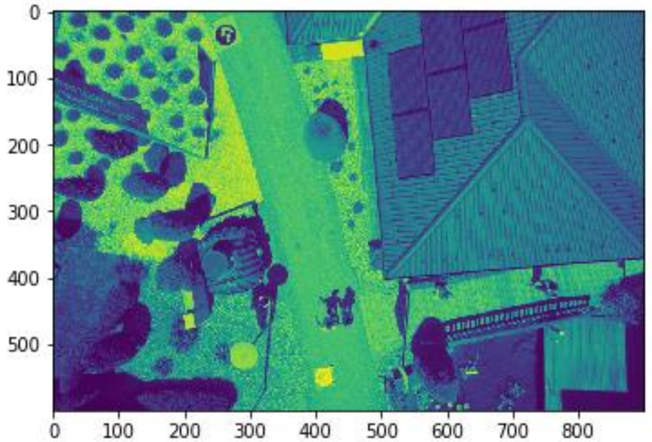
Main image



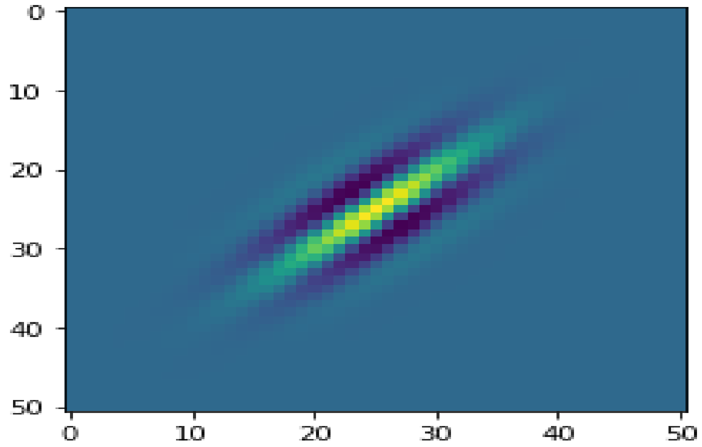
kernel function, sigma=3 k=5



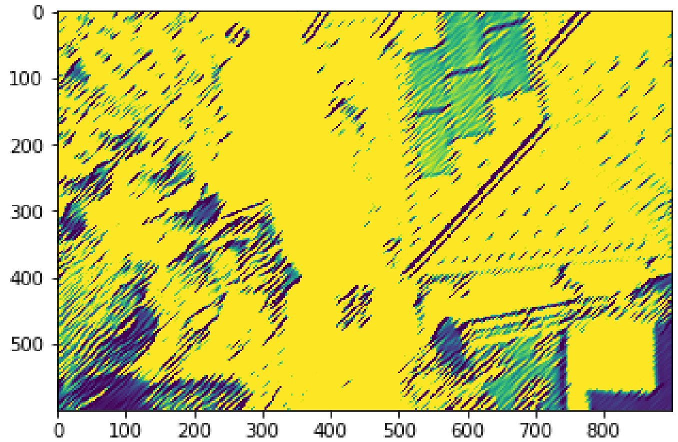
Output image



Main image

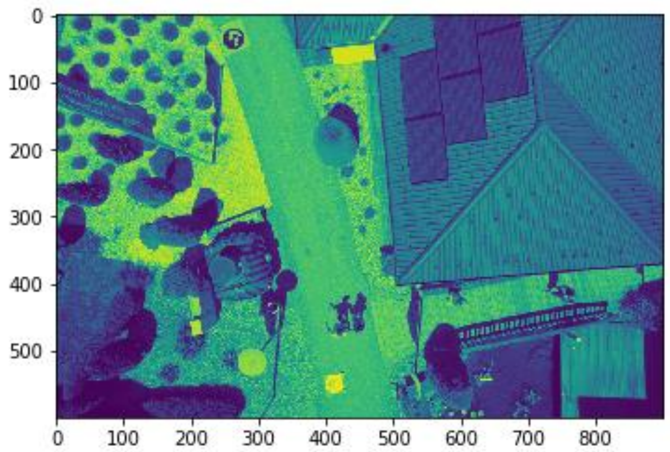


kernel function ,sigma=3 k=50

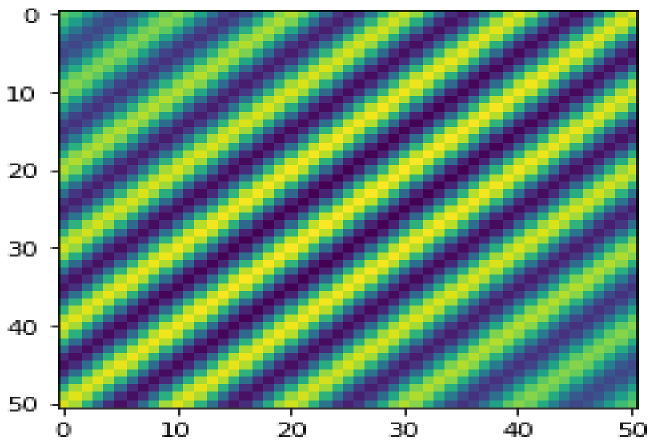


Output image

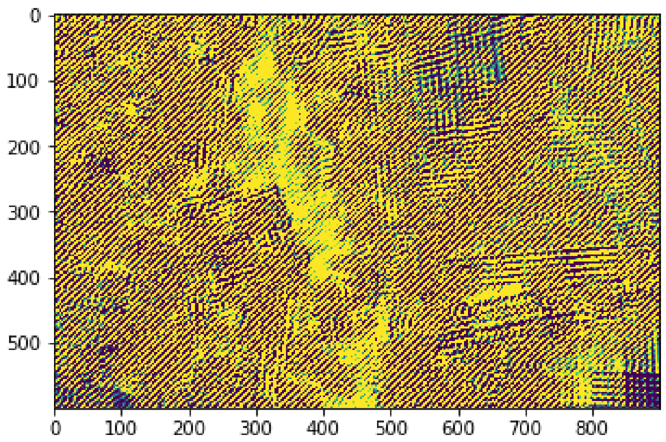
Gabor Filter (Feature Extraction)



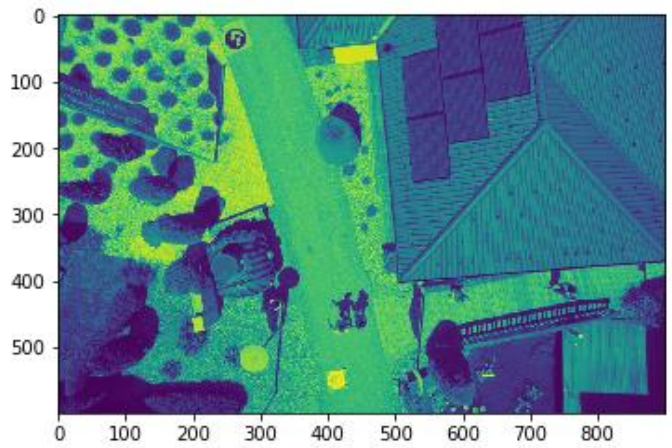
Main image



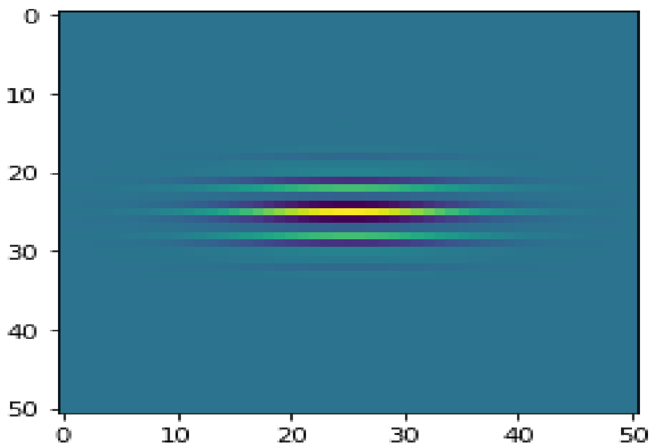
kernel function ,sigma=30,k=50



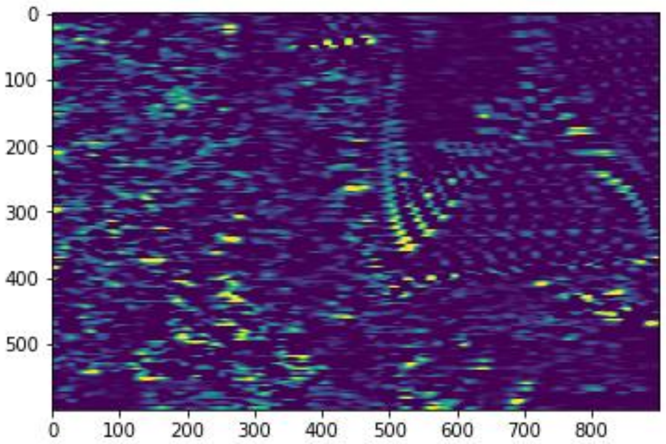
Output image



Main image



kernel function ,sigma=30,k=50



Output image

Detector and Descriptor (Feature Extraction)

Detector

- FAST
- SURF
- BRISK

Descriptor

- HOG
- LBP
- SURF
- BRISK

https://scikit-learn.org/stable/modules/feature_selection.html

Filtering Method (Feature Extraction)

Filtering Methods

- **Gabor filter**
- **Median filter**
- **Gaussian filter**
- **Hessian filter**
- **meijering**
- **sato**
- **frangi**

https://scikit-learn.org/stable/modules/feature_selection.html

Segmentation based on Image Features

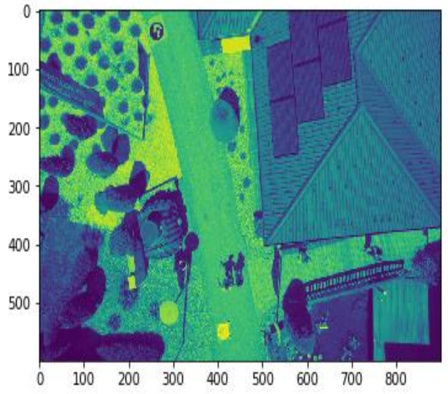
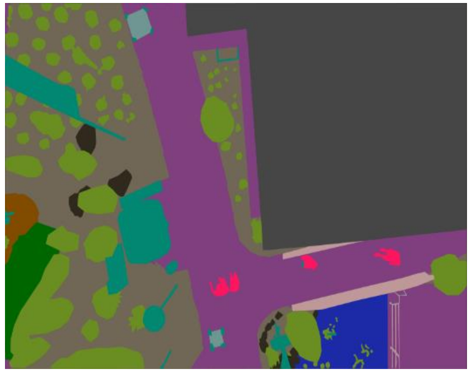


Image 300x300

1x90000



mask 300x300

1x90000

(pixels)	f1	f2								fn	Labels(depends on classes)
1											0
2											3
											0
											2
											0
											1
											1
											1
											1
90000											3

Feature Scaling or Normalization Methods

- **Feature scaling** is a method used to standardize the range of independent variables or **features** of data. In data processing, it is also known as data **normalization** and is generally performed during the data preprocessing step.

- **Why Scaling**
- Most of the times, your dataset will contain features highly varying in magnitudes, units, and range. But since most of the machine learning algorithms use Euclidian distance between two data points in their computations, this is a problem.
- If left alone, these algorithms only take in the magnitude of features neglecting the units. The results would vary greatly between different units, 5kg, and 5000gms. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.

Feature Scaling or Normalization Methods

- In this lecture we explore 3 methods of feature scaling that are implemented in scikit-learn:

1 StandardScaler	2 MinMaxScaler	3 RobustScaler	4 Normalizer

1. Standard Scaler

- The StandardScaler assumes your data is normally distributed within each feature and will scale them such that the distribution is now centered around 0, with a standard deviation of 1.
- The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

$$z = \frac{x - \mu}{\sigma}$$

μ=Mean

$\sigma = standard\ deviation$

Feature Scaling or Normalization Methods

2. MinMaxScaler

- The MinMaxScaler is the probably the most famous scaling algorithm, and It essentially shrinks the range such that the range is now between 0 and 1 (or -1 to 1 if there are negative values).
- This scaler works better for cases in which the standard scaler might not work so well. If the distribution is not Gaussian or the standard deviation is very small, the min-max scaler works better.
- However, it is sensitive to outliers, so if there are outliers in the data, you might want to consider the Robust Scaler below.follows the following formula for each feature:

$$X_{new} = \frac{X_i - \min(X)}{\max(X) - \min(X)}$$

Syntax :

```
from sklearn import preprocessing
scaler = preprocessing.MinMaxScaler()
scaled_df = scaler.fit_transform(df)
```

Feature Scaling or Normalization Methods

3. RobustScaler

- The RobustScaler uses a similar method to the Min-Max scaler but it instead uses the interquartile range, rather than the min-max, so that it is robust to outliers. Therefore it follows the formula:
- xi–Q1(x)/Q3(x)–Q1(x)
- For each feature. Of course, this means it is using the less of the data for scaling so it’s more suitable for when there are outliers in the data.

Syntax:

```
from sklearn import preprocessing
scaler = preprocessing.RobustScaler()
robust_scaled_df = scaler.fit_transform(x)
```

Feature Scaling or Normalization Methods

4. Normalizer

- The normalizer scales each value by dividing each value by its magnitude in n-dimensional space for n number of features. Say your features were x, y, and z Cartesian co-ordinates your scaled value for x would be:
- Each point is now within 1 unit of the origin on this Cartesian coordinate system.
- X=xi/np.sqrt(xi**2+yi**2+zi**2

Syntax :

```
from sklearn import preprocessing
scaler = preprocessing.Normalizer()
scaled_df = scaler.fit_transform(df)
```

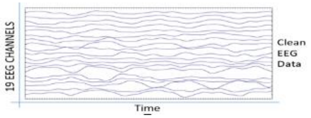
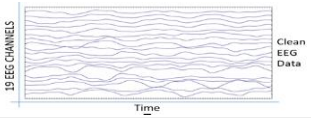
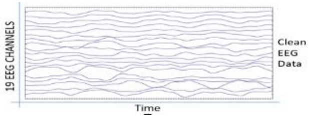
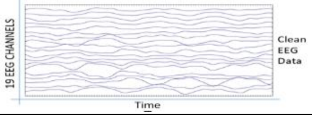
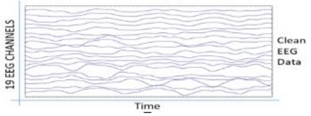
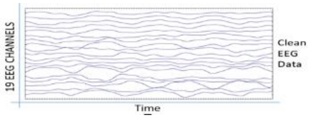
Feature Scaling or Normalization Methods

- **Where and when to Scale**
- **Rule of thumb I follow here is an algorithm that computes distance or assumes normality, scales your features!!!**

Some examples of algorithms where feature scaling matters are:

1. k-nearest neighbors with a Euclidean distance measure is sensitive to magnitudes and hence should be scaled for all features to weigh in equally.
2. Scaling is critical while performing Principal Component Analysis(PCA). PCA tries to get the features with maximum variance and the variance is high for high magnitude features. This skews the PCA towards high magnitude features.
3. We can speed up gradient descent by scaling. This is because θ will descend quickly on small ranges and slowly on large ranges, and so will oscillate inefficiently down to the optimum when the variables are very uneven.
4. Tree-based models are not distance-based models and can handle varying ranges of features. Hence, Scaling is not required while modeling trees.
5. Algorithms like Linear Discriminant Analysis(LDA), Naive Bayes is by design equipped to handle this and gives weights to the features accordingly. Performing a feature scaling in these algorithms may not have much effect.

Basic Dimensions in 1D-CNN for Training and Testing

No of samples	Classes	Dimension	Labels
1	Patient1	19x256 	0
2	Patient1	19x256 	0
	Patient1	----	0
	Patient1	---	0
50	Patient1	19x256 	0
51	Patient2	19x256 	1
	Patient2	19x256 	1
	Patient2	----	1
99	Patient2	----	1
100	Patient2	19x256 	1

Total samples x timestep x features

100 x 256 x 19

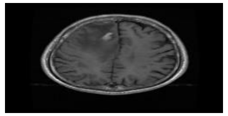
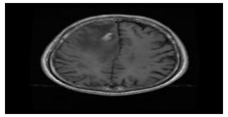
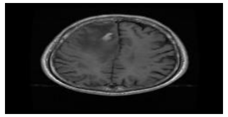
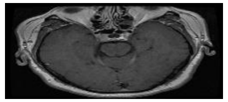
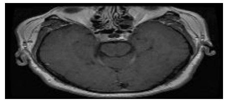
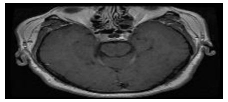
Training samples

80 x 256 x 19

Testing samples

20 x 256 x 19

Basic Dimensions in 2D-CNN for Training and Testing

No of samples	Classes	Dimension	Labels
1	Patient1	224x224x3 	0
2	Patient1	224x224x3 	0
	Patient1	----	0
	Patient1	---	0
50	Patient1	224x224x3 	0
51	Patient2	224x224x3 	1
	Patient2	224x224x3 	1
	Patient2	----	1
99	Patient2	----	1
100	Patient2	224x224x3 	1

Total samples x row x cols x channels

100 x 224x224x3

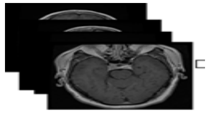
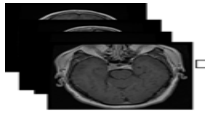
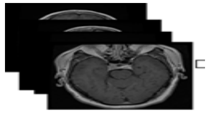
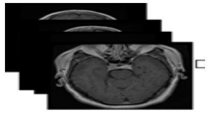
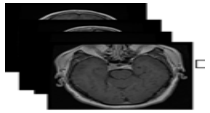
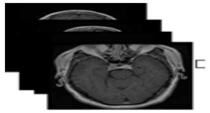
Training samples

80x224x224x3

Testing samples

20 x 224x224x3

Basic Dimensions in 3D-CNN for Training and Testing

No of samples	Classes	Dimension	Labels
1	Patient1	30x224x224x3 	0
2	Patient1	30x224x224x3 	0
	Patient1	----	0
	Patient1	---	0
50	Patient1	30x224x224x3 	0
51	Patient2	30x224x224x3 	1
	Patient2	30x224x224x3 	1
	Patient2	----	1
99	Patient2	----	1
100	Patient2	30x224x224x3 	1

Total samples x slices rows x cols x channels

100 x30x 224x224x3

Training samples

80x30x224x224x3

Testing samples

20 x30x 224x224x3

Data pre-processing and Acquisitions Libraries

Python
https://www.w3schools.com/python/python_intro.asp
NumPy
https://numpy.org/
Pandas
https://pandas.pydata.org/
Matplotlib
https://matplotlib.org/
ITK(segmentation and registration toolkit)
https://itk.org/
ITK-snap for 3d Modeling and visualization
http://www.itksnap.org/pmwiki/pmwiki.php
NiBabel processing nifty (Neuroimaging Informatics Technology Initiative) image
https://nipy.org/nibabel/gettingstarted.html

Libraries used for Deep Learning and Machine Learning Implementation

Machine Learning library

Scikit-Learn
https://scikit-learn.org/stable/
Scikit-learn plot
https://scikit-plot.readthedocs.io/en/stable/Quickstart.html
Yellowbrick (visualization)
https://www.scikit-yb.org/en/latest/
MATLAB
https://fr.mathworks.com/products/matlab.html

Deep Learning libraries

Tensorflow
https://www.tensorflow.org/tutorials
Pytorch
https://pytorch.org/docs/stable/index.html
Keras
https://keras.io/
MATLAB
https://fr.mathworks.com/products/matlab.html

Medical Image Datasets and Competitions

TABLE 5: Examples of popular databases used by Medical Image Analysis techniques that exploit Deep Learning.

Sr.	Database	Anatomic site	Image modality	Main task	Patients/Images
1	ILD [254]	Lung	CT	Classification	120 patients
2	LIDC-IDRI [90]	Lung	CT	Detection	1,018 patients
3	ADNI [58]	Brain	MRI	Classification	¿800 patients
4	MURA [111]	Musculoskeletal	X-ray	Detection	40,561 images
5	TCIA	Multiple	Multiple	Multiple	¿35,000 patients
6	BRATS [274]	Brain	MRI	Segmentation	-
7	DDSM [275]	Breast	Mamography	Segmentation	2,620 patients
8	MESSIDOR-2 [276], [277]	Eye	OCT	Classification	1,748 images
9	ChestX-ray14 [278]	Chest	X-ray	Multiple	¿100,000 images
10	ACDC 2017	Brain	MRI	Classification	150 patients
11	2015 MICCAI Gland Challenge	Glands	Histopathology	Segmentation	165 images
12	OAI	Knee	X-ray, MRI	Multiple	4,796 patients
13	DRIVE [128], [279]	Eye	SLO	Segmentation	400 patients
14	STARE [130]	Eye	SLO	Segmentation	400 images
15	CHASEDB1 [129]	Eye	SLO	Segmentation	28 images
16	OASIS-3 [57], [280], [281], [282], [283], [284], [117]	Brain	MRI, PET	Segmentation	1,098 patients
17	MIAS [285]	Breast	Mammography	Classification	322 patients
18	ISLES 2018	Brain	MRI	Segmentation	103 patients
19	HVSMR 2018 [286]	Heart	CMR	Segmentation	4 patients
20	CAMELYON17 [113]	Breast	WSI	Segmentation	899 images
21	ISIC 2018	Skin	JPEG	Detection	2,600 images
22	OpenNeuro	Brain	Multiple	Classification	4,718 patients
23	ABIDE	Brain	MRI	Disease Diagnosis	1,114 patients
24	INbreast [243]	Breast	Mammography	Detection/Classification	410 images

Source: https://arxiv.org/pdf/1902.05655.pdf

Medical Image Datasets and Competitions

Table 5
A short list of medical imaging competitions.

Name	Summary	Link
Grand-Challenges	Grand challenges in biomedical image analysis. Hosts and lists a large number of competitions	https://grand-challenge.org/
RSNA Pneumonia Detection Challenge	Automatically locate lung opacities on chest radiographs	https://www.kaggle.com/c/rsna-pneumonia-detection-challenge
HVSMR 2016	Segment the blood pool and myocardium from a 3D cardio-vascular magnetic resonance image	http://segchd.csail.mit.edu/
ISLES 2018	Ischemic Stroke Lesion Segmentation 2018. The goal is to segment stroke lesions based on acute CT perfusion data.	http://www.isles-challenge.org/
BraTS 2018	Multimodal Brain Tumor Segmentation. The goal is to segment brain tumors in multimodal MRI scans.	http://www.med.upenn.edu/sbia/brats2018.html
CAMELYON17	The goal is to develop algorithms for automated detection and classification of breast cancer metastases in whole-slide images of histological lymph node sections.	https://camelyon17.grand-challenge.org/Home
ISIC 2018	Skin Lesion Analysis Towards Melanoma Detection	https://challenge2018.isic-archive.com/
Kaggle's 2018 Data Science Bowl	Spot Nuclei. Speed Cures.	https://www.kaggle.com/c/data-science-bowl-2018
Kaggle's 2017 Data Science Bowl	Turning Machine Intelligence Against Lung Cancer	https://www.kaggle.com/c/data-science-bowl-2017
Kaggle's 2016 Data Science Bowl	Transforming How We Diagnose Heart Disease	https://www.kaggle.com/c/second-annual-data-science-bowl
MURA	Determine whether a bone X-ray is normal or abnormal	https://stanfordmlgroup.github.io/competitions/mura/

Source: https://www.sciencedirect.com/science/article/pii/S0939388918301181

Machine Learning Applications

Information Fusion 52 (2019) 1–12

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

ELSEVIER

Full Length Article

Ensembles for feature selection: A review and future trends

Verónica Bolón-Canedo, Amparo Alonso-Betanzos*

CITIC Research Centre on Information and Communication Technology, University of A Coruña, Campus de Elviña s/n 15071, A Coruña, Spain

A R T I C L E   I N F O

Keywords:
Ensemble learning
Feature selection

A B S T R A C T

Ensemble learning is a prolific field in Machine Learning since it is based on the assumption that combining the output of multiple models is better than using a single model, and it usually provides good results. Normally, it has been commonly employed for classification, but it can be used to improve other disciplines such as feature selection. Feature selection consists of selecting the relevant features for a problem and discard those irrelevant or redundant, with the main goal of improving classification accuracy. In this work, we provide the reader with the basic concepts necessary to build an ensemble for feature selection, as well as reviewing the up-to-date advances and commenting on the future trends that are still to be faced.

Machine Learning Applications

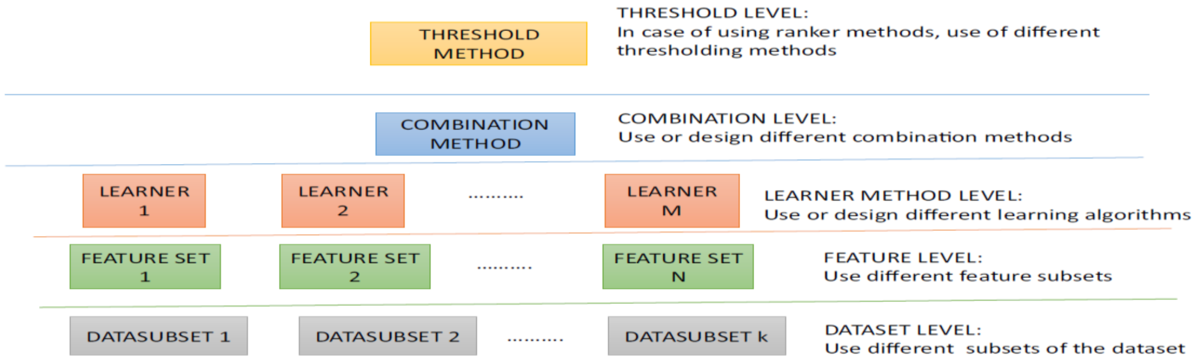


Fig. 1. The different levels than can be varied in an ensemble for feature selection.

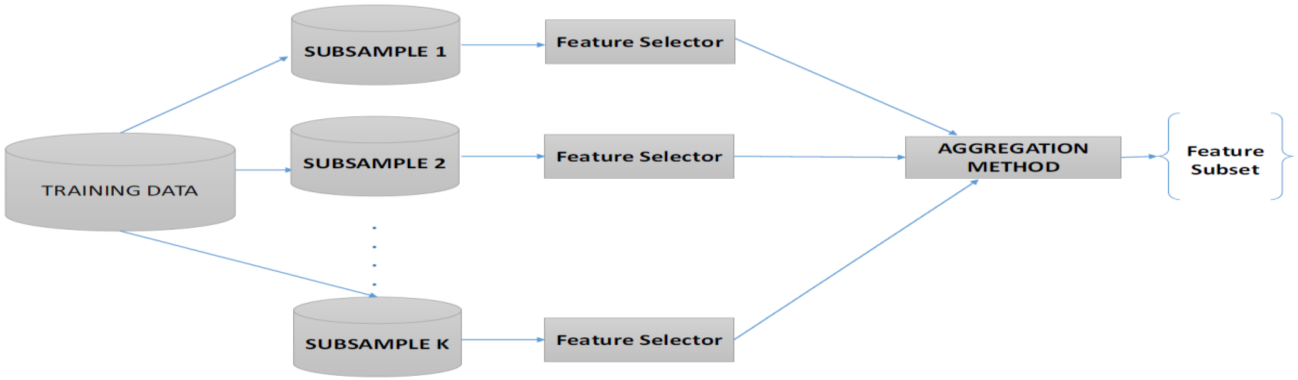


Fig. 2. A scheme of the Homogeneous approach.

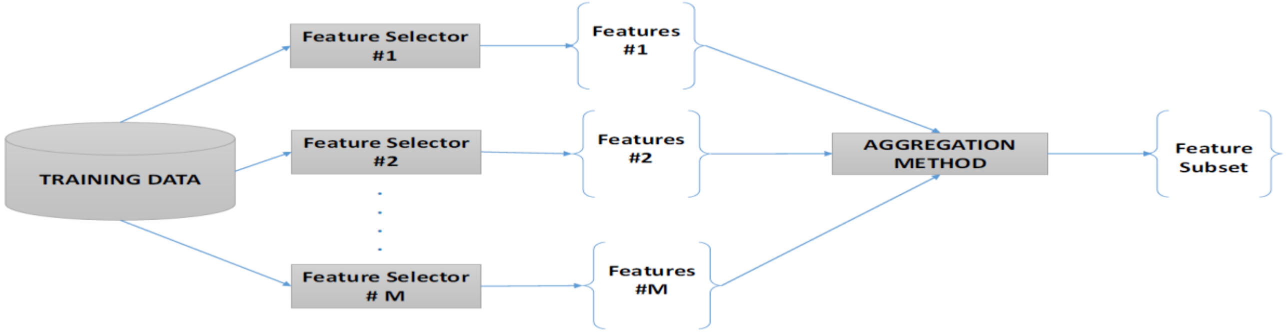


Fig. 3. A scheme of the Heterogeneous approach.

Information Fusion 52 (2019) 1–12

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

ELSEVIER

Full Length Article

Ensembles for feature selection: A review and future trends

Verónica Bolón-Canedo, Amparo Alonso-Betanzos*

CITIC Research Centre on Information and Communication Technology, University of A Coruña, Campus de Elviña s/n 15071, A Coruña, Spain

ARTICLE INFO

Keywords:
Ensemble learning
Feature selection

ABSTRACT

Ensemble learning is a prolific field in Machine Learning since it is based on the assumption that combining the output of multiple models is better than using a single model, and it usually provides good results. Normally, it has been commonly employed for classification, but it can be used to improve other disciplines such as feature selection. Feature selection consists of selecting the relevant features for a problem and discard those irrelevant or redundant, with the main goal of improving classification accuracy. In this work, we provide the reader with the basic concepts necessary to build an ensemble for feature selection, as well as reviewing the up-to-date advances and commenting on the future trends that are still to be faced.

Machine Learning Applications

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

ELSEVIER

INFORMATION FUSION

Full Length Article

Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities

Check for updates

Marinka Zitnik [a,*], Francis Nguyen [b,c], Bo Wang [d], Jure Leskovec [a,e,*], Anna Goldenberg [f,g,h,*], Michael M. Hoffman [b,c,g,h,*]

[a] *Department of Computer Science, Stanford University, Stanford, CA, USA*
[b] *Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada*
[c] *Princess Margaret Cancer Centre, Toronto, ON, Canada*
[d] *Hikvision Research Institute, Santa Clara, CA, USA*
[e] *Chan Zuckerberg Biohub, San Francisco, CA, USA*
[f] *Genetics & Genome Biology, SickKids Research Institute, Toronto, ON, Canada*
[g] *Department of Computer Science, University of Toronto, Toronto, ON, Canada*
[h] *Vector Institute, Toronto, ON, Canada*

ARTICLE INFO

Keywords:
Computational biology
Personalized medicine
Systems biology
Heterogeneous data
Machine learning

ABSTRACT

New technologies have enabled the investigation of biology and human health at an unprecedented scale and in multiple dimensions. These dimensions include a myriad of properties describing genome, epigenome, transcriptome, microbiome, phenotype, and lifestyle. No single data type, however, can capture the complexity of all the factors relevant to understanding a phenomenon such as a disease. Integrative methods that combine data from multiple technologies have thus emerged as critical statistical and computational approaches. The key challenge in developing such approaches is the identification of effective models to provide a comprehensive and relevant systems view. An ideal method can answer a biological or medical question, identifying important features and predicting outcomes, by harnessing heterogeneous data across several dimensions of biological variation. In this Review, we describe the principles of data integration and discuss current methods and available implementations. We provide examples of successful data integration in biology and medicine. Finally, we discuss current challenges in biomedical integrative methods and our perspective on the future development of the field.

Machine Learning Applications

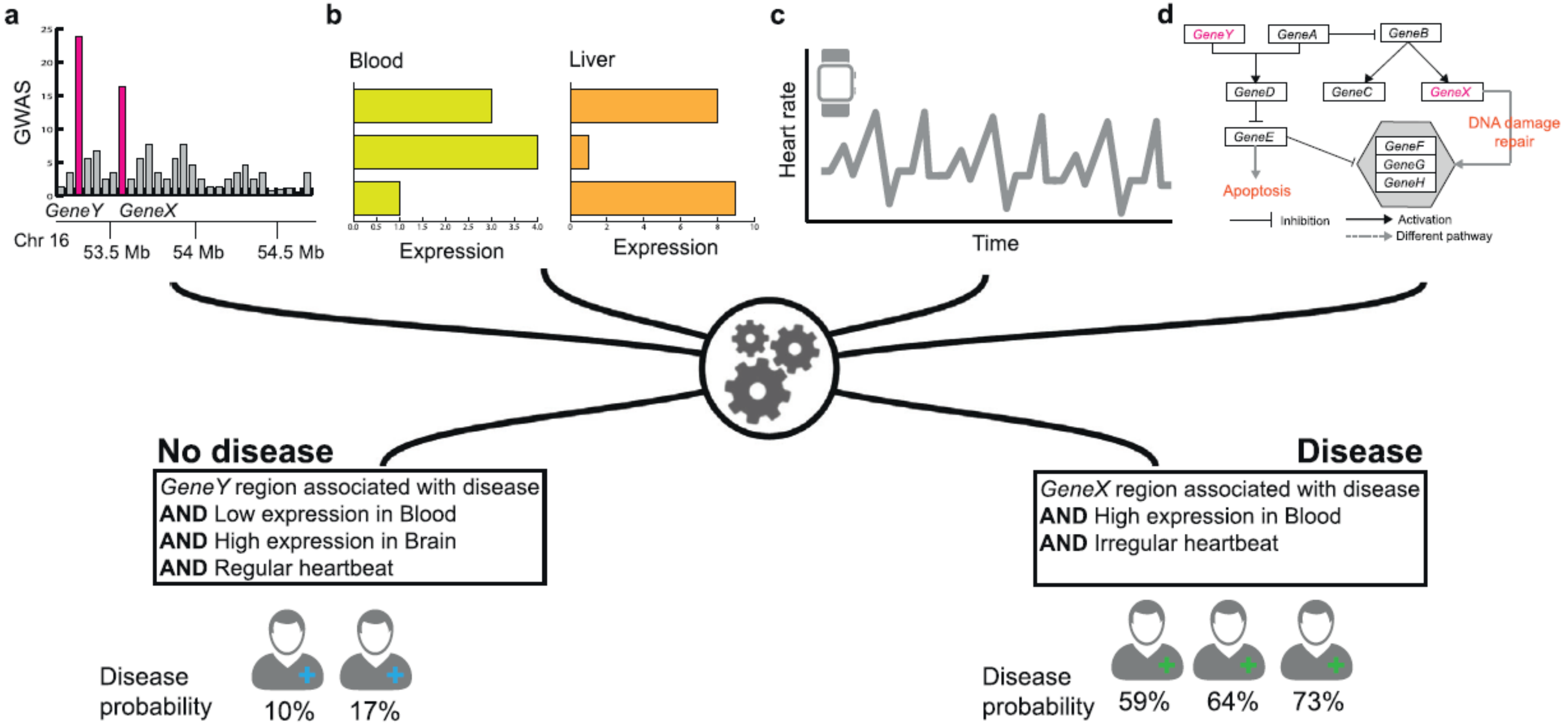
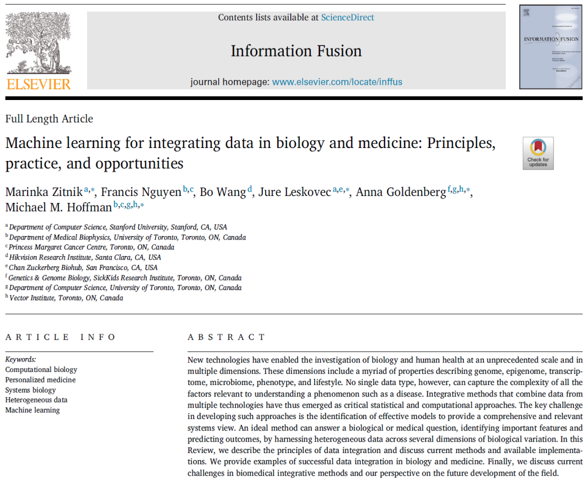


Fig. 1. The importance of data integration in biomedicine. Considering variation in only a single data type can miss many important patterns that can only be observed by considering multiple levels of biomedical data. Shown is a hypothetical example using disease diagnostics as a point of interest. When a new patient arrives to the clinic, (a) domain experts sequence the patient's genome and compare it with a database to identify mutations and disease-causing genes, (b) perform laboratory tests using tissue samples, and (c) process information about the patient's behavior and lifestyle. (d) The patient's genomic, transcriptomic, and lifestyle information is combined with curated databases of biomedical knowledge (*e.g.*, disease and metabolic pathways). Finally, a machine learning algorithm predicts probability that the patient will develop a particular disease in near future. To make accurate prediction, the machine learning model needs to use many different types of data. This example illustrates that accurate prediction can only be made by analyzing multiple types of patient's data.



Contents lists available at ScienceDirect
Information Fusion
journal homepage: www.elsevier.com/locate/inffus
ELSEVIER
Full Length Article

Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities

Marinka Zitnik[a,*], Francis Nguyen[b,c], Bo Wang[d], Jure Leskovec[a,e,*], Anna Goldenberg[f,g,h,*], Michael M. Hoffman[b,c,g,h,*]

[a] Department of Computer Science, Stanford University, Stanford, CA, USA
[b] Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada
[c] Princess Margaret Cancer Centre, Toronto, ON, Canada
[d] Hikvision Research Institute, Santa Clara, CA, USA
[e] Chan Zuckerberg Biohub, San Francisco, CA, USA
[f] Genetics & Genome Biology, SickKids Research Institute, Toronto, ON, Canada
[g] Department of Computer Science, University of Toronto, Toronto, ON, Canada
[h] Vector Institute, Toronto, ON, Canada

ARTICLE INFO

Keywords:
Computational biology
Personalized medicine
Systems biology
Heterogeneous data
Machine learning

ABSTRACT

New technologies have enabled the investigation of biology and human health at an unprecedented scale and in multiple dimensions. These dimensions include a myriad of properties describing genome, epigenome, transcriptome, microbiome, phenotype, and lifestyle. No single data type, however, can capture the complexity of all the factors relevant to understanding a phenomenon such as a disease. Integrative methods that combine data from multiple technologies have thus emerged as critical statistical and computational approaches. The key challenge in developing such approaches is the identification of effective models to provide a comprehensive and relevant systems view. An ideal method can answer a biological or medical question, identifying important features and predicting outcomes, by harnessing heterogeneous data across several dimensions of biological variation. In this Review, we describe the principles of data integration and discuss current methods and available implementations. We provide examples of successful data integration in biology and medicine. Finally, we discuss current challenges in biomedical integrative methods and our perspective on the future development of the field.

Machine Learning Applications

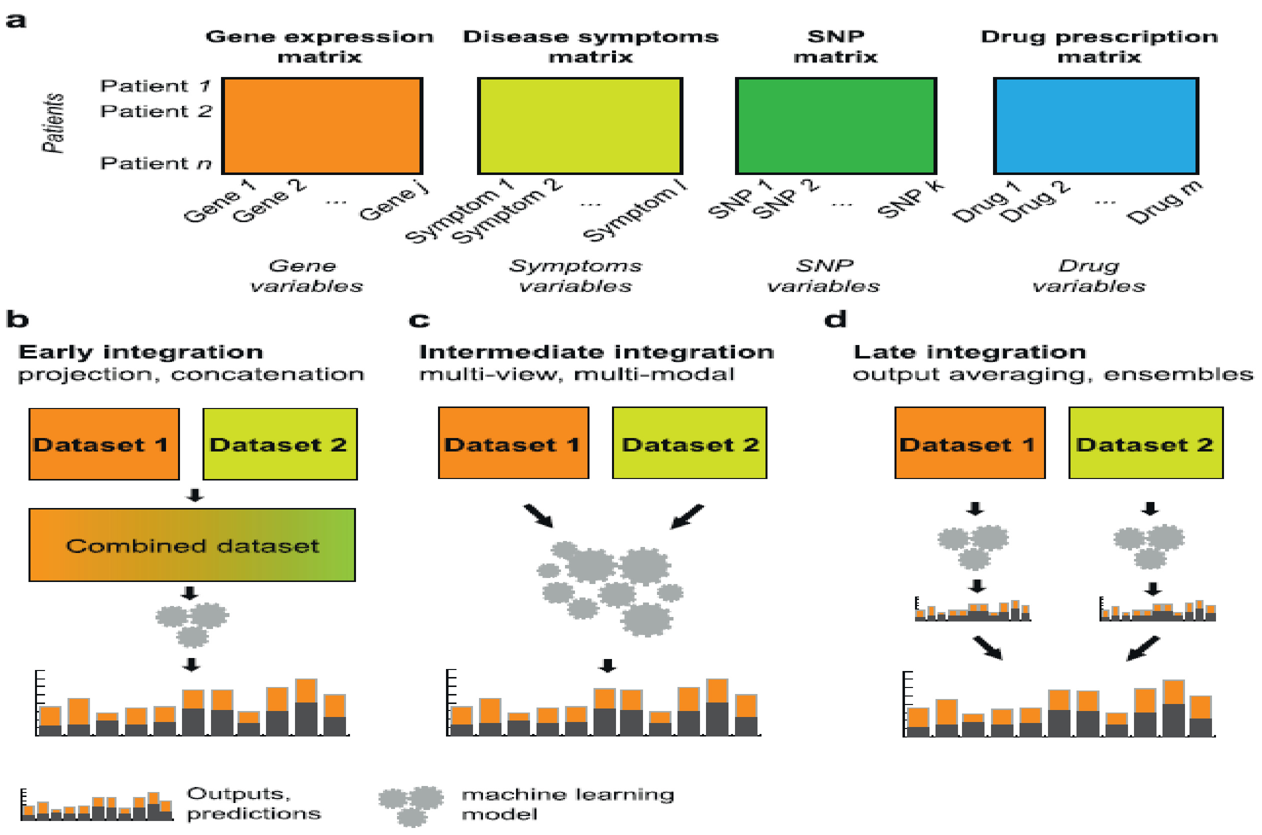
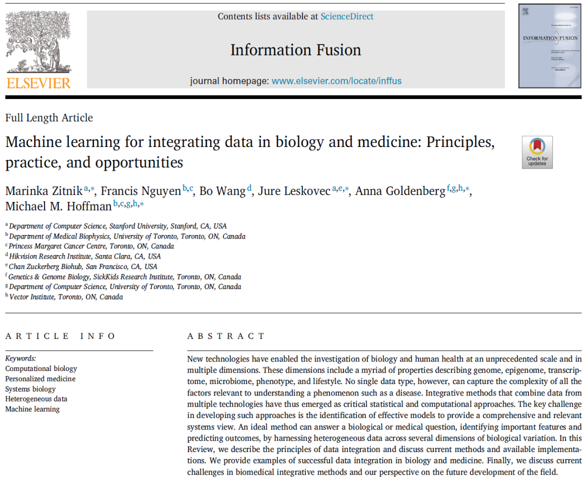


Fig. 2. Categorization of approaches for data integration. (a) Examples of multi-omics data about patients. (b–d) Data integration approaches can be divided into three categories. (b) *Early integration approaches* involve combining datasets from different data types at the raw or processed level before analysis and prediction. (c) *Intermediate integration approaches* transform or map the underlying datasets at the same time as they estimate model parameters. (d) *Late integration approaches* perform analysis on each dataset independently, which is followed by integration of the resulting models to generate predictions, *e.g.*, prognosis for a particular patient. SNP, single-nucleotide polymorphism.



Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities

Marinka Zitnik[a,*], Francis Nguyen[b,c], Bo Wang[d], Jure Leskovec[a,e,*], Anna Goldenberg[f,g,h,*], Michael M. Hoffman[b,c,g,h,*]

[a] *Department of Computer Science, Stanford University, Stanford, CA, USA*
[b] *Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada*
[c] *Princess Margaret Cancer Centre, Toronto, ON, Canada*
[d] *Hikvision Research Institute, Santa Clara, CA, USA*
[e] *Chan Zuckerberg Biohub, San Francisco, CA, USA*
[f] *Genetics & Genome Biology, SickKids Research Institute, Toronto, ON, Canada*
[g] *Department of Computer Science, University of Toronto, Toronto, ON, Canada*
[h] *Vector Institute, Toronto, ON, Canada*

ARTICLE INFO

Keywords:
Computational biology
Personalized medicine
Systems biology
Heterogeneous data
Machine learning

ABSTRACT

New technologies have enabled the investigation of biology and human health at an unprecedented scale and in multiple dimensions. These dimensions include a myriad of properties describing genome, epigenome, transcriptome, microbiome, phenotype, and lifestyle. No single data type, however, can capture the complexity of all the factors relevant to understanding a phenomenon such as a disease. Integrative methods that combine data from multiple technologies have thus emerged as critical statistical and computational approaches. The key challenge in developing such approaches is the identification of effective models to provide a comprehensive and relevant systems view. An ideal method can answer a biological or medical question, identifying important features and predicting outcomes, by harnessing heterogeneous data across several dimensions of biological variation. In this Review, we describe the principles of data integration and discuss current methods and available implementations. We provide examples of successful data integration in biology and medicine. Finally, we discuss current challenges in biomedical integrative methods and our perspective on the future development of the field.

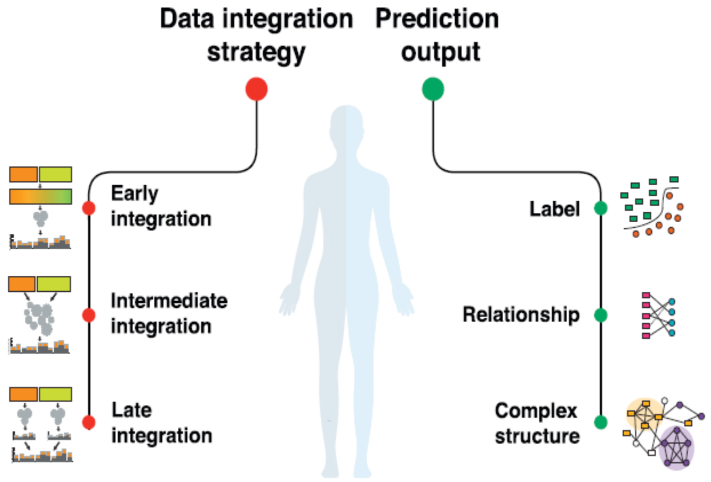


Fig. 3. Data integration. Data integration approaches combine multiples sources of information in a statistically meaningful way to provide a comprehensive analysis of biomedical dara. Broadly, existing approaches use three distinct strategies (*i.e.*, early, intermediate, and late integration; see also Fig. 2) and produce three types of prediction outputs (*i.e.*, a label representing probability of an entity belonging to a given class; a relationship representing probability of an association between two entities; and a complex structure, such as an inferred network or a partitioning of entities into groups).

Machine Learning Applications

Information Fusion 50 (2019) 92–111

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0

Alberto Diez-Olivan [a], Javier Del Ser [a,b,c,*], Diego Galar [a,d], Basilio Sierra [e]

[a] TECNALIA, Donostia-San Sebastián 20009, Spain
[b] Department of Communications Engineering, University of the Basque Country (UPV/EHU), Bilbao 48013, Spain
[c] Basque Center for Applied Mathematics (BCAM), Bilbao, Bizkaia 48009, Spain
[d] Department of Civil, Environmental and Natural Resources Engineering, Operation, Maintenance and Acoustics, Luleå University of Technology, Luleå, Sweden
[e] Department of Computer Sciences and Artificial Intelligence, University of the Basque Country (UPV/EHU), Donostia-San Sebastián 20018, Spain

A R T I C L E I N F O

Keywords:
Data-driven prognosis
Data fusion
Machine learning
Industry 4.0

A B S T R A C T

The so-called “smartization” of manufacturing industries has been conceived as the fourth industrial revolution or Industry 4.0, a paradigm shift propelled by the upsurge and progressive maturity of new Information and Communication Technologies (ICT) applied to industrial processes and products. From a data science perspective, this paradigm shift allows extracting relevant knowledge from monitored assets through the adoption of intelligent monitoring and data fusion strategies, as well as by the application of machine learning and optimization methods. One of the main goals of data science in this context is to effectively predict abnormal behaviors in industrial machinery, tools and processes so as to anticipate critical events and damage, eventually causing important economical losses and safety issues. In this context, data-driven prognosis is gradually gaining attention in different industrial sectors. This paper provides a comprehensive survey of the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity and unexplored challenges. To this end, a principled categorization of the utilized feature extraction techniques and machine learning methods will be provided on the basis of its intended purpose: analyze what caused the failure (descriptive), determine when the monitored asset will fail (predictive) or decide what to do so as to minimize its impact on the industry at hand (prescriptive). This threefold analysis, along with a discussion on its hardware and software implications, intends to serve as a stepping stone for future researchers and practitioners to join the community investigating on this vibrant field.

Machine Learning Applications

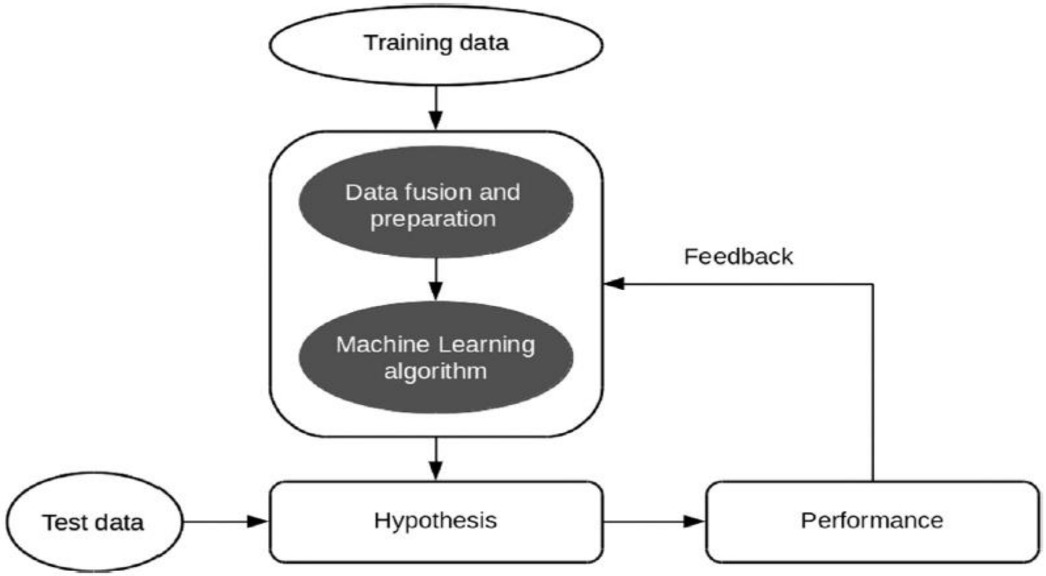


Fig. 1. Generic schematic diagram of a machine learning process.

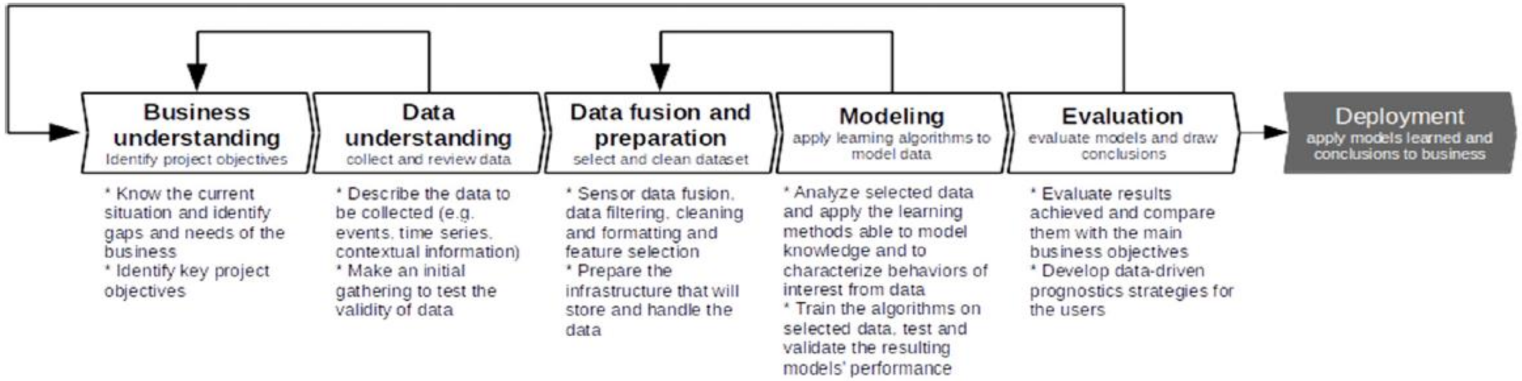
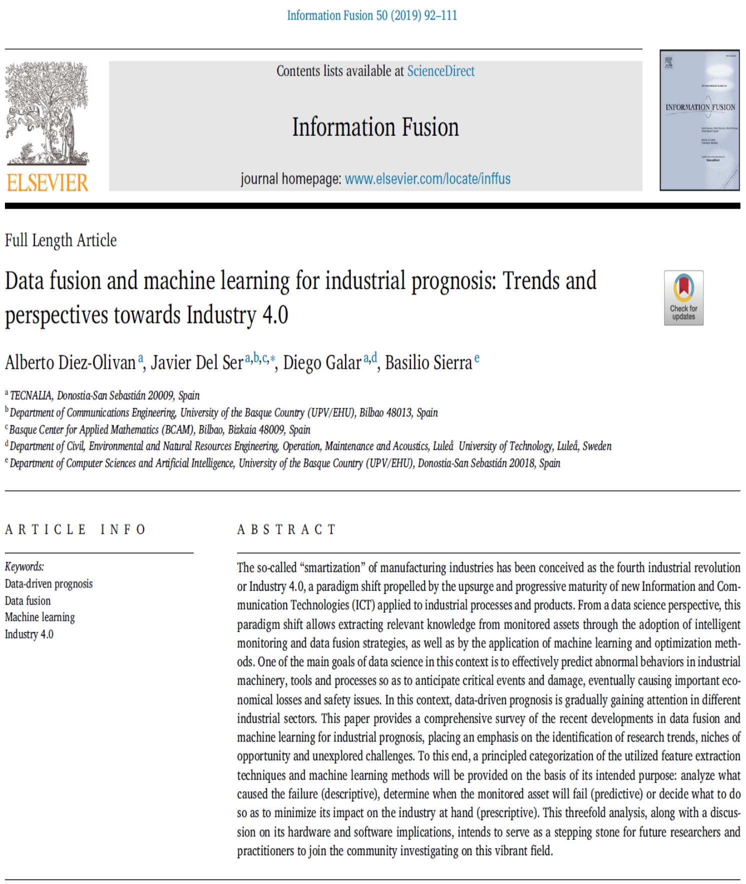


Fig. 2. The CRISP methodology for data-driven industrial prognosis.



Information Fusion 50 (2019) 92–111

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

ELSEVIER

Full Length Article

Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0

Alberto Diez-Olivan[a], Javier Del Ser[a,b,c,*], Diego Galar[a,d], Basilio Sierra[e]

[a] TECNALIA, Donostia-San Sebastián 20009, Spain
[b] Department of Communications Engineering, University of the Basque Country (UPV/EHU), Bilbao 48013, Spain
[c] Basque Center for Applied Mathematics (BCAM), Bilbao, Bizkaia 48009, Spain
[d] Department of Civil, Environmental and Natural Resources Engineering, Operation, Maintenance and Acoustics, Luleå University of Technology, Luleå, Sweden
[e] Department of Computer Sciences and Artificial Intelligence, University of the Basque Country (UPV/EHU), Donostia-San Sebastián 20018, Spain

ARTICLE INFO

Keywords:
Data-driven prognosis
Data fusion
Machine learning
Industry 4.0

ABSTRACT

The so-called "smartization" of manufacturing industries has been conceived as the fourth industrial revolution or Industry 4.0, a paradigm shift propelled by the upsurge and progressive maturity of new Information and Communication Technologies (ICT) applied to industrial processes and products. From a data science perspective, this paradigm shift allows extracting relevant knowledge from monitored assets through the adoption of intelligent monitoring and data fusion strategies, as well as by the application of machine learning and optimization methods. One of the main goals of data science in this context is to effectively predict abnormal behaviors in industrial machinery, tools and processes so as to anticipate critical events and damage, eventually causing important economical losses and safety issues. In this context, data-driven prognosis is gradually gaining attention in different industrial sectors. This paper provides a comprehensive survey of the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity and unexplored challenges. To this end, a principled categorization of the utilized feature extraction techniques and machine learning methods will be provided on the basis of its intended purpose: analyze what caused the failure (descriptive), determine when the monitored asset will fail (predictive) or decide what to do so as to minimize its impact on the industry at hand (prescriptive). This threefold analysis, along with a discussion on its hardware and software implications, intends to serve as a stepping stone for future researchers and practitioners to join the community investigating on this vibrant field.

Machine Learning Applications

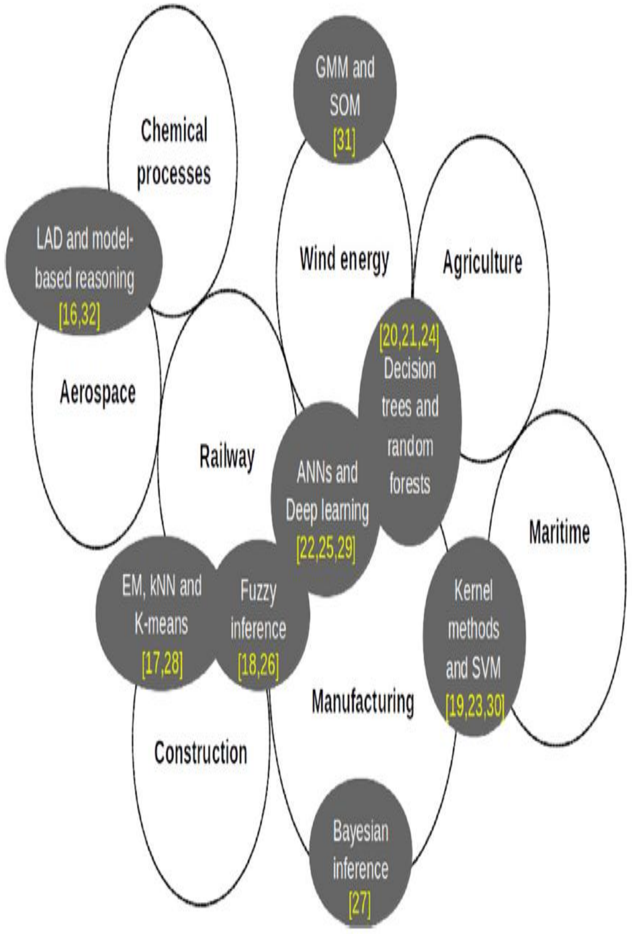


Fig. 4. Solutions and industrial sectors addressed for descriptive prognosis.

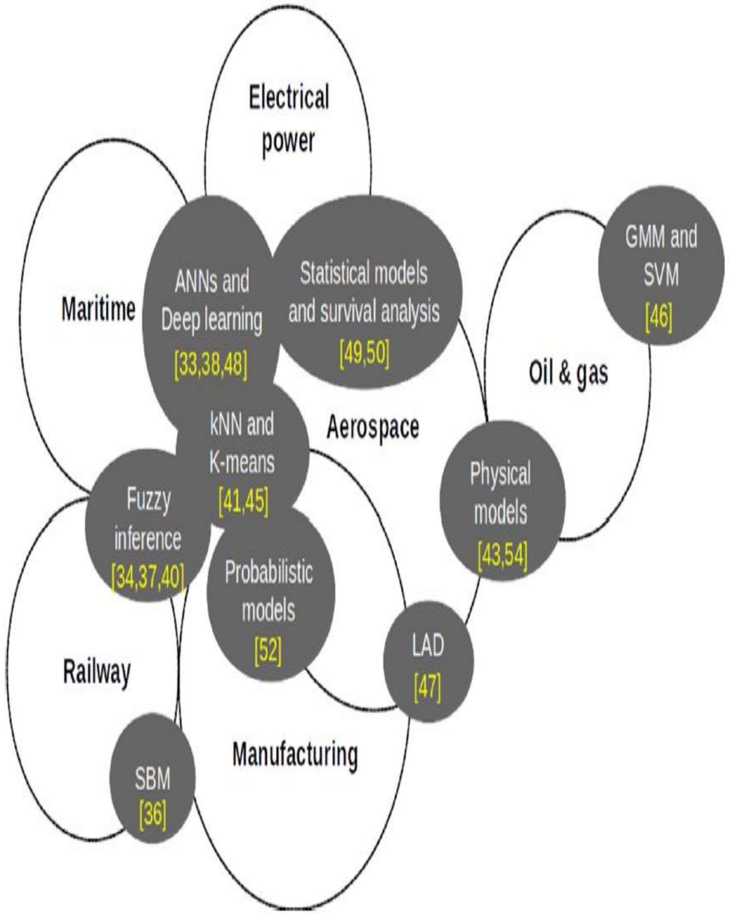
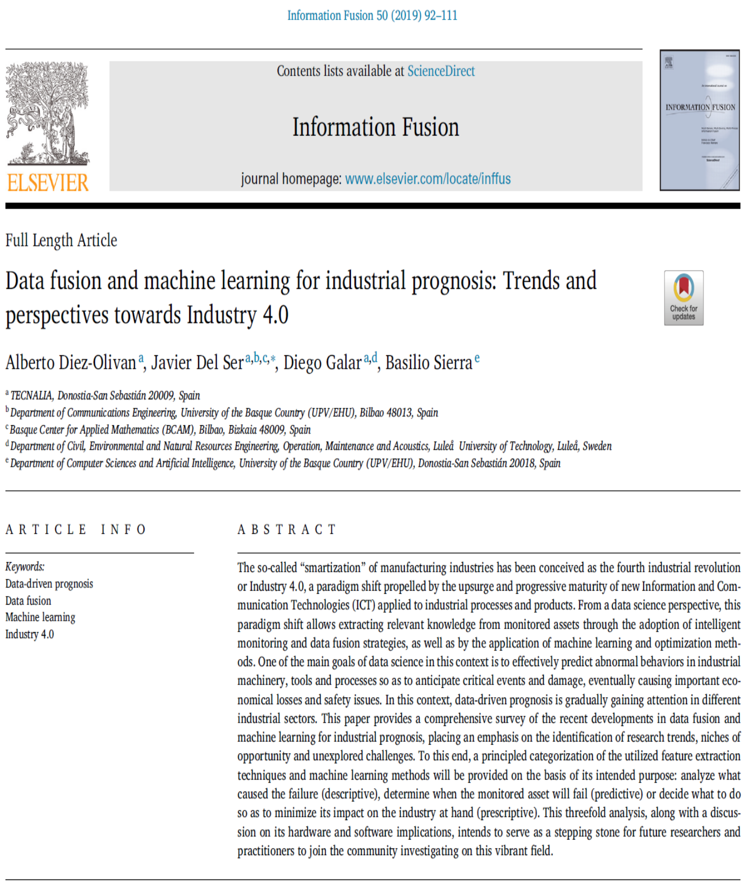


Fig. 5. Solutions and industrial sectors addressed for predictive prognostics.



Information Fusion 50 (2019) 92–111

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0

Alberto Diez-Olivan[a], Javier Del Ser[a,b,c,*], Diego Galar[a,d], Basilio Sierra[e]

[a] TECNALIA, Donostia-San Sebastián 20009, Spain
[b] Department of Communications Engineering, University of the Basque Country (UPV/EHU), Bilbao 48013, Spain
[c] Basque Center for Applied Mathematics (BCAM), Bilbao, Bizkaia 48009, Spain
[d] Department of Civil, Environmental and Natural Resources Engineering, Operation, Maintenance and Acoustics, Luleå University of Technology, Luleå, Sweden
[e] Department of Computer Sciences and Artificial Intelligence, University of the Basque Country (UPV/EHU), Donostia-San Sebastián 20018, Spain

ARTICLE INFO

Keywords:
Data-driven prognosis
Data fusion
Machine learning
Industry 4.0

ABSTRACT

The so-called “smartization” of manufacturing industries has been conceived as the fourth industrial revolution or Industry 4.0, a paradigm shift propelled by the upsurge and progressive maturity of new Information and Communication Technologies (ICT) applied to industrial processes and products. From a data science perspective, this paradigm shift allows extracting relevant knowledge from monitored assets through the adoption of intelligent monitoring and data fusion strategies, as well as by the application of machine learning and optimization methods. One of the main goals of data science in this context is to effectively predict abnormal behaviors in industrial machinery, tools and processes so as to anticipate critical events and damage, eventually causing important economical losses and safety issues. In this context, data-driven prognosis is gradually gaining attention in different industrial sectors. This paper provides a comprehensive survey of the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity and unexplored challenges. To this end, a principled categorization of the utilized feature extraction techniques and machine learning methods will be provided on the basis of its intended purpose: analyze what caused the failure (descriptive), determine when the monitored asset will fail (predictive) or decide what to do so as to minimize its impact on the industry at hand (prescriptive). This threefold analysis, along with a discussion on its hardware and software implications, intends to serve as a stepping stone for future researchers and practitioners to join the community investigating on this vibrant field.

Machine Learning Applications

Information Fusion 53 (2020) 222–239

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Online heart monitoring systems on the internet of health things environments: A survey, a reference model and an outlook

Marcus A.G. Santos [a], Roberto Munoz [b,c], Rodrigo Olivares [d], Pedro P. Rebouças Filho [e], Javier Del Ser [f], Victor Hugo C. de Albuquerque [a,*]

[a] *University of Fortaleza, Fortaleza, CE, Brazil*
[b] *School of Informatics Engineering, Universidad de Valparaíso, Valparaíso, Chile*
[c] *Centro de Investigación y Desarrollo en Ingeniería en Salud, Universidad de Valparaíso, Valparaíso, Chile*
[d] *Pontificia Universidad Católica de Valparaíso, Valparaíso, 2362807, Chile*
[e] *Federal Institute of Education, Science and Technology of Ceará, Fortaleza, CE, Brazil*
[f] *TECNALIA, Donostia-San Sebastián, Spain. Department of Communications Engineering, University of the Basque Country, Bilbao, Spain. Basque Center for Applied Mathematics, Bilbao, Bizkaia, Spain*

A R T I C L E I N F O

Keywords:
Internet of health things
Heart
Bio sensors
Online monitoring
Reference model

A B S T R A C T

The Internet of Health Things promotes personalized and higher standards of care. Its application is diverse and attracts the attention of a substantial section of the scientific community. This approach has also been applied by people looking to enhance quality of life by using this technology. In this paper, we perform a survey that aims to present and analyze the advances of the latest studies based on medical care and assisted environment. We focus on articles for online monitoring, detection, and support of the diagnosis of cardiovascular diseases. Our research covers published manuscripts in scientific journals and recognized conferences since the year 2015. Also, we present a reference model based on the evaluation of the resources used from the selected studies. Finally, our proposal aims to help future enthusiasts to discover and enumerate the required factors for the development of a prototype for online heart monitoring purposes.

Machine Learning Applications

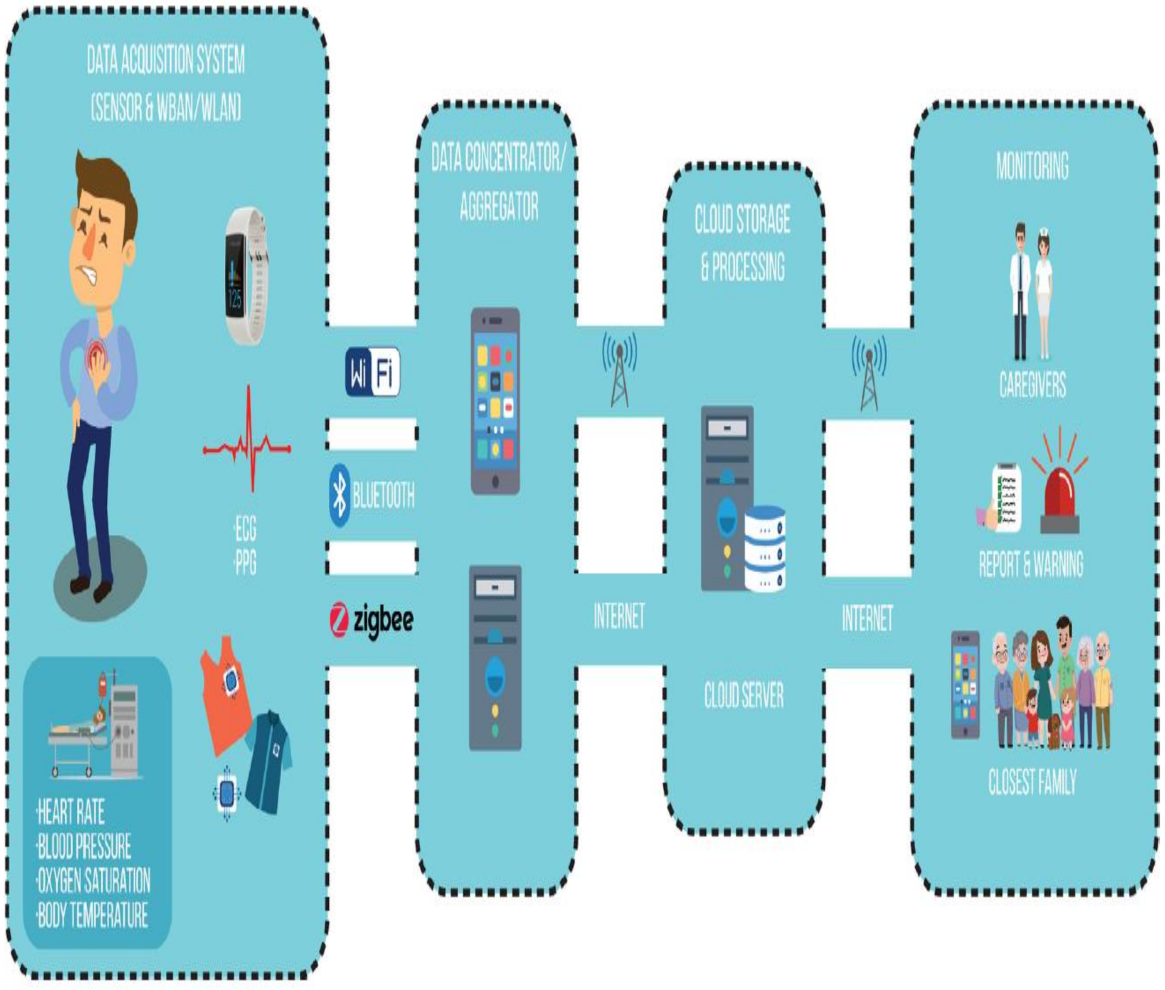


Fig. 1. IoT of the Heart.

Information Fusion 53 (2020) 222–239

Contents lists available at ScienceDirect

ELSEVIER

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Online heart monitoring systems on the internet of health things environments: A survey, a reference model and an outlook

Marcus A.G. Santos[a], Roberto Munoz[b,c], Rodrigo Olivares[d], Pedro P. Rebouças Filho[e], Javier Del Ser[f], Victor Hugo C. de Albuquerque[a,*]

[a] *University of Fortaleza, Fortaleza, CE, Brazil*
[b] *School of Informatics Engineering, Universidad de Valparaíso, Valparaíso, Chile*
[c] *Centro de Investigación y Desarrollo en Ingeniería en Salud, Universidad de Valparaíso, Valparaíso, Chile*
[d] *Pontificia Universidad Católica de Valparaíso, Valparaíso, 2362807, Chile*
[e] *Federal Institute of Education, Science and Technology of Ceará, Fortaleza, CE, Brazil*
[f] *TECNALIA, Donostia-San Sebastián, Spain. Department of Communications Engineering, University of the Basque Country, Bilbao, Spain. Basque Center for Applied Mathematics, Bilbao, Bizkaia, Spain*

A R T I C L E I N F O

Keywords:
Internet of health things
Heart
Bio sensors
Online monitoring
Reference model

A B S T R A C T

The Internet of Health Things promotes personalized and higher standards of care. Its application is diverse and attracts the attention of a substantial section of the scientific community. This approach has also been applied by people looking to enhance quality of life by using this technology. In this paper, we perform a survey that aims to present and analyze the advances of the latest studies based on medical care and assisted environment. We focus on articles for online monitoring, detection, and support of the diagnosis of cardiovascular diseases. Our research covers published manuscripts in scientific journals and recognized conferences since the year 2015. Also, we present a reference model based on the evaluation of the resources used from the selected studies. Finally, our proposal aims to help future enthusiasts to discover and enumerate the required factors for the development of a prototype for online heart monitoring purposes.

Machine Learning Applications

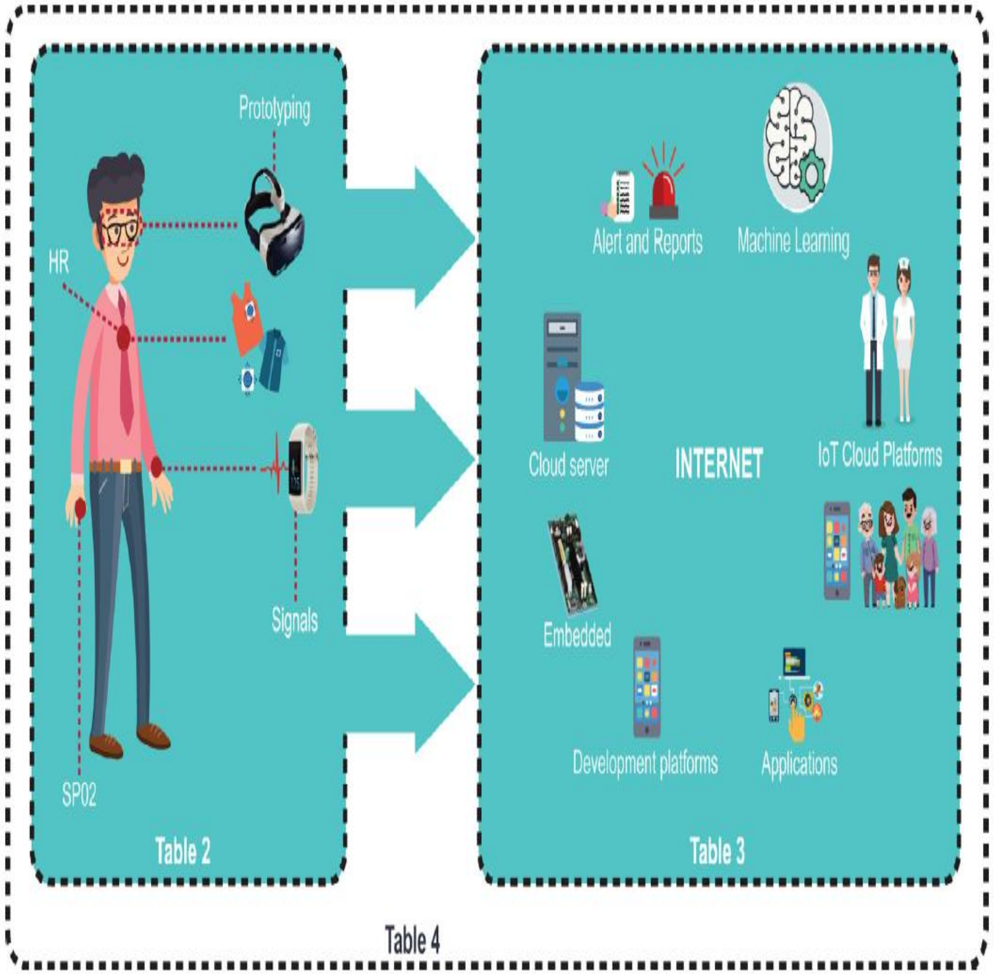


Fig. 2. Development framework.

Information Fusion 53 (2020) 222–239

Contents lists available at ScienceDirect

ELSEVIER

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Online heart monitoring systems on the internet of health things environments: A survey, a reference model and an outlook

Marcus A.G. Santos[a], Roberto Munoz[b,c], Rodrigo Olivares[d], Pedro P. Rebouças Filho[e], Javier Del Ser[f], Victor Hugo C. de Albuquerque[a,*]

[a] *University of Fortaleza, Fortaleza, CE, Brazil*
[b] *School of Informatics Engineering, Universidad de Valparaíso, Valparaíso, Chile*
[c] *Centro de Investigación y Desarrollo en Ingeniería en Salud, Universidad de Valparaíso, Valparaíso, Chile*
[d] *Pontificia Universidad Católica de Valparaíso, Valparaíso, 2362807, Chile*
[e] *Federal Institute of Education, Science and Technology of Ceará, Fortaleza, CE, Brazil*
[f] *TECNALIA, Donostia-San Sebastián, Spain. Department of Communications Engineering, University of the Basque Country, Bilbao, Spain. Basque Center for Applied Mathematics, Bilbao, Bizkaia, Spain*

A R T I C L E   I N F O

Keywords:
Internet of health things
Heart
Bio sensors
Online monitoring
Reference model

A B S T R A C T

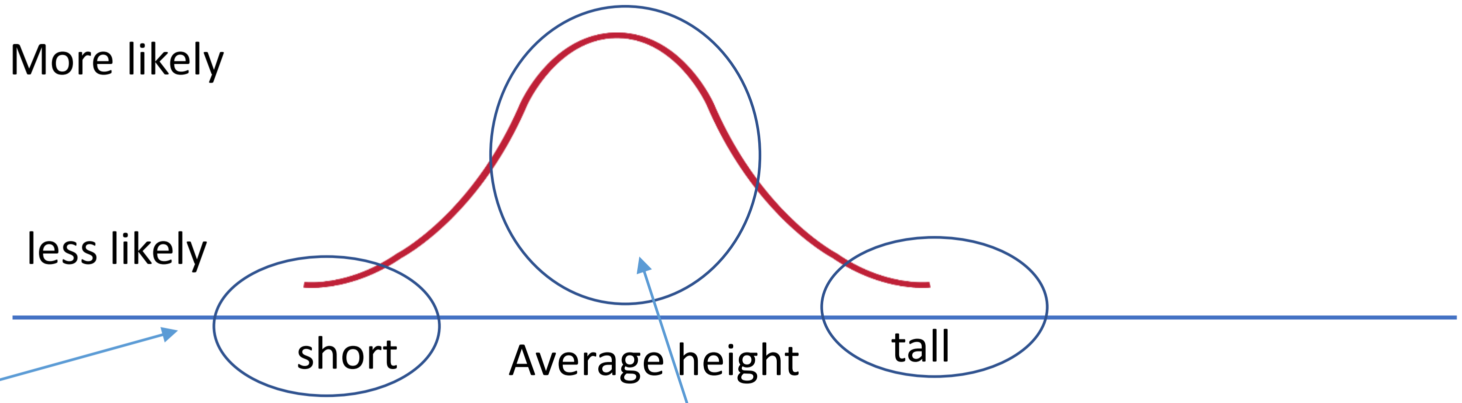
The Internet of Health Things promotes personalized and higher standards of care. Its application is diverse and attracts the attention of a substantial section of the scientific community. This approach has also been applied by people looking to enhance quality of life by using this technology. In this paper, we perform a survey that aims to present and analyze the advances of the latest studies based on medical care and assisted environment. We focus on articles for online monitoring, detection, and support of the diagnosis of cardiovascular diseases. Our research covers published manuscripts in scientific journals and recognized conferences since the year 2015. Also, we present a reference model based on the evaluation of the resources used from the selected studies. Finally, our proposal aims to help future enthusiasts to discover and enumerate the required factors for the development of a prototype for online heart monitoring purposes.

Normal Distribution

Normal Distribution

y-axis is the relative probability of observing some one

who is really short or really tall or who has an average height



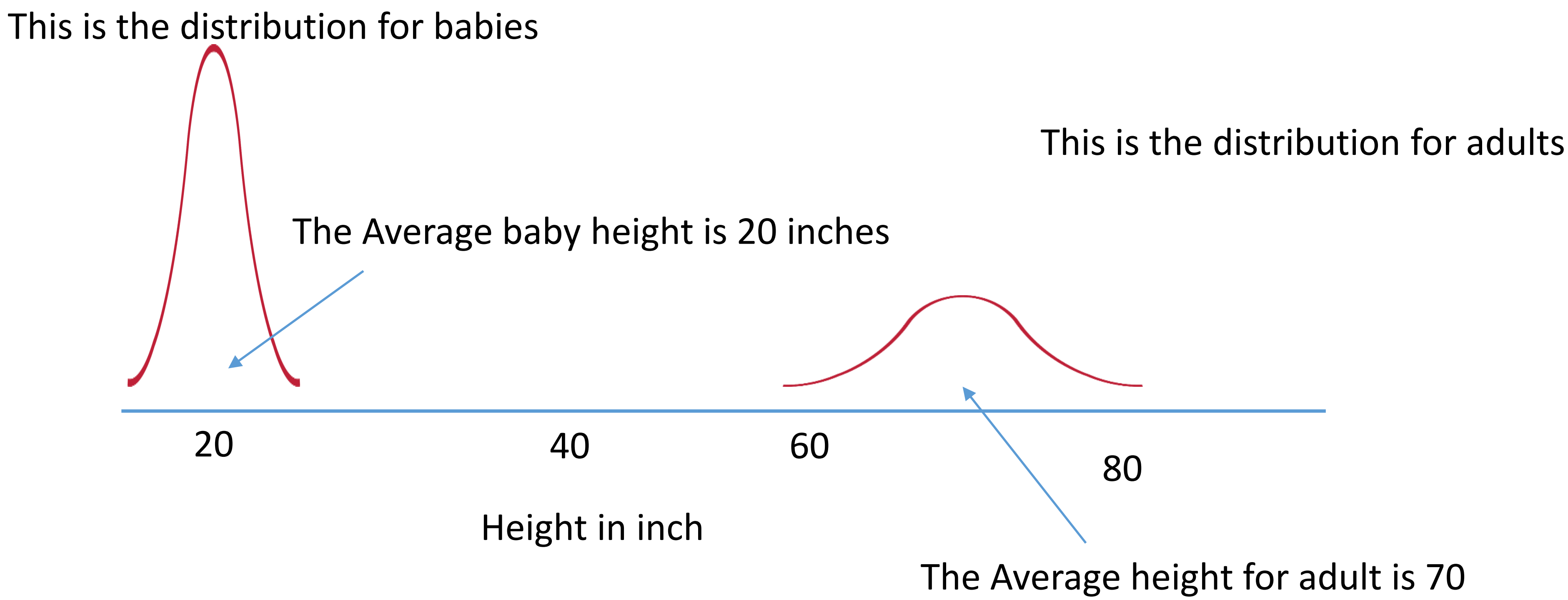
for example, its relatively rare to see someone who is super short

for example, its relatively rare to see someone who is super short

but its quite common to see someone who is close to the average height

Normal Distribution

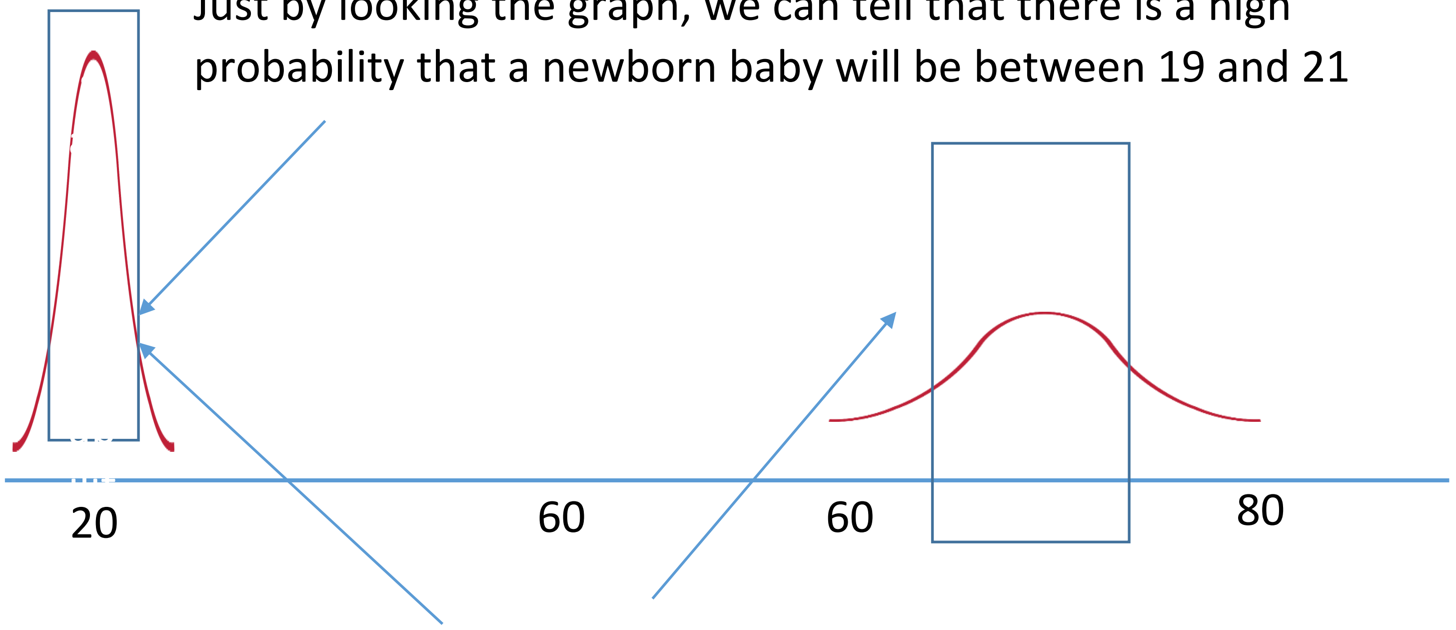
Two normal distribution of height male humans when born and as adults



Normal Distribution

Two normal distribution of height male humans when born and as adults

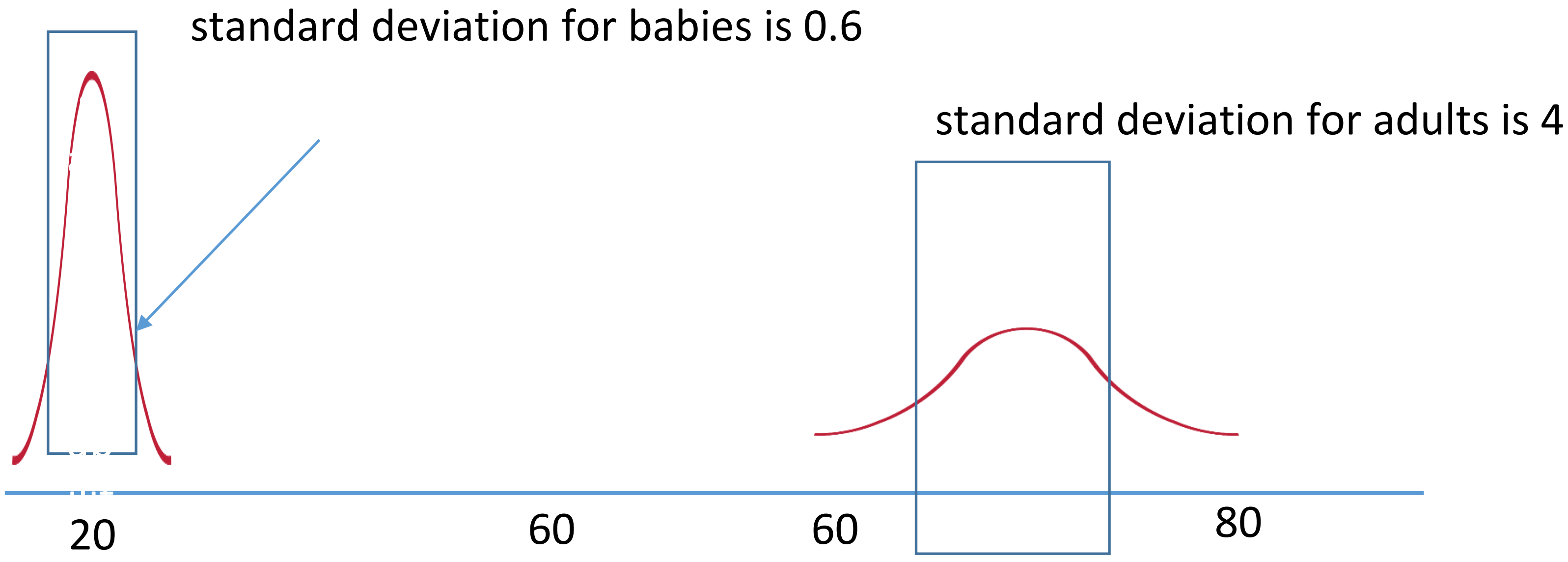
Just by looking the graph, we can tell that there is a high probability that a newborn baby will be between 19 and 21



Just by looking the graph, we can tell that babies have relatively small standard deviation as compared to adults

Normal Distribution

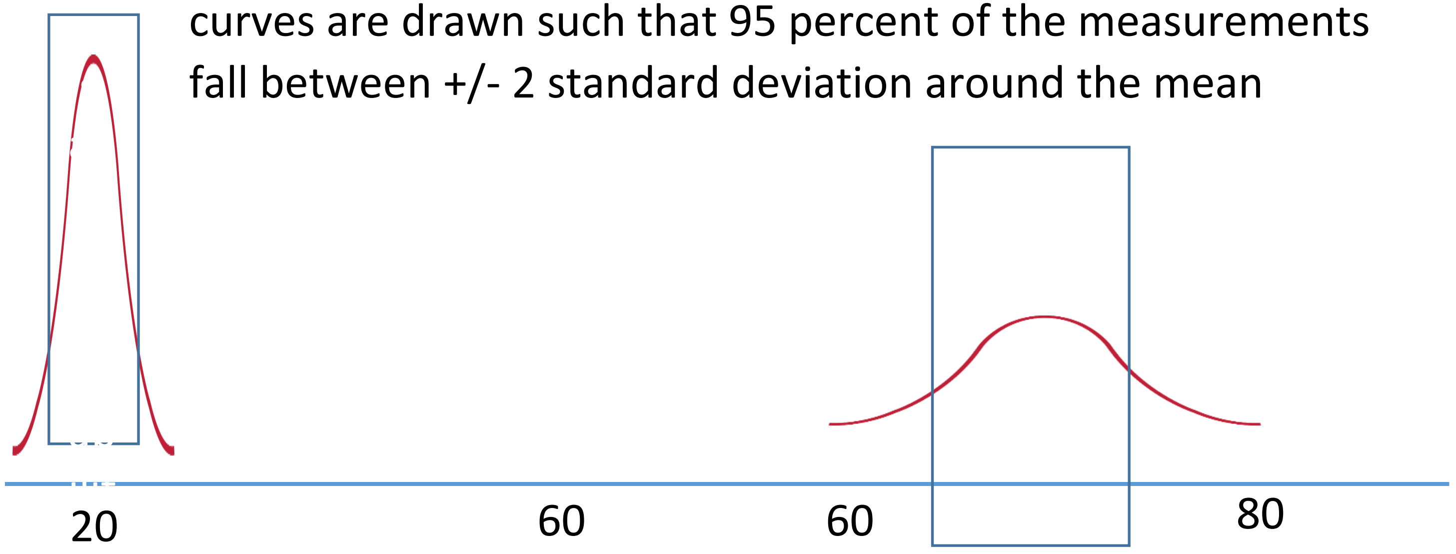
Two normal distribution of height male humans when born and as adults



Normal Distribution

Two normal distribution of height male humans when born and as adults

knowing the standard deviation is helpful because normal curves are drawn such that 95 percent of the measurements fall between +/- 2 standard deviation around the mean

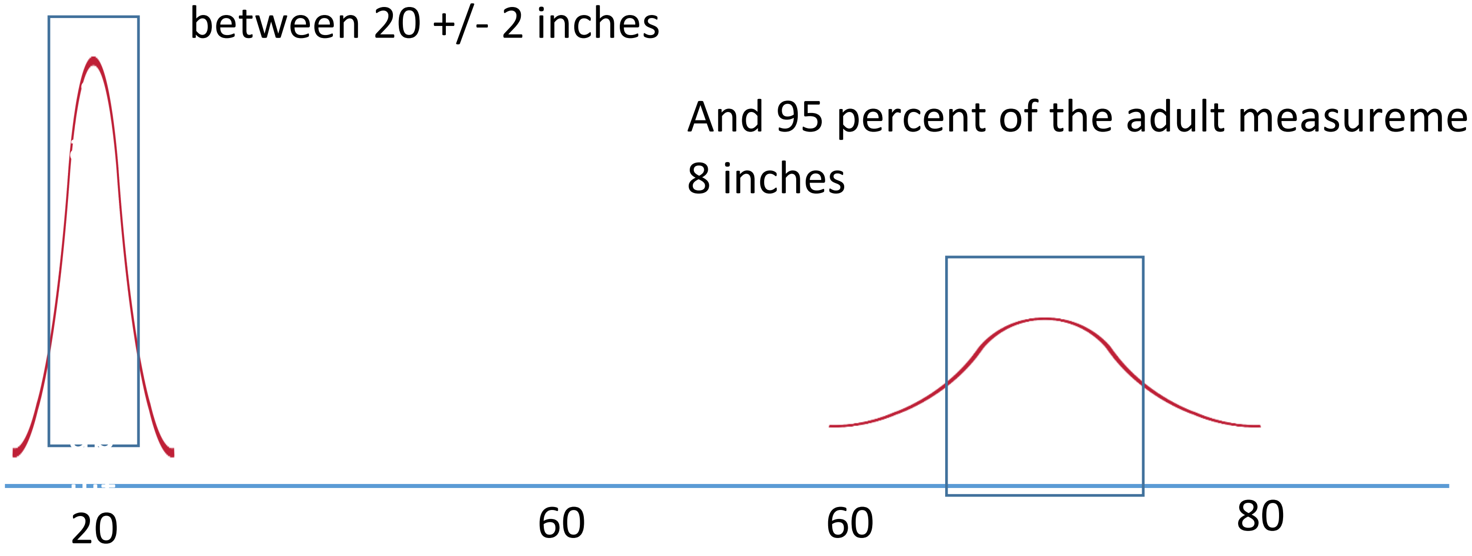


Normal Distribution

Two normal distribution of height male humans when born and as adults

this means that 95 percent of the baby measurements fall between 20 +/- 2 inches

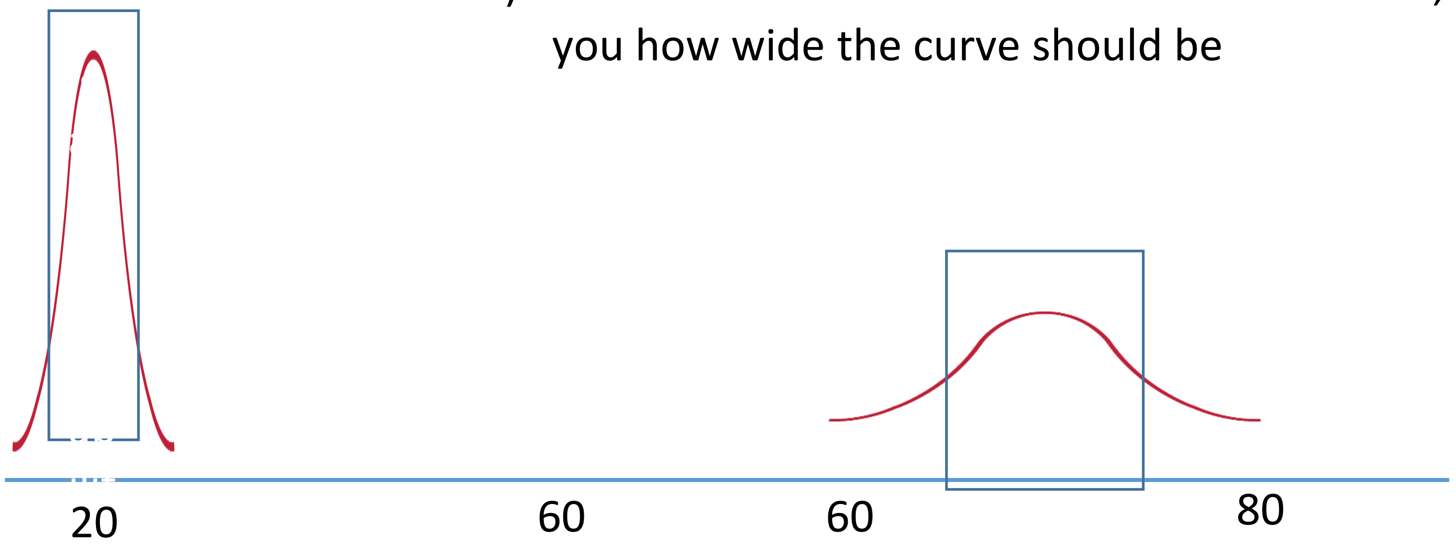
And 95 percent of the adult measurements fall between 70 +/- 8 inches



Normal Distribution

To draw a normal distribution, you need to know:

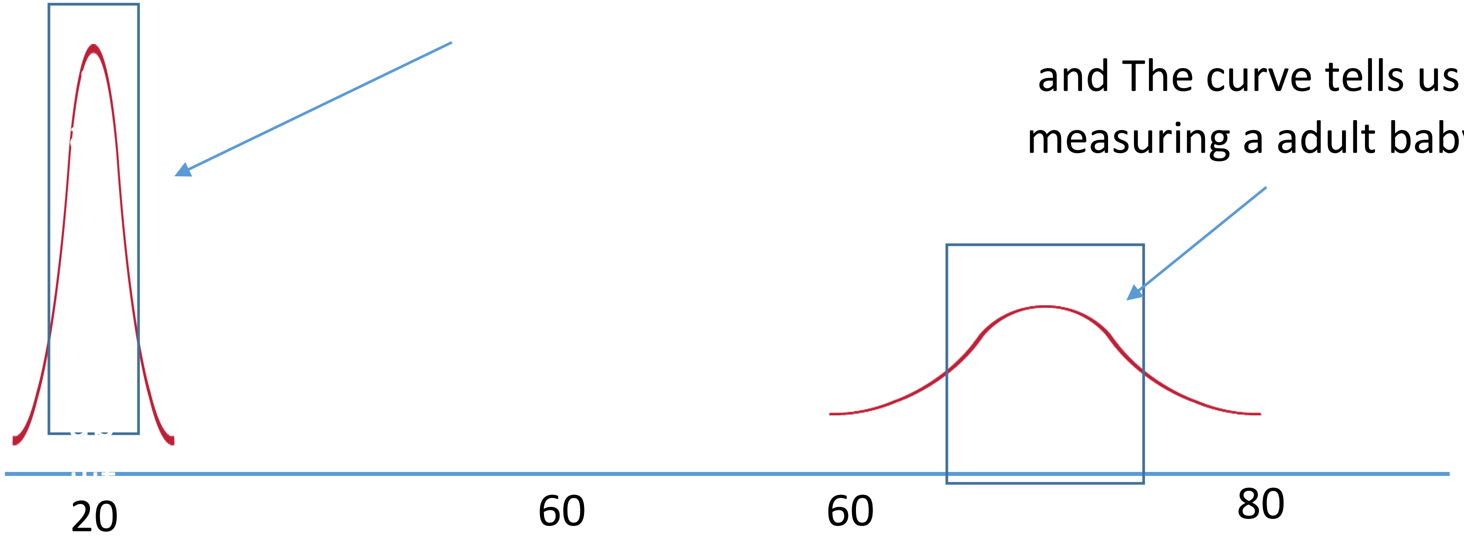
1) Average measurement. This tells you where the center of the curve
2) The standard deviation of the measurements, this tells us you how wide the curve should be



Normal Distribution

The curve tells us that there is a high probability of measuring a newborn baby boy within +/- 1.2 inches of the mean

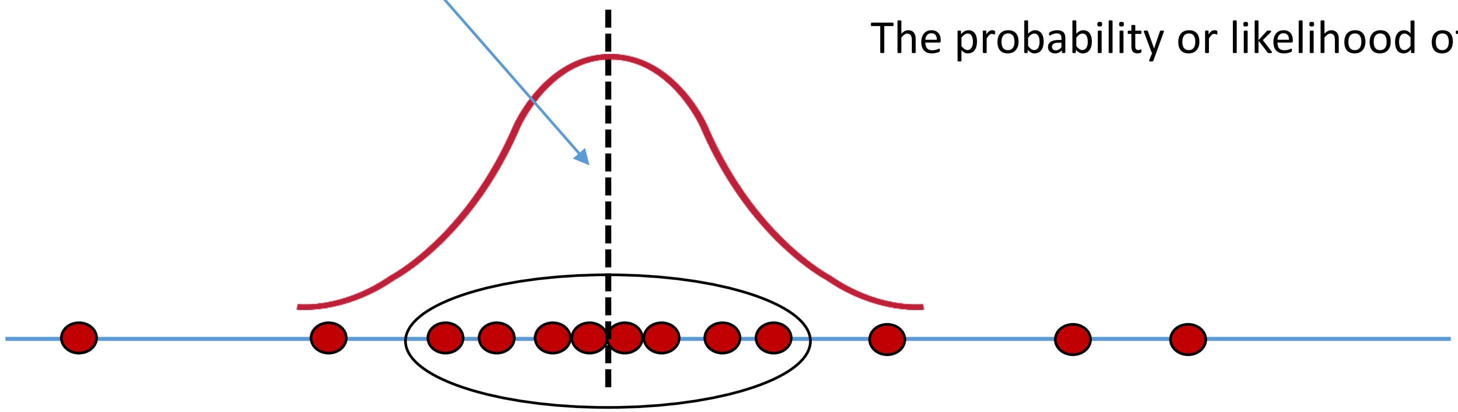
and The curve tells us that there is a low probability of measuring a adult baby boy within +/- 1.2 inches of the mean



Maximum Likelihood for Normal Distribution

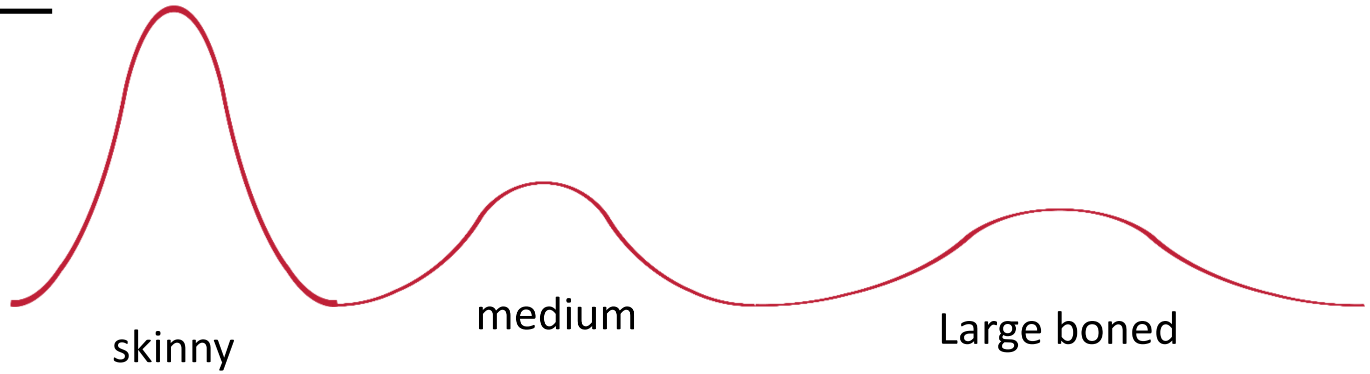
According to normal distribution with mean here

The probability or likelihood of observing these weights is high



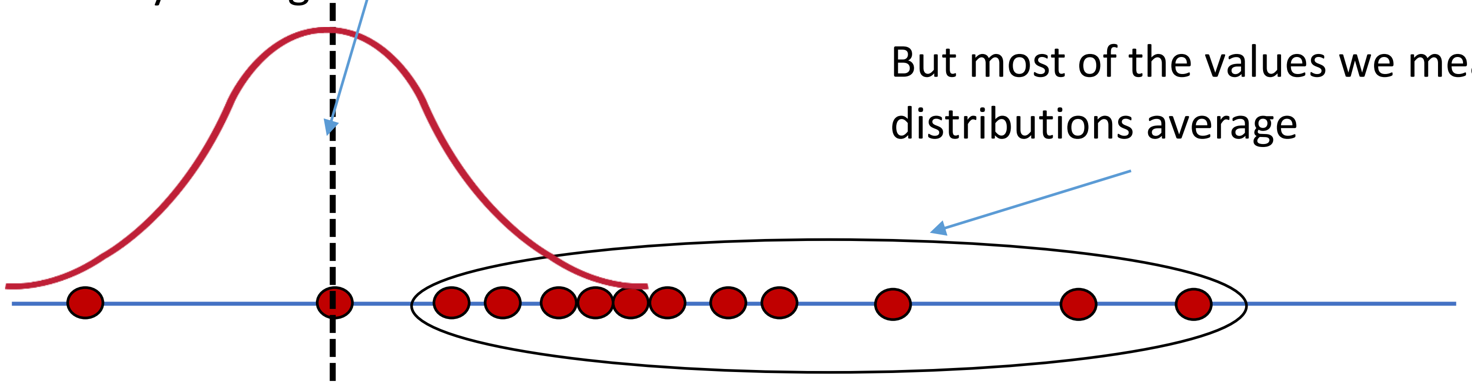
Maximum Likelihood for Normal Distribution

Maximum Likelihood for Normal Distribution

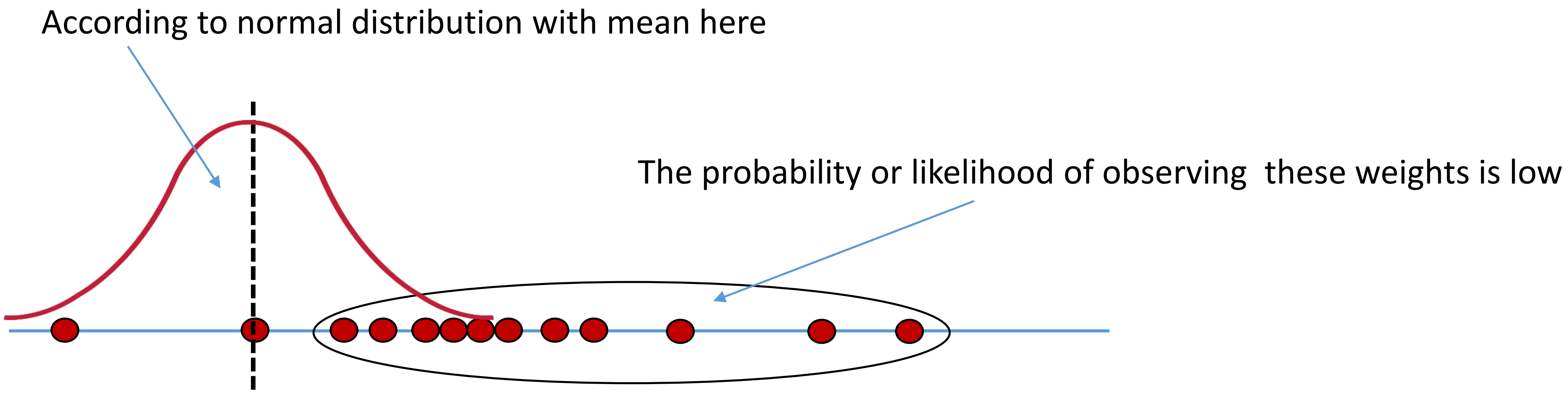


This distribution says most of the values you measure should be near my average

But most of the values we measured are far from the distributions average

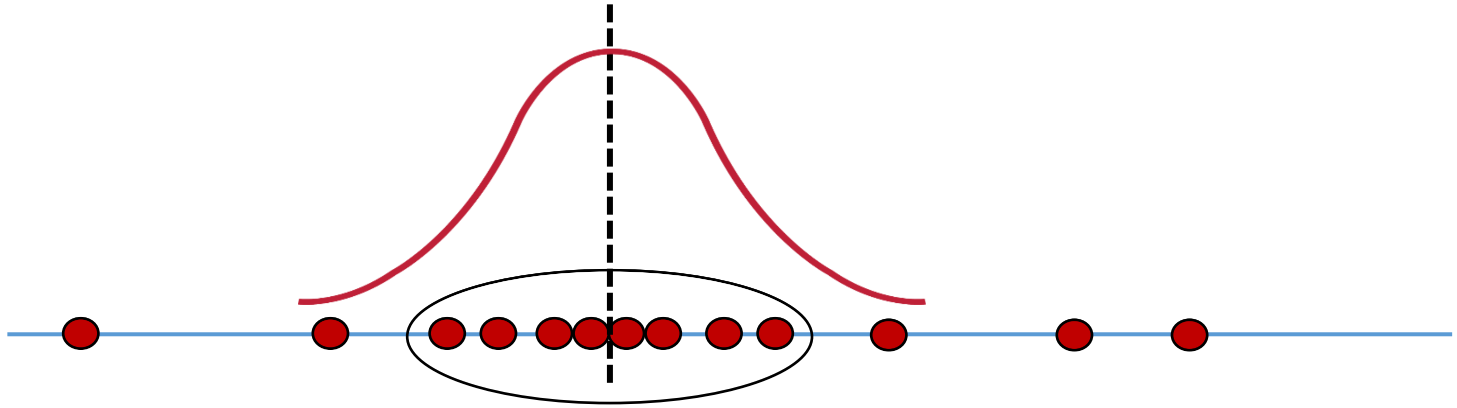


Maximum Likelihood for Normal Distribution



Maximum Likelihood for Normal Distribution

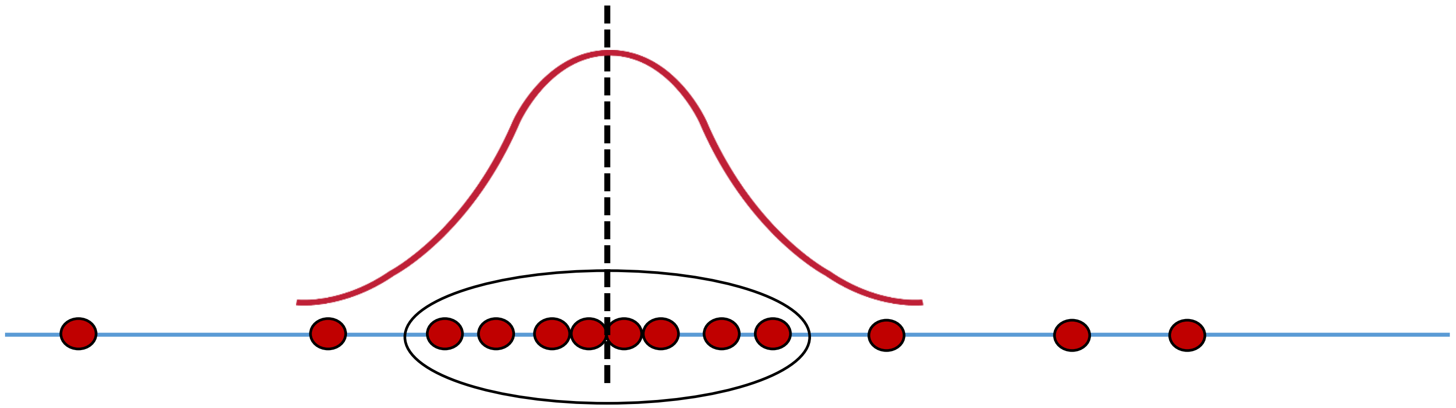
What if we shifted the normal distribution over, so that its mean was the same as the average weight?



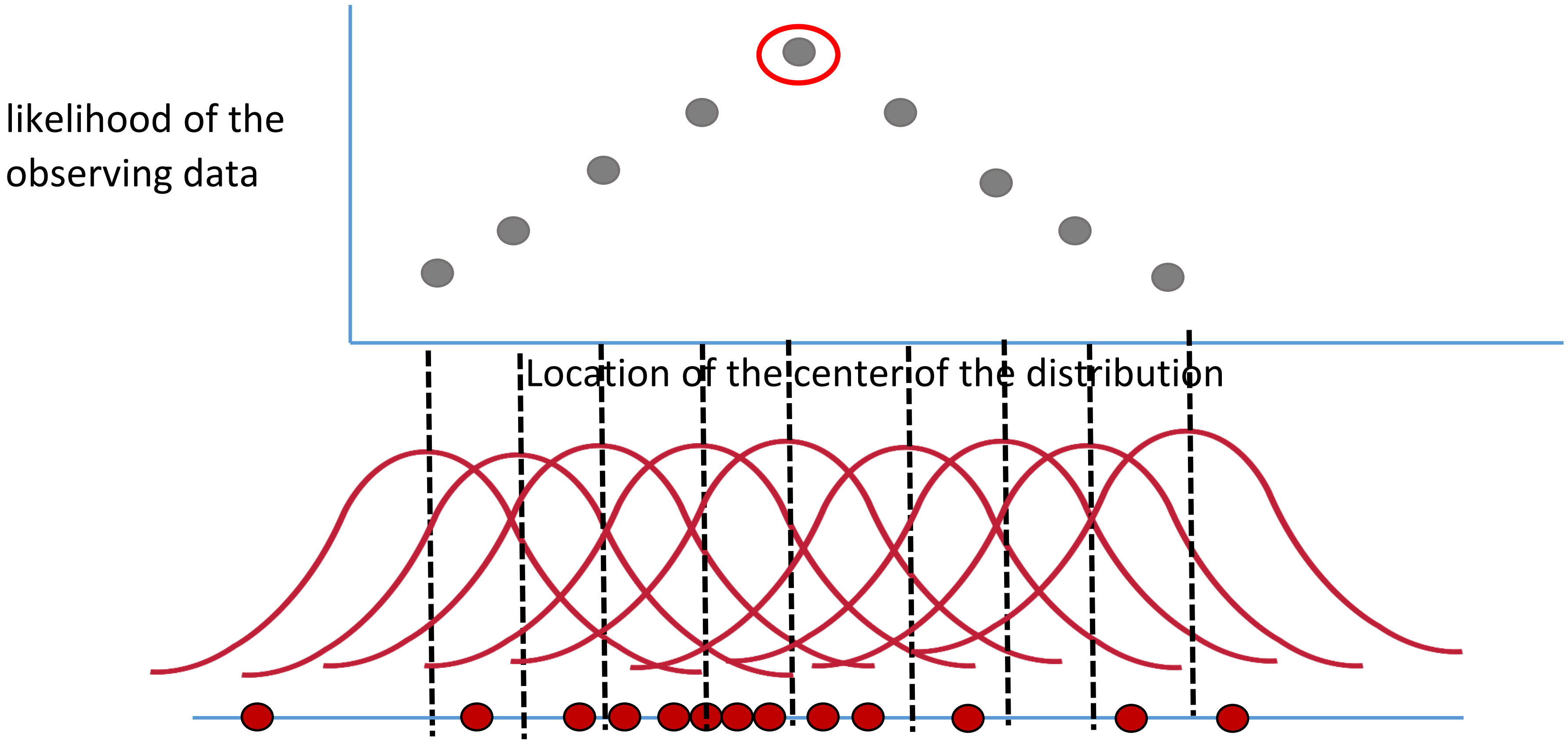
Maximum Likelihood for Normal Distribution

According to normal distribution with mean here

The probability or likelihood of observing these weights is high

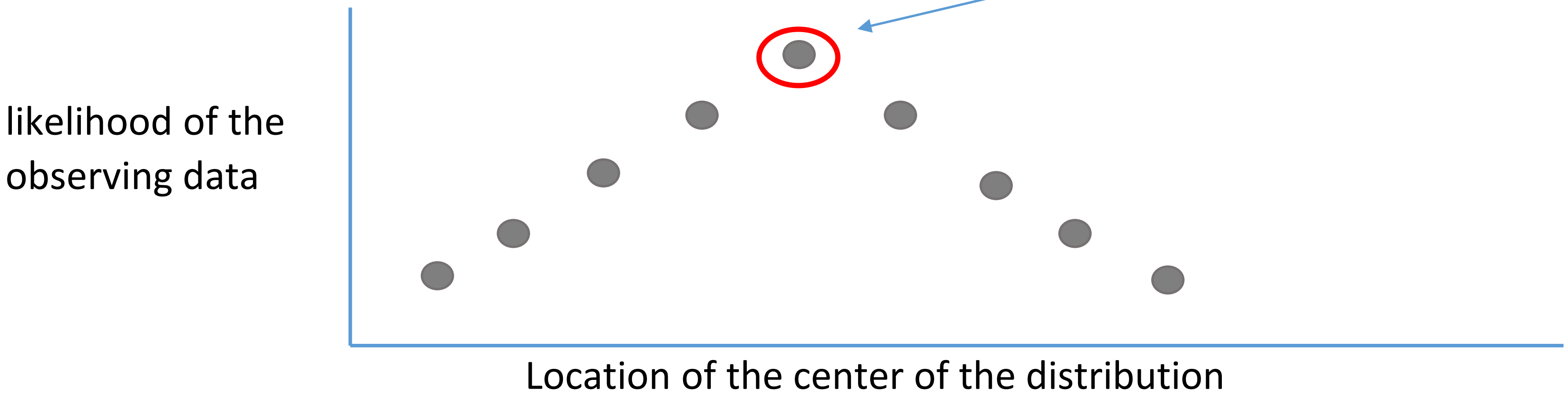


Maximum Likelihood for Normal Distribution



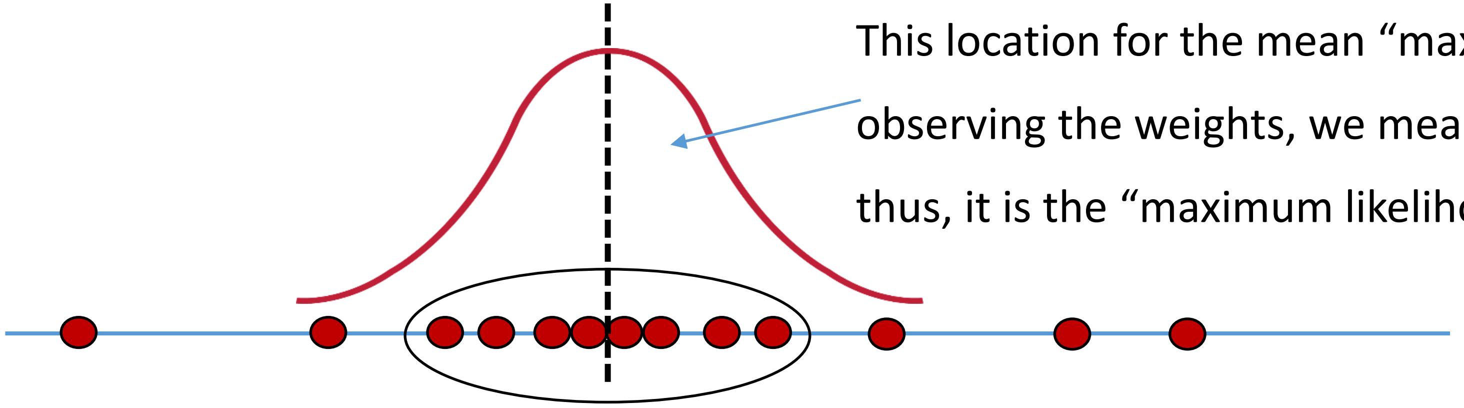
Maximum Likelihood for Normal Distribution

We want the location that "maximizes the likelihood" of observing the weights, we measured



This location for the mean "maximizes the likelihood" of observing the weights, we measured

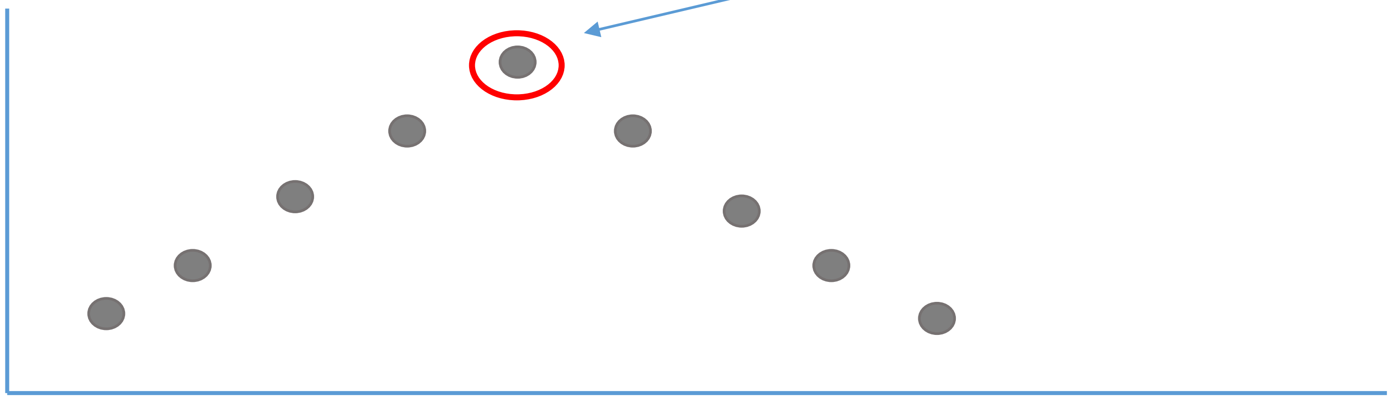
thus, it is the "maximum likelihood estimate for the mean"



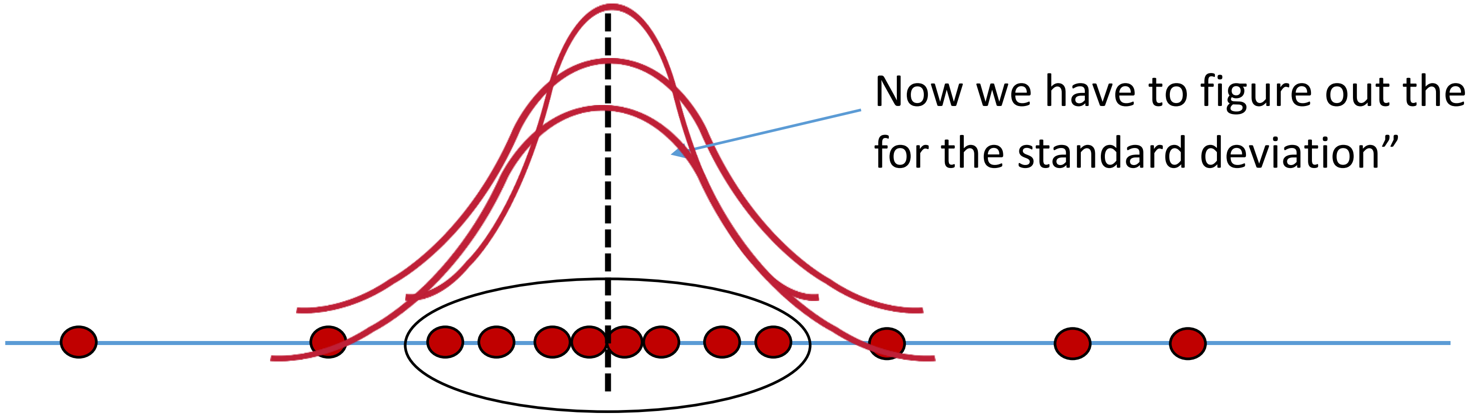
Maximum Likelihood for Normal Distribution

We want the location that "maximizes the likelihood" of observing the weights, we measured

likelihood of the observing data

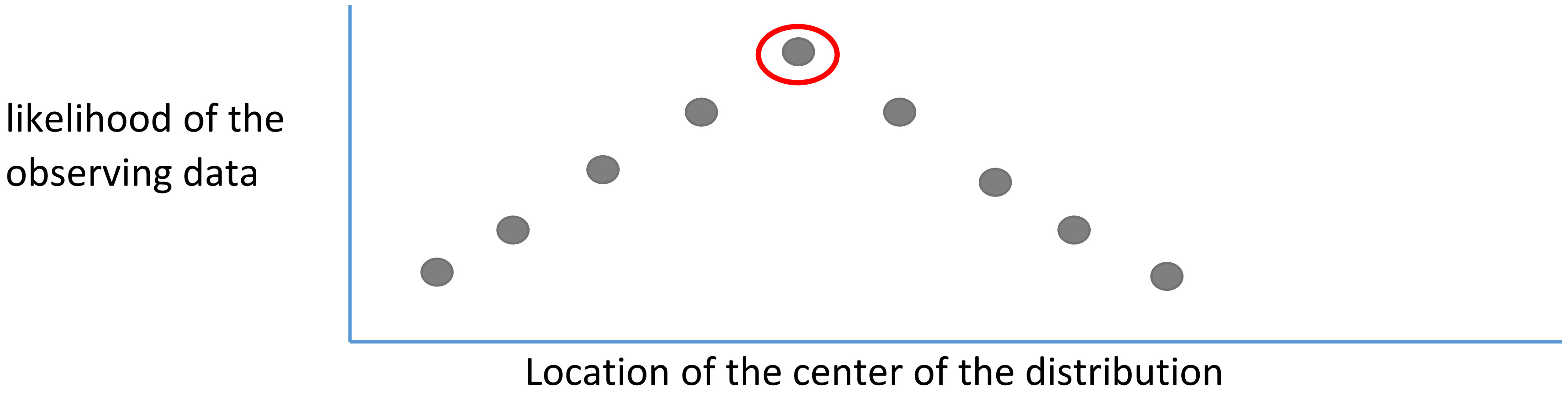


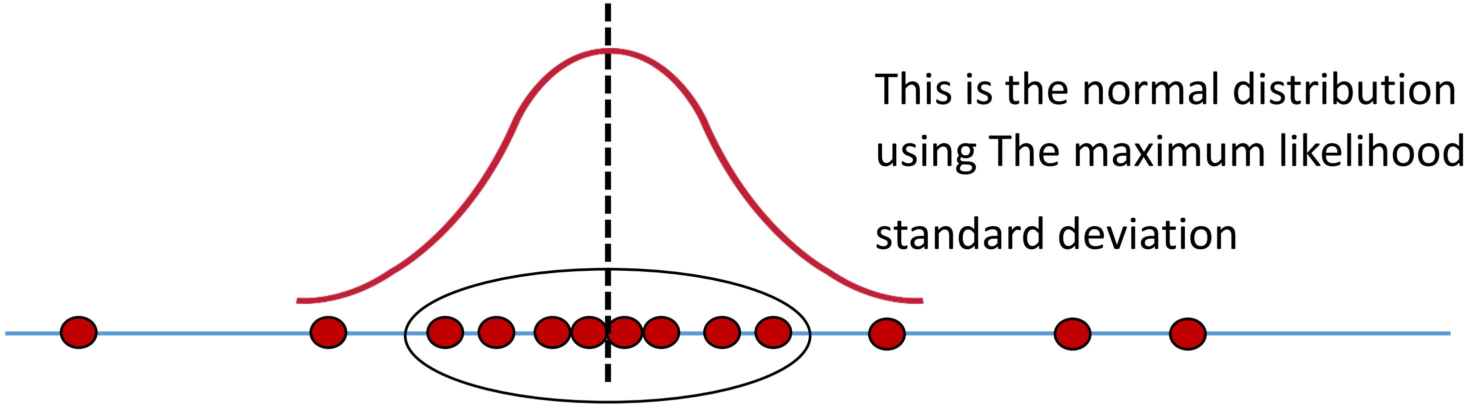
Location of the center of the distribution



Now we have to figure out the "maximum likelihood estimate for the standard deviation"

Maximum Likelihood for Normal Distribution





This is the normal distribution that has been fit to the data by using The maximum likelihood estimations for the mean and standard deviation

Maximum Likelihood for Normal Distribution

In everyday conversion probability and likelihood mean the same thing and likelihood specifically refers to this situation we have covered here; where you are trying to find the optimal value for the mean or standard deviation for a distribution given a bunch of observed measurement

Maximum Likelihood for Binomial Distribution

Maximum likelihood for Binomial distribution

Binomial distribution $= \boldsymbol{pr}(\boldsymbol{x}\backslash \mathrm{n}, \boldsymbol{p}) = (\frac{\boldsymbol{n}!}{\boldsymbol{x}!(\boldsymbol{n}-\boldsymbol{x})!})\boldsymbol{p}^{\boldsymbol{x}}(\mathbf{1}-\boldsymbol{p})^{\boldsymbol{n}-\boldsymbol{x}}$**))**

L$(\boldsymbol{p} = \mathbf{0.5}\backslash \mathrm{n} = \mathbf{7}, \boldsymbol{x} = \mathbf{4}) = (\frac{7!}{\mathbf{4}!(\mathbf{7}-\mathbf{4})!})\mathbf{0.5}^{\mathbf{4}}(\mathbf{1}-\mathbf{0.5})^{\mathbf{7}-\mathbf{4}}$**))=0.273**

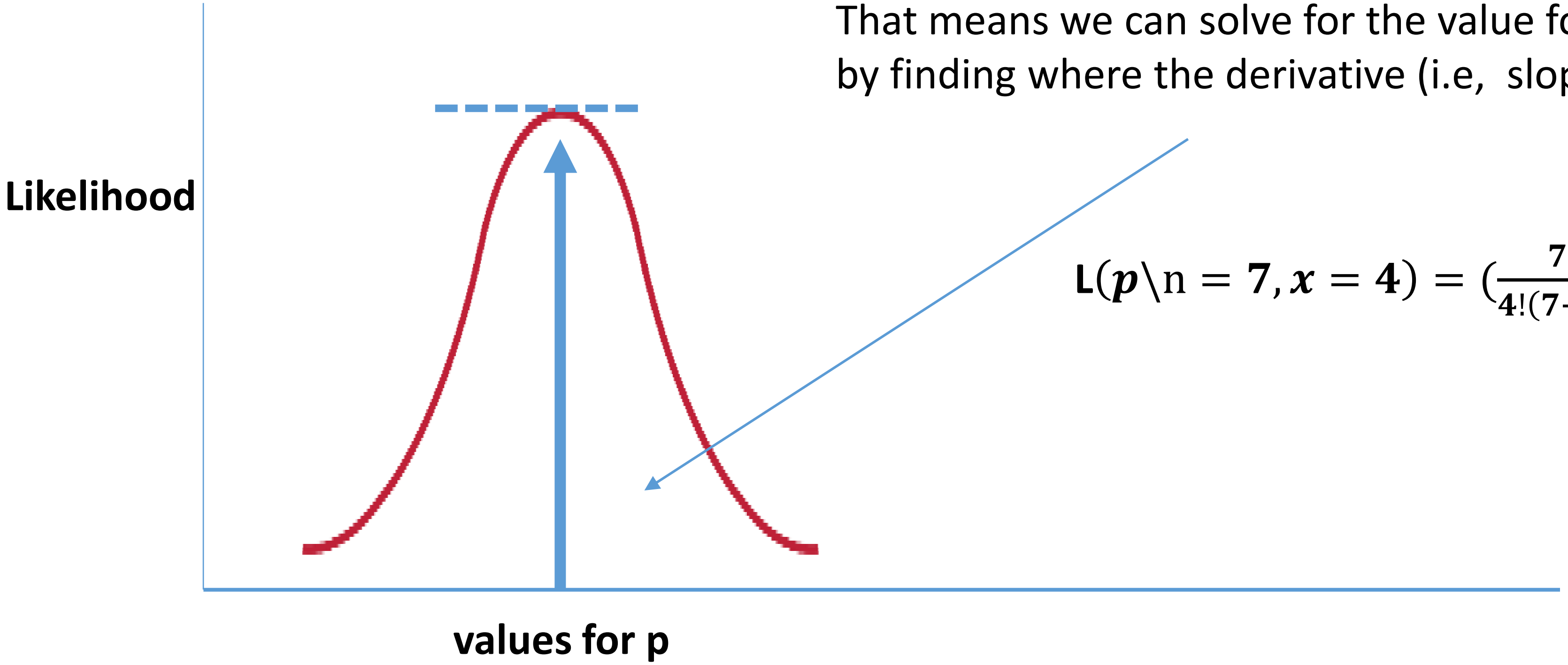
L$(\boldsymbol{p} = \mathbf{0.25}\backslash \mathrm{n} = \mathbf{7}, \boldsymbol{x} = \mathbf{4}) = (\frac{7!}{\mathbf{4}!(\mathbf{7}-\mathbf{4})!})\mathbf{0.25}^{\mathbf{4}}(\mathbf{1}-\mathbf{0.25})^{\mathbf{7}-\mathbf{4}}$**))=0.058**

L$(\boldsymbol{p} = \mathbf{0.57}\backslash \mathrm{n} = \mathbf{7}, \boldsymbol{x} = \mathbf{4}) = (\frac{7!}{\mathbf{4}!(\mathbf{7}-\mathbf{4})!})\mathbf{0.57}^{\mathbf{4}}(\mathbf{1}-\mathbf{0.57})^{\mathbf{7}-\mathbf{4}}$**))=0.294**

Maximum likelihood for Binomial distribution

Binomial distribution$= \boldsymbol{pr(x \backslash \mathrm{n}, p) = (\frac{n!}{x!(n-x)!})p^{x}(1-p)^{n-x}}$**)**

L$(\boldsymbol{p} = \mathbf{0.57} \backslash \mathrm{n} = \mathbf{7}, \boldsymbol{x} = \mathbf{4}) = (\frac{\mathbf{7!}}{\mathbf{4!(7-4)!}})\mathbf{0.57^{4}(1-0.57)^{7-4}}$**)=0.294**



That means we can solve for the value for p that results in the maximum likelihood by finding where the derivative (i.e, slope) is equal to o

L$(\boldsymbol{p} \backslash \mathrm{n} = \mathbf{7}, \boldsymbol{x} = \mathbf{4}) = (\frac{\mathbf{7!}}{\mathbf{4!(7-4)!}})\boldsymbol{p}^{\mathbf{4}}(\mathbf{1} - \boldsymbol{p})^{\mathbf{7-4}}$**)**

Maximum likelihood for Binomial distribution

Binomial distribution$= pr(x\backslash \mathrm{n}, p) = (\frac{n!}{x!(n-x)!})p^{x}(1-p)^{n-x}))$

$$ln\ \mathrm{L}(p = 0.57\backslash \mathrm{n} = 7, x = 4) = ln\left[(\frac{n!}{x!(n-x)!})p^{x}(1-p)^{n-x}))\right]$$

We do this because the original likelihood function and its log will both reach the maximum using the same value for p and its way easier to take the derivate of the log of the likelihood function compared to the original function

$$ln\ \mathrm{L}(p = 0.57\backslash \mathrm{n} = 7, x = 4) = ln\left(\frac{7!}{4!(7-4)!}\right) + ln\left(p^{4}\right) + ln\left[(1-p)^{7-4})\right]$$

$$ln\ \mathrm{L}(p = 0.57\backslash \mathrm{n} = 7, x = 4) = ln\left(\frac{7!}{4!(7-4)!}\right) + ln\left(p^{4}\right) + ln\left[(1-p)^{7-4})\right]$$

Maximum likelihood for Binomial distribution

$$d\ln \mathrm{L}(p = 0.57 \backslash \mathrm{n}/dp = 7, x = 4) = p = 4/7$$

This formula will give us the maximum likelihood estimate for p when there x “successes” in n “trials”

This formula will give us the maximum likelihood estimate for p is x, the number of “successes” divided by n the total number of “trails”

Multinomial Distribution

Multinomial Distribution

Multinomial Distribution Example

Type of blood group	O	A	A	AB
probability	0.44	0.42	0.10	0.04

In the random sample of 10 peoples what is the probability of 6 have O, 2 have A and 1 has B and 1 has AB

$$P = (X_1 = x_1, \dots, X_k = x_k) = \frac{n!}{x_1! \dots x_k!} {p_1}^{x_1}\, {p_2}^{x_2} \dots {p_k}^{x_k}$$

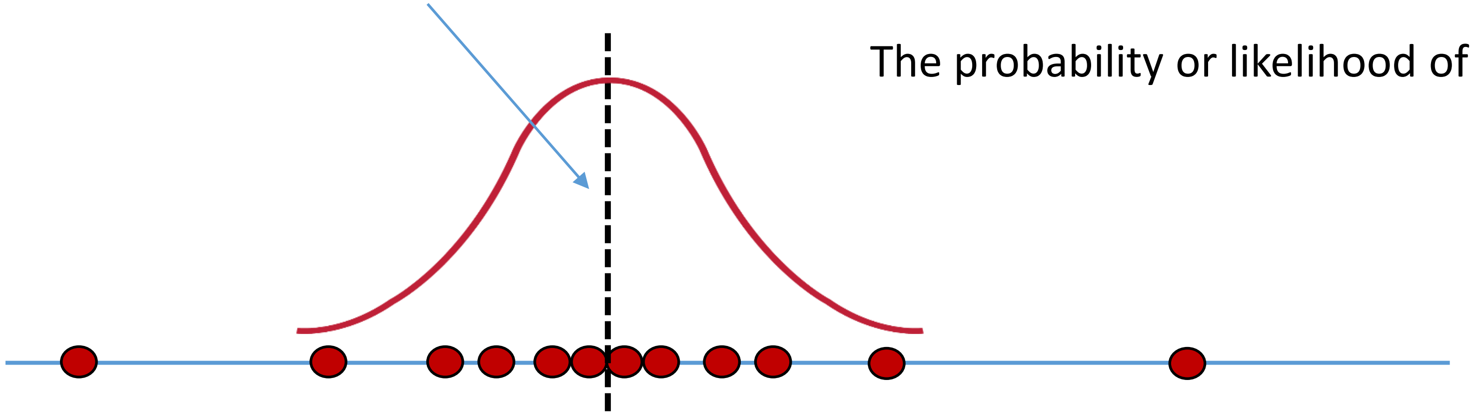
$$P = (X_1 = 6, X_2 = 2, X_3 = 1, X_4 = 1) = \frac{10!}{6!2!1!1!} 0.44^6\, 0.42^2\, 0.10^1\, 0.04^1$$

Maximum likelihood for Normal Distribution (Mathematical Verification)

Maximum Likelihood for Normal Distribution

According to normal distribution with mean here

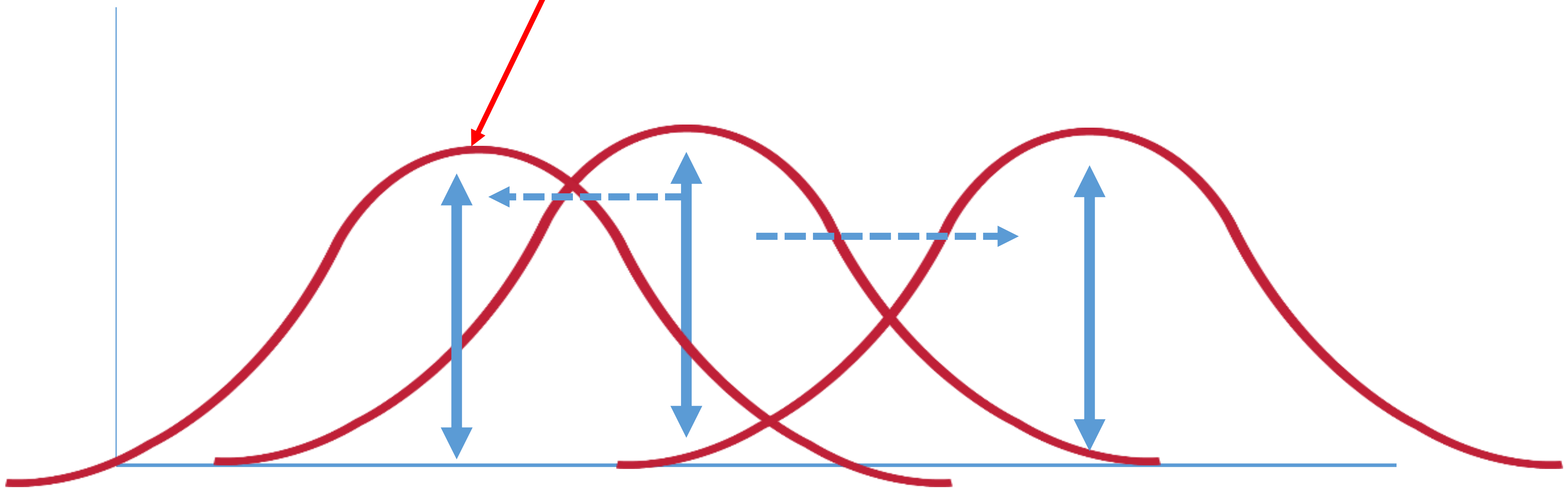
The probability or likelihood of observing these weights is high



Maximum Likelihood for Normal Distribution

First parameter, the Greek letter μ determines the location of the normal distributions mean A smaller value for μ moves the mean of the distribution to the left

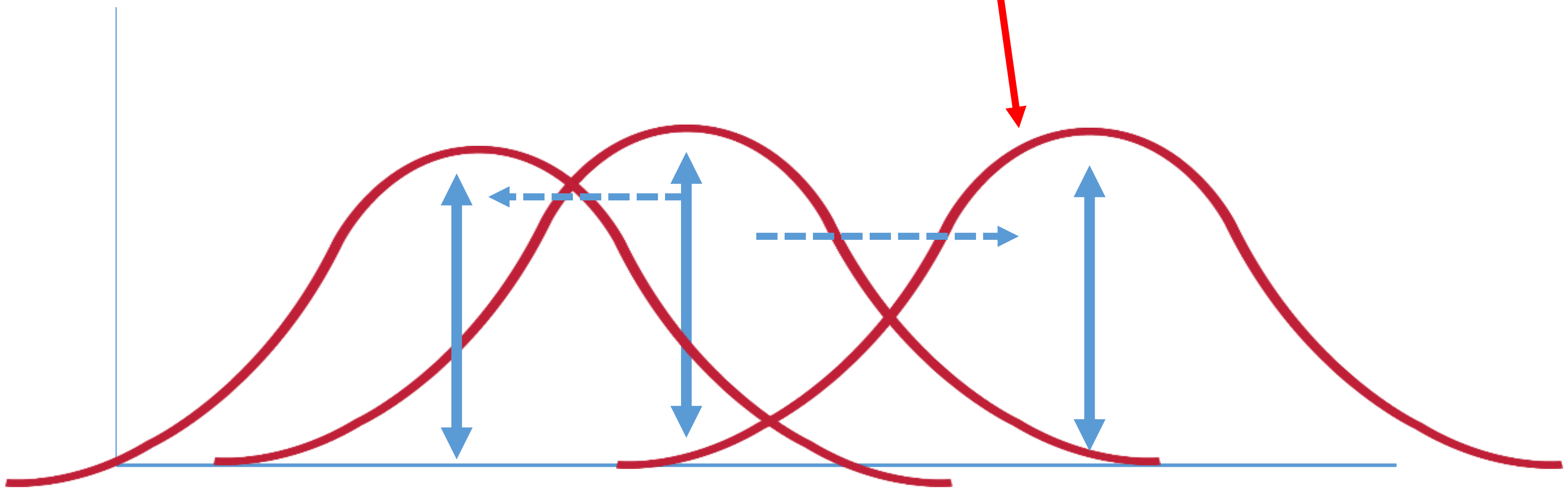
$$\mathrm{pr}(x\backslash\mu,\sigma)\ \frac{1}{\sqrt{2\pi\sigma^2}}\,e^{-(x-\mu)^2/2\sigma^2)}$$



Maximum Likelihood for Normal Distribution

First paramet and a larger value for μ moves the mean of the distribution to the right

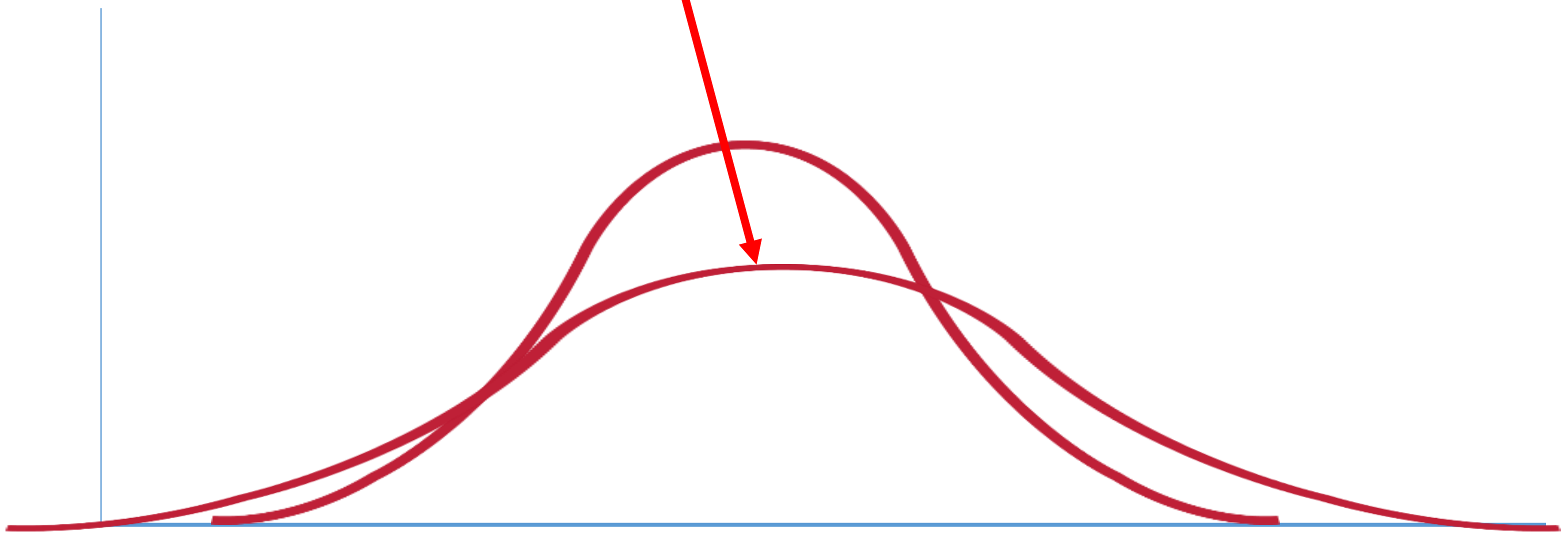
$$\mathrm{pr}(\mathrm{x}\backslash\mu,\sigma)\,\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2)}$$



Maximum Likelihood for Normal Distribution

the seccond parameret, the greek charecter σ is the standard deviation and dteremines the normal distributions width and a larger value for sigma makes the normal curve shorter and wider

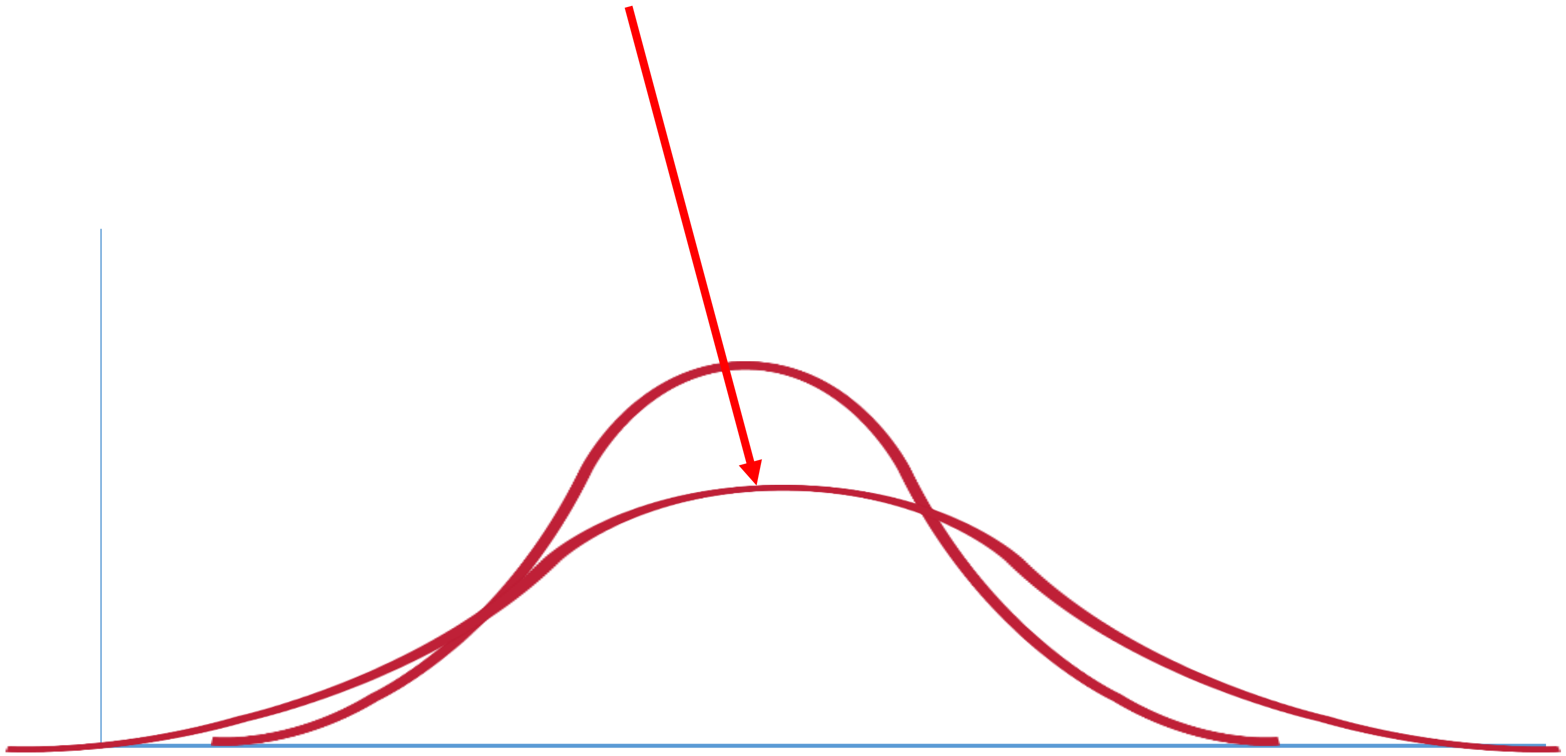
$$\text{pr}(\text{x}\backslash\mu,\sigma)\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2)}$$



Maximum Likelihood for Normal Distribution

a larger value for sigma makes the normal curve shorter and wider

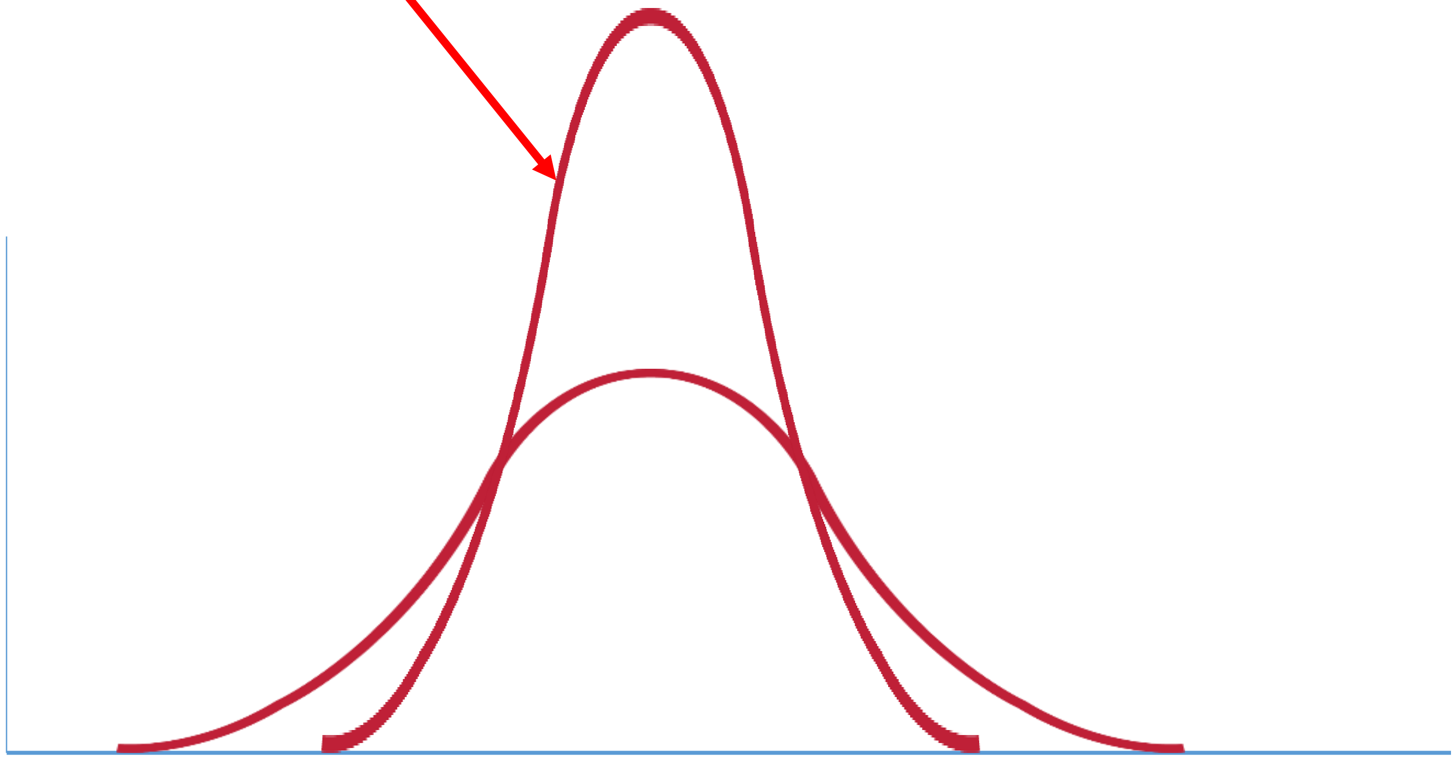
$$\mathrm{pr}(\mathrm{x}\backslash\mu,\sigma)\,\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2)}$$



Maximum Likelihood for Normal Distribution

A larger value for sigma makes the normal curve shorter and wider and a smaller value for sigma makes the normal curve taller and narrower

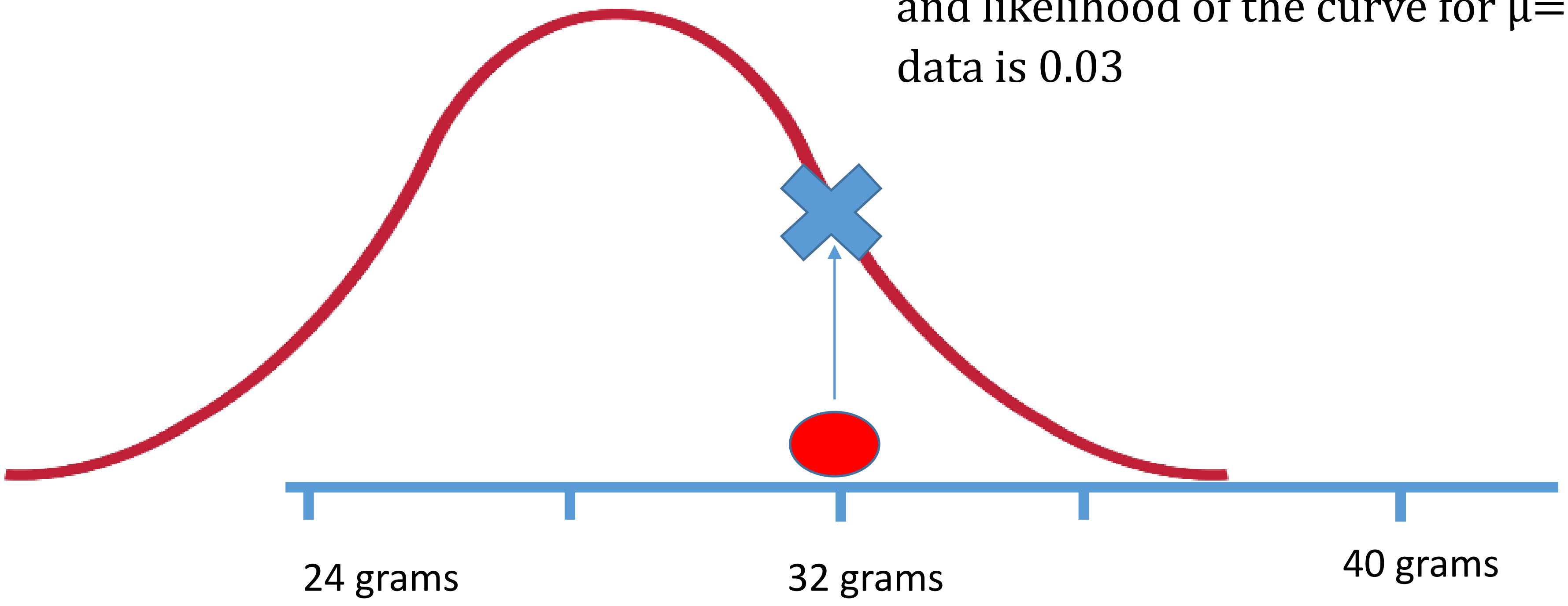
$$\mathrm{pr}(x \backslash \mu, \sigma)\, \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2)}$$



Maximum Likelihood for Normal Distribution

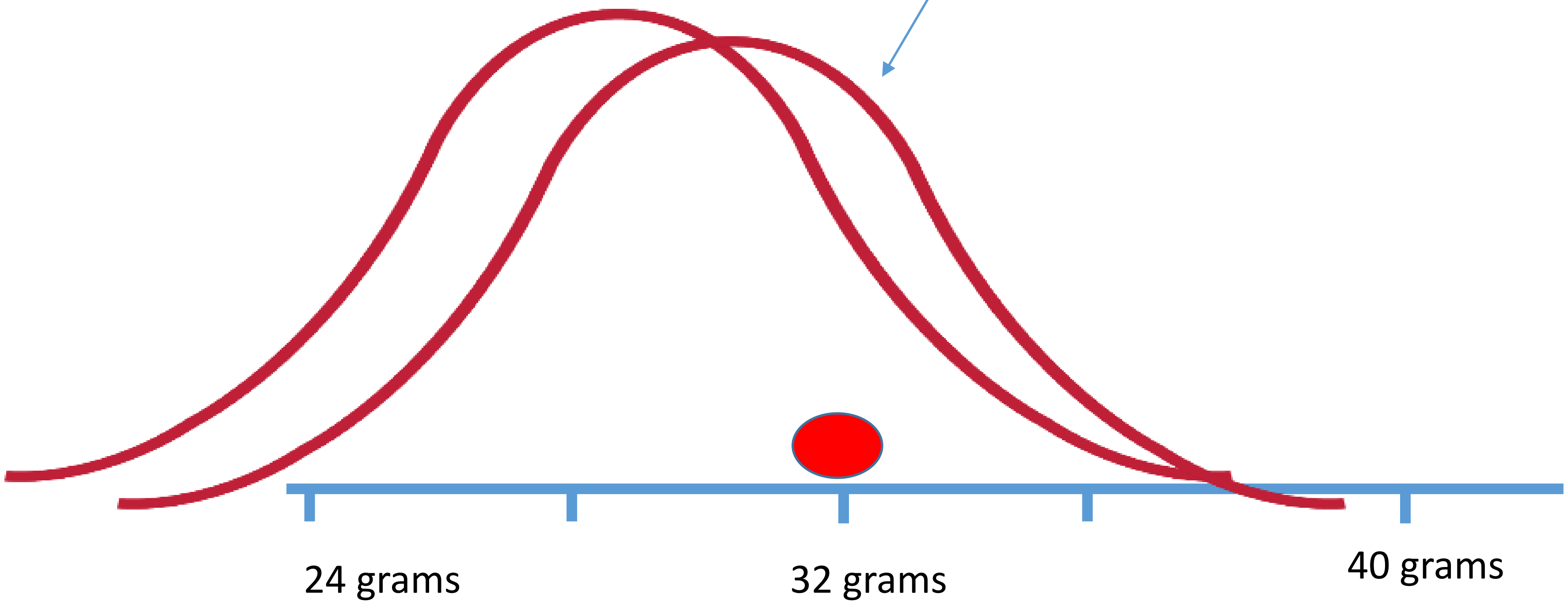
$$L(\mu = 28, \sigma = 2 \backslash x = 32) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-30)^2/22^2} = 0.03$$

and likelihood of the curve for μ=28 and $\sigma = 2$ given the data is 0.03



Maximum Likelihood for Normal Distribution

now we can shift the distribution a little bit to the right by setting $\mu = 30$ and then calculate the likelihood

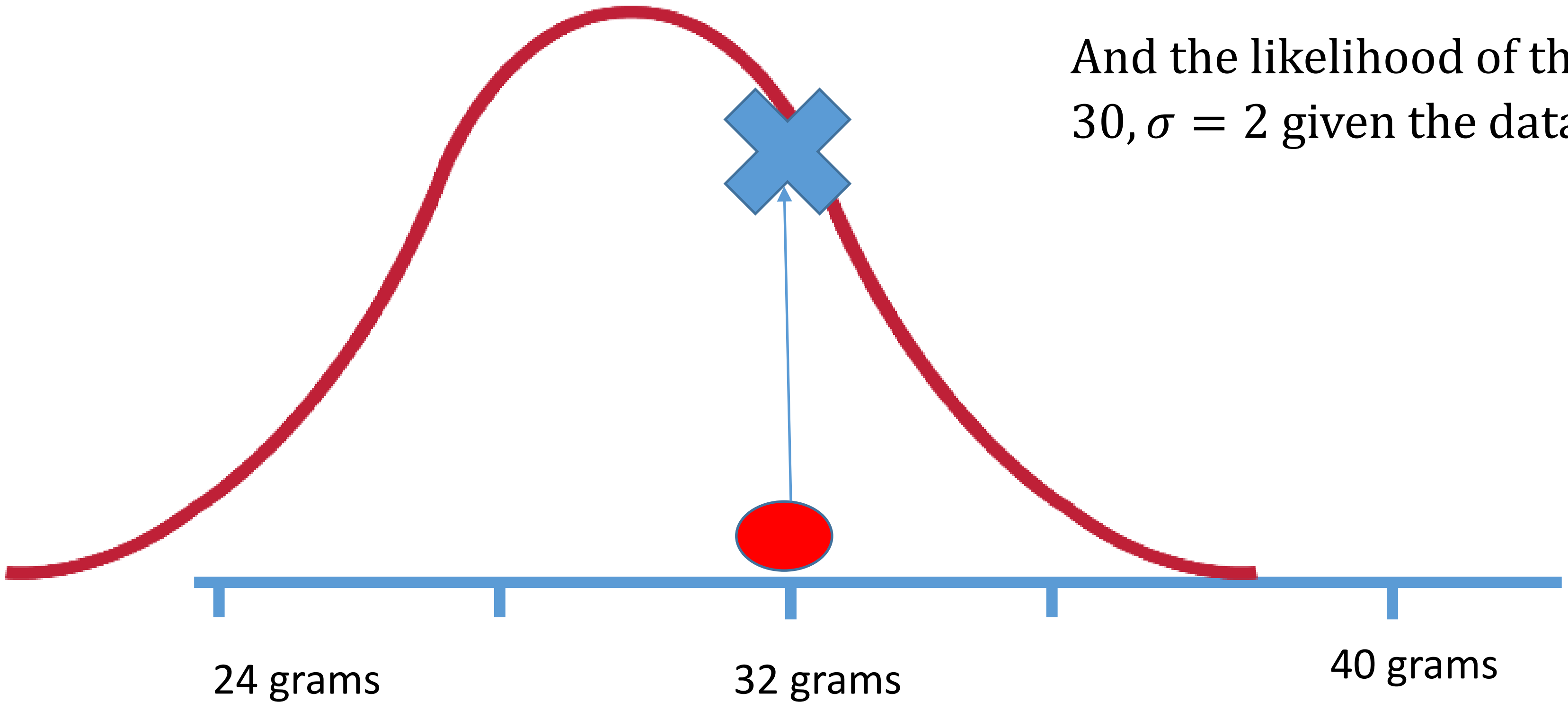


Maximum Likelihood for Normal Distribution

$$L(\mu = 30, \sigma = 2 \backslash x = 32) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-30)^2/22^2)} = 0.12$$

Again, we just plug the numbers into the likelihood function

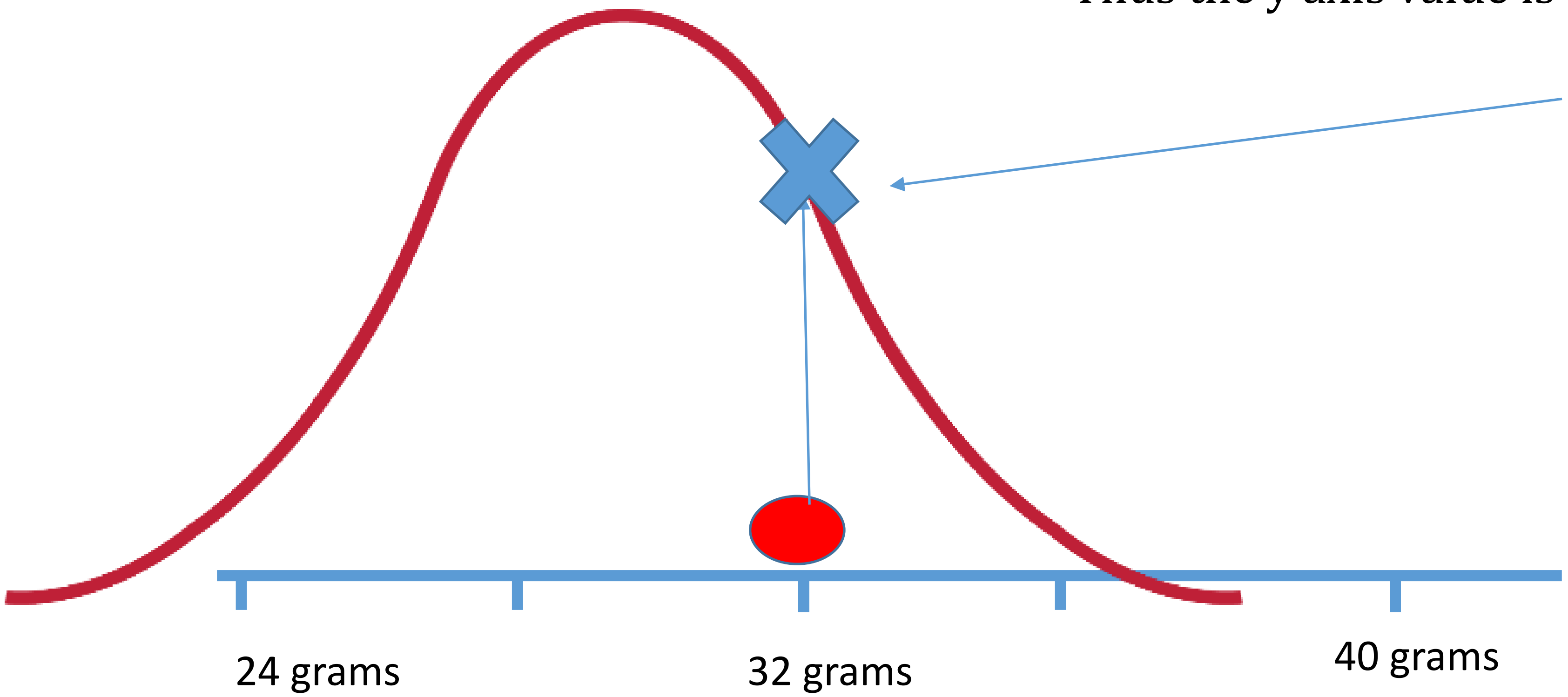
And the likelihood of the new curve with $\mu = 30, \sigma = 2$ given the data is 0.12



Maximum Likelihood for Normal Distribution

$$L(\mu = 30, \sigma = 2\backslash x = 32) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-30)^2/22^2)} = 0.12$$

Thus the y axis value is 0.12

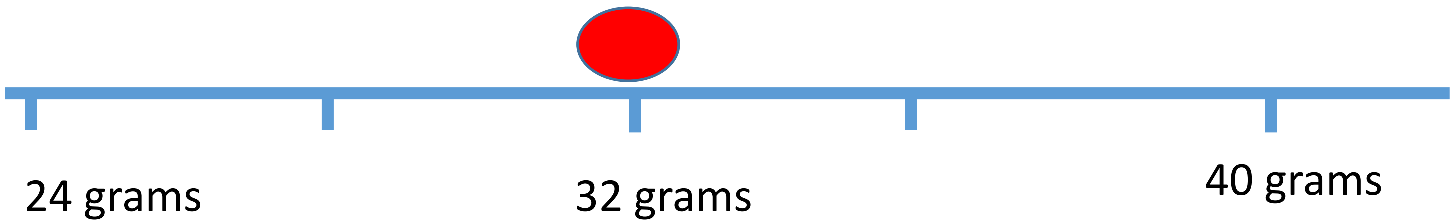


Maximum Likelihood for Normal Distribution

$$L(\mu \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-\mu)^2/22^2} = 0.12$$

Again, we just plug the numbers into the likelihood function. if we decide to fix σ =2, so that it is a given , just like the data…

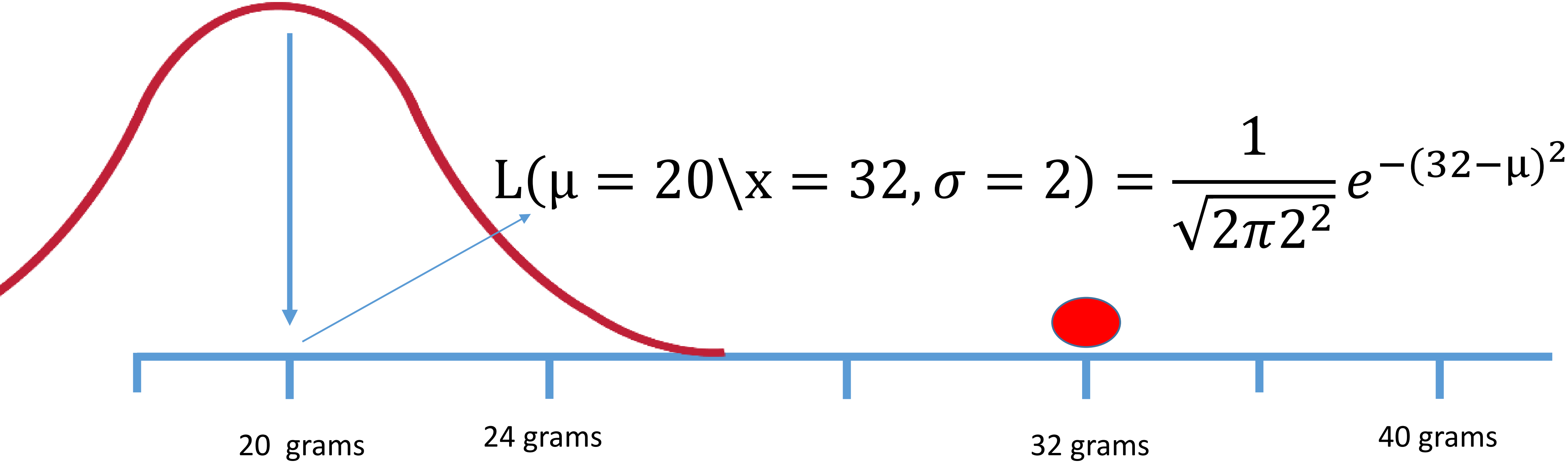
Then, we can plug in a whole bunch of values for μ and see which one gives the maximum likelihood



Maximum Likelihood for Normal Distribution

For example, if we start with the mean of the distribution over here on the left at 20 grams…

$$L(\mu = 20 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-\mu)^2/2 2^2)}$$

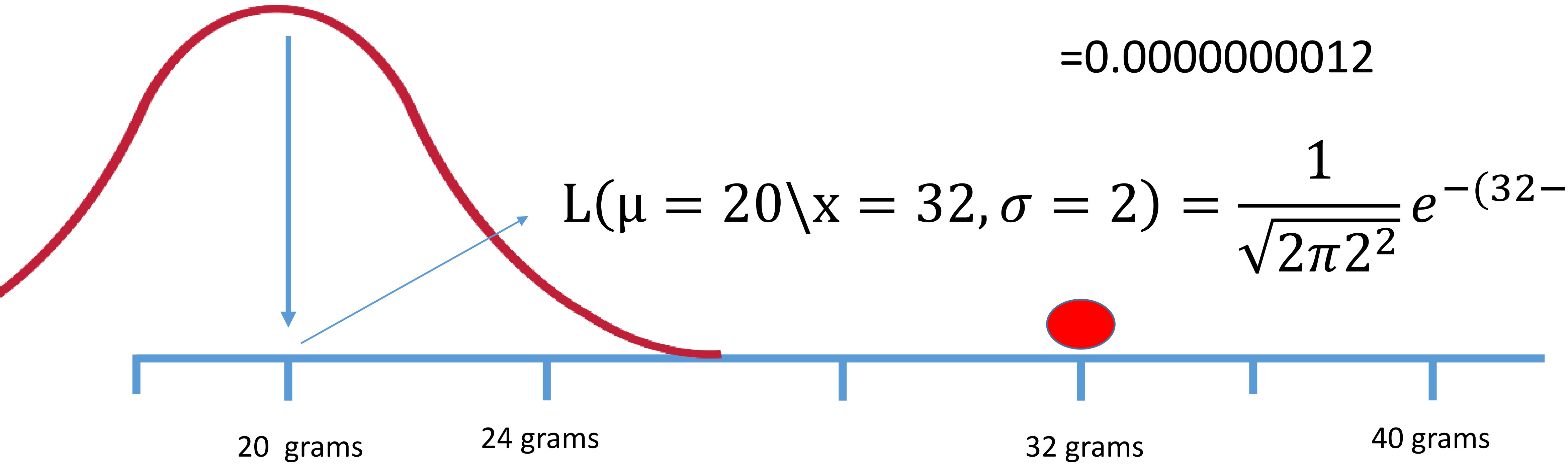


Maximum Likelihood for Normal Distribution

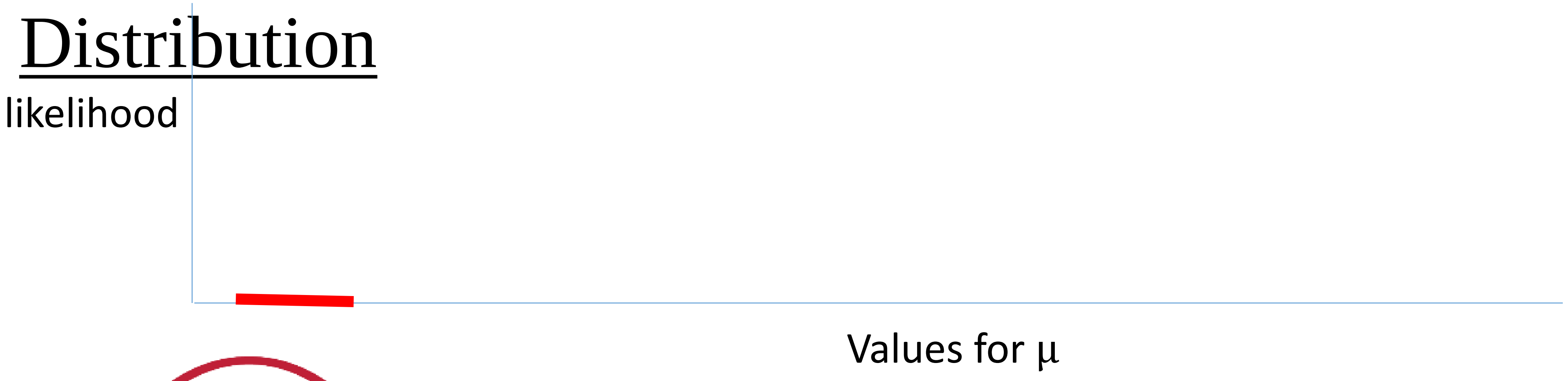
And we get a small likelihood

=0.0000000012

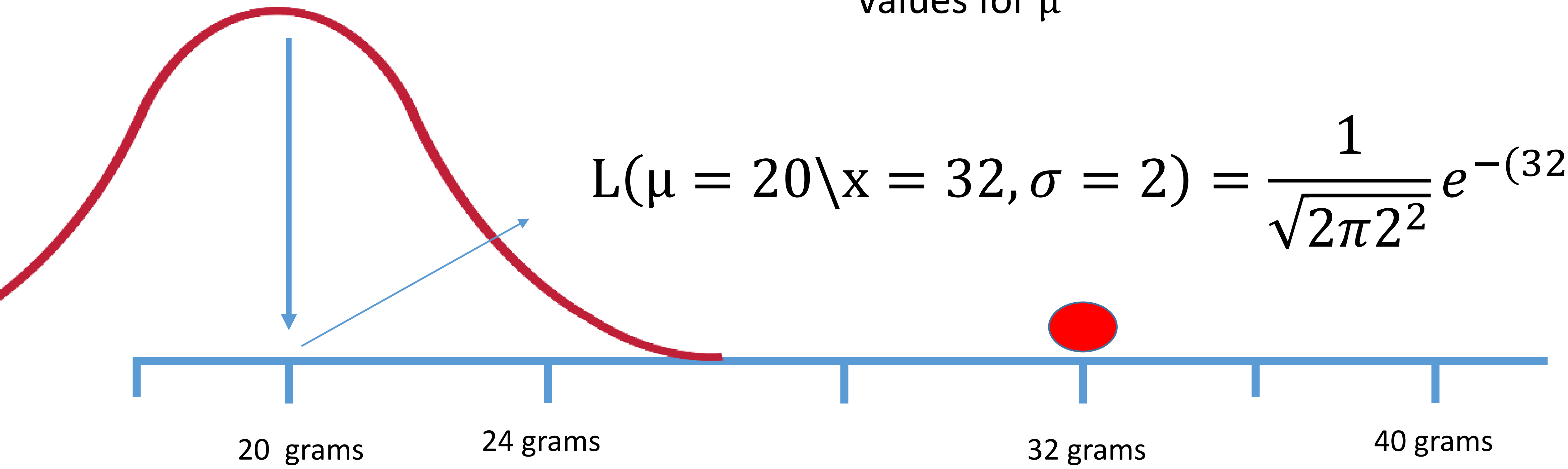
$$L(\mu = 20 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-20)^2/22}$$



Maximum Likelihood for Normal Distribution

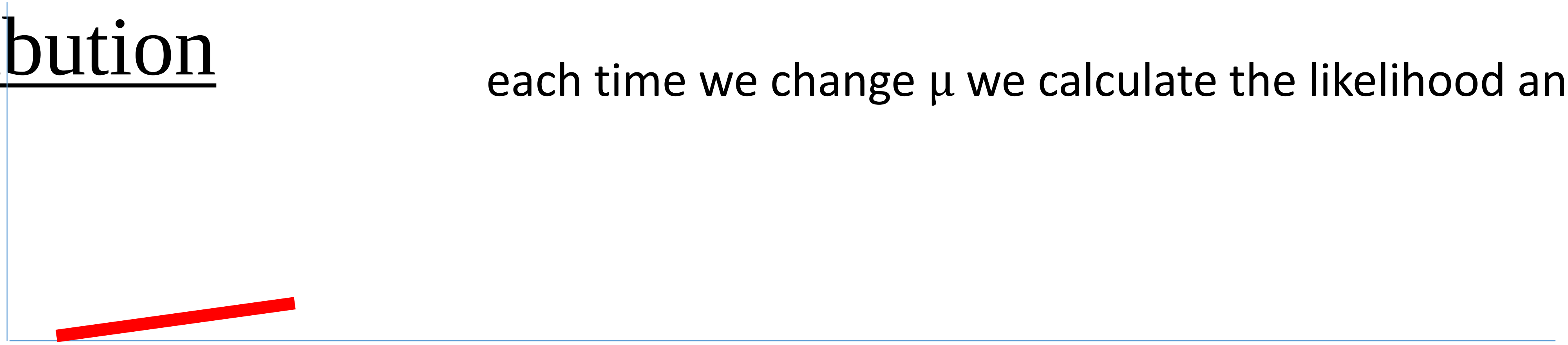


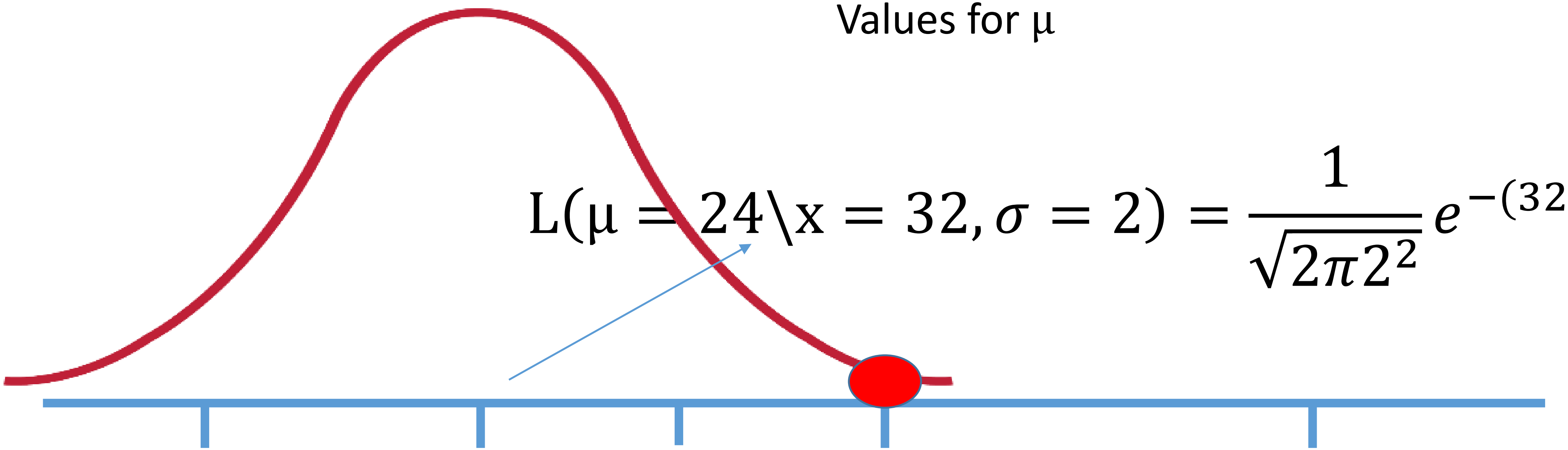
$$L(\mu = 20 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-20)^2/}$$



Maximum Likelihood for Normal Distribution

each time we change μ we calculate the likelihood and plot it

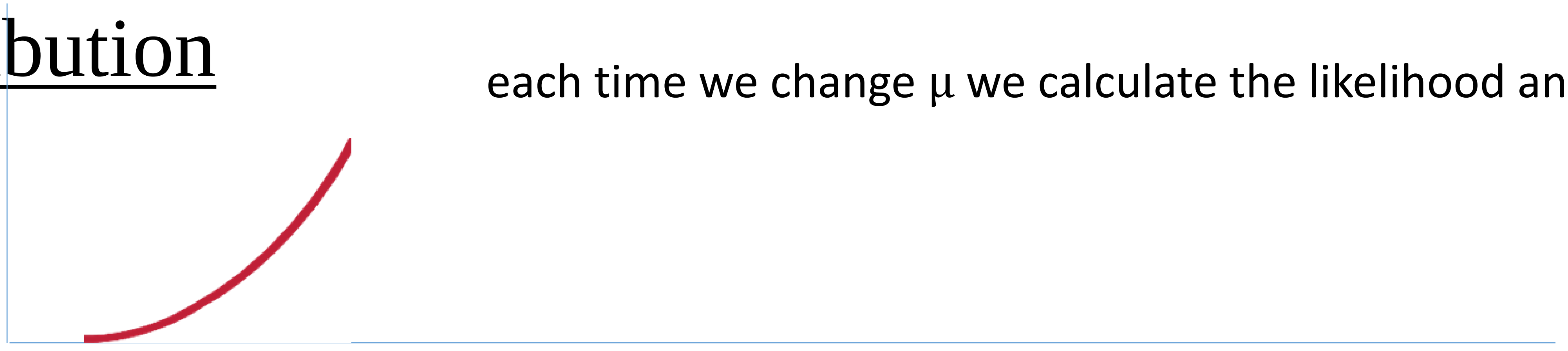


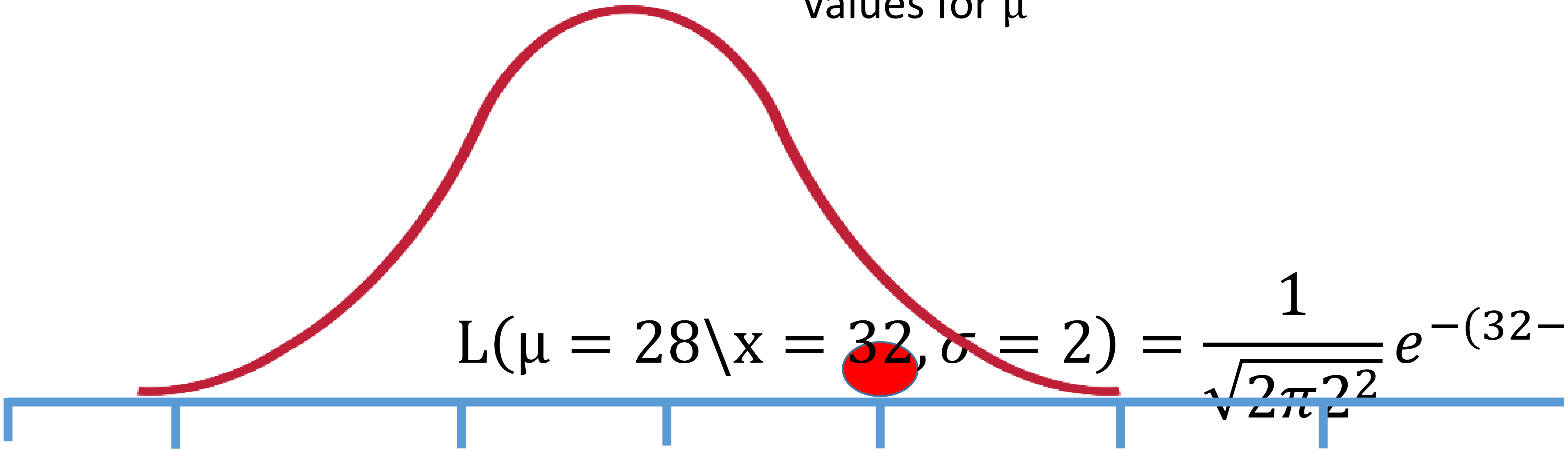


$$L(\mu = 24 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-24)^2/22}$$

Maximum Likelihood for Normal Distribution

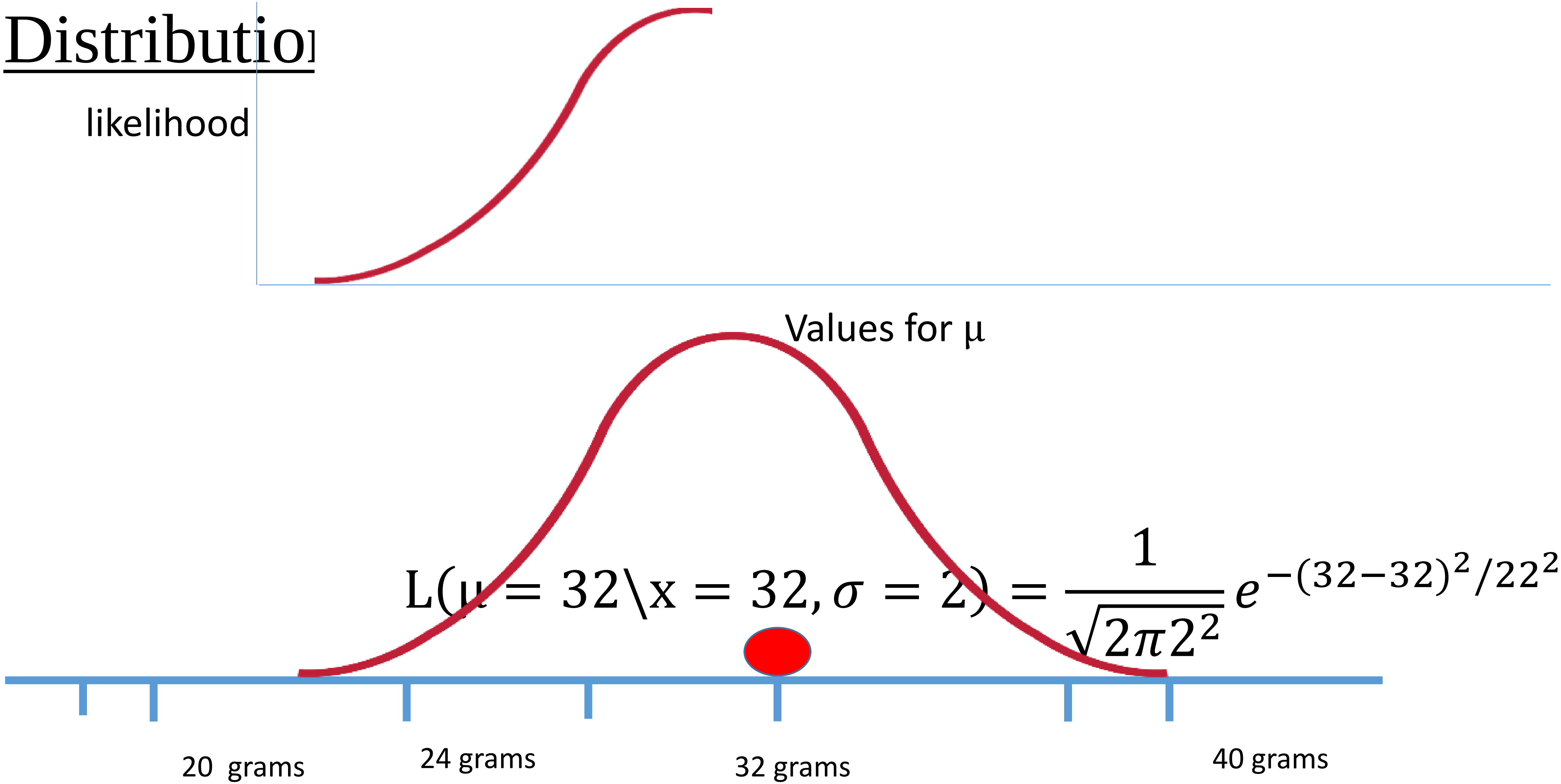
each time we change μ we calculate the likelihood and plot it





$$L(\mu = 28 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-28)^2/2 2^2}$$

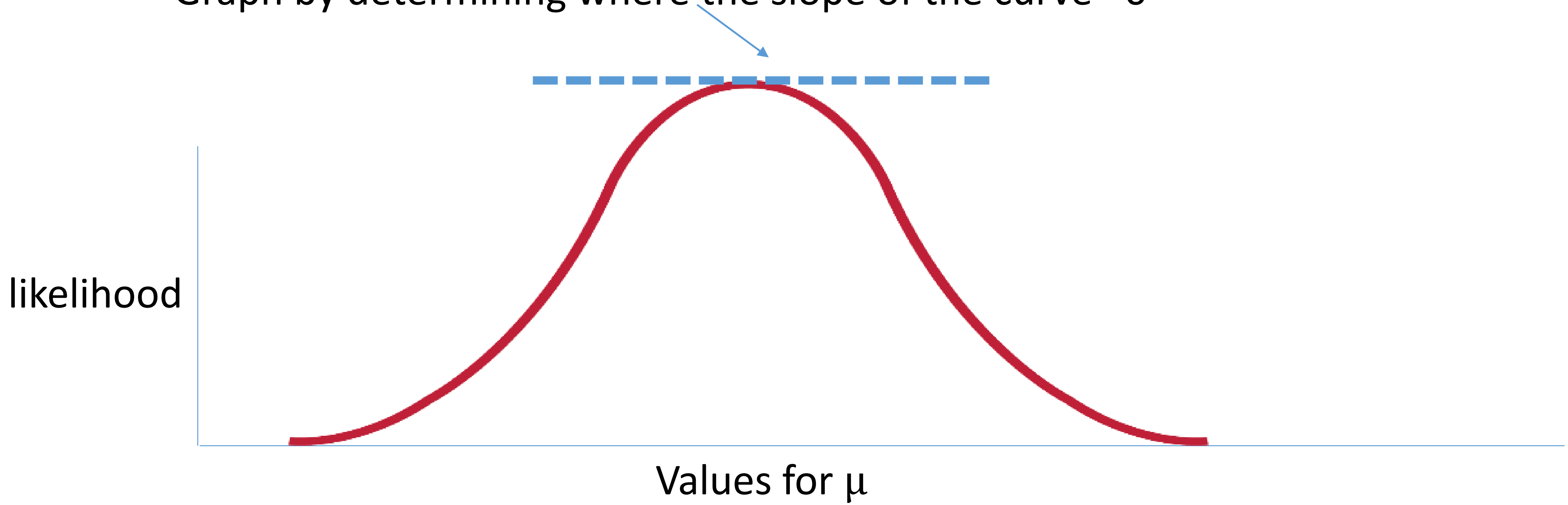
Maximum Likelihood for Normal Distributio



$$L(\mu = 32 \backslash x = 32, \sigma = 2) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-32)^2/22^2)}$$

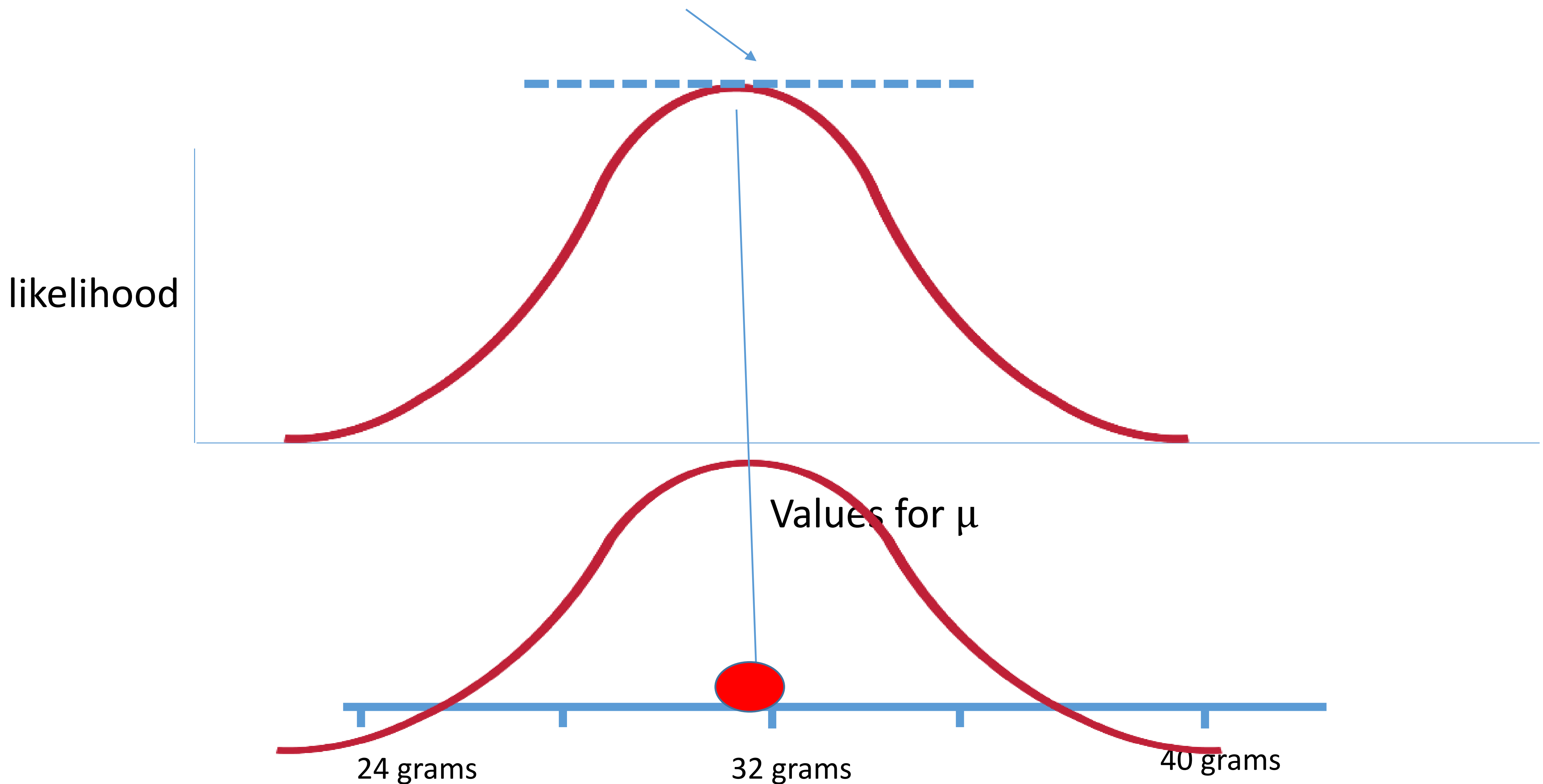
Maximum Likelihood for Normal Distribution

Each time we change μ we calculate we can identify the peak in the likelihood Graph by determining where the slope of the curve =0



Maximum Likelihood for Normal Distribution

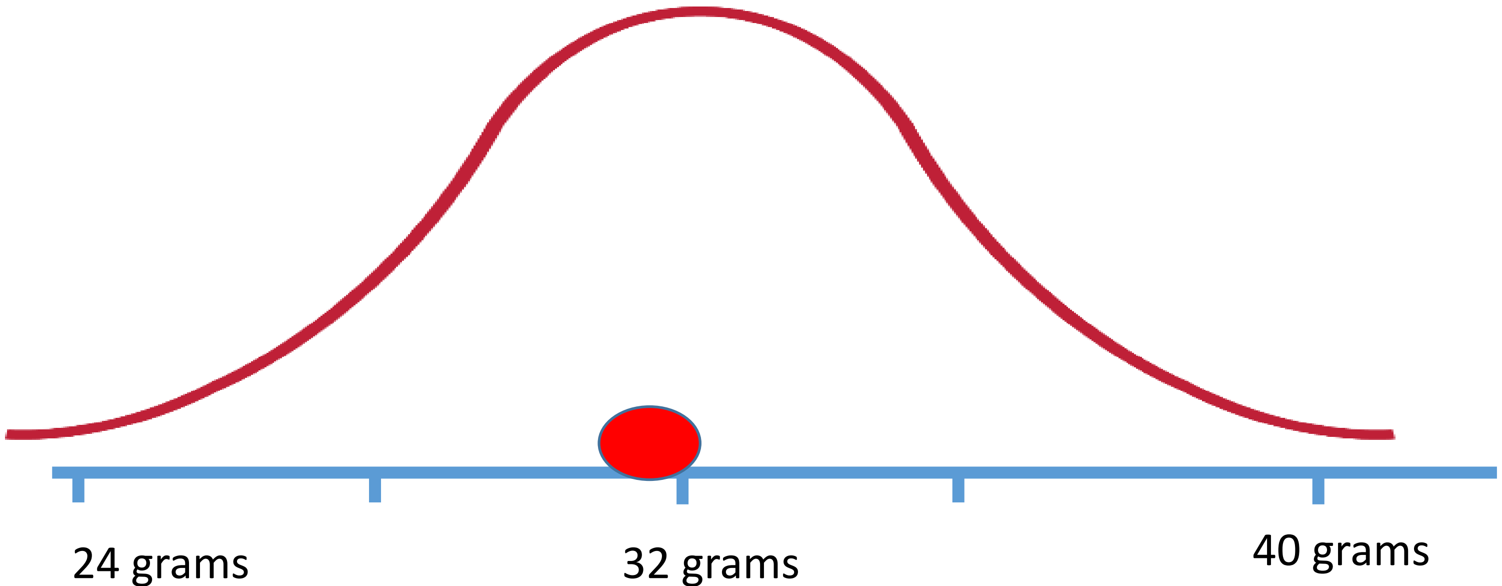
In this case the slope =0 when μ=32



Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\sigma \backslash \mathrm{x} = 32, \mu = 32) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(32-32)^2/2\sigma^2)}$$

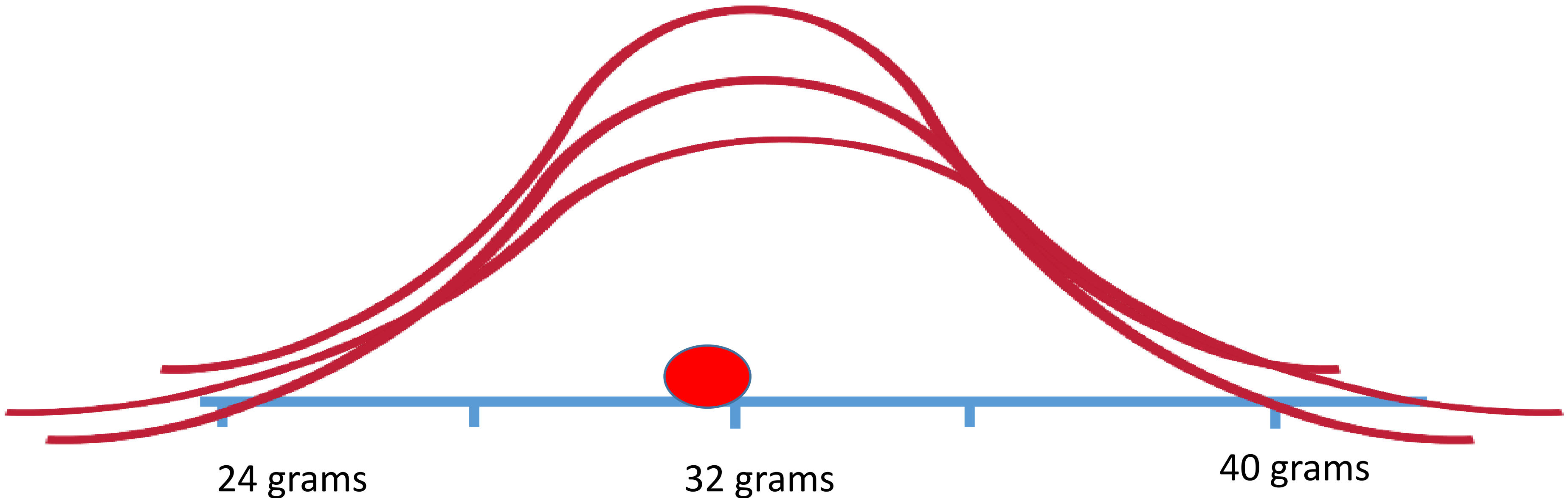
Now we can fix μ=32 and treat it like a given just like the data



Maximum Likelihood for Normal Distribution

$$L(\sigma \backslash x = 32, \mu = 32) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(32-32)^2/2\sigma^2)}$$

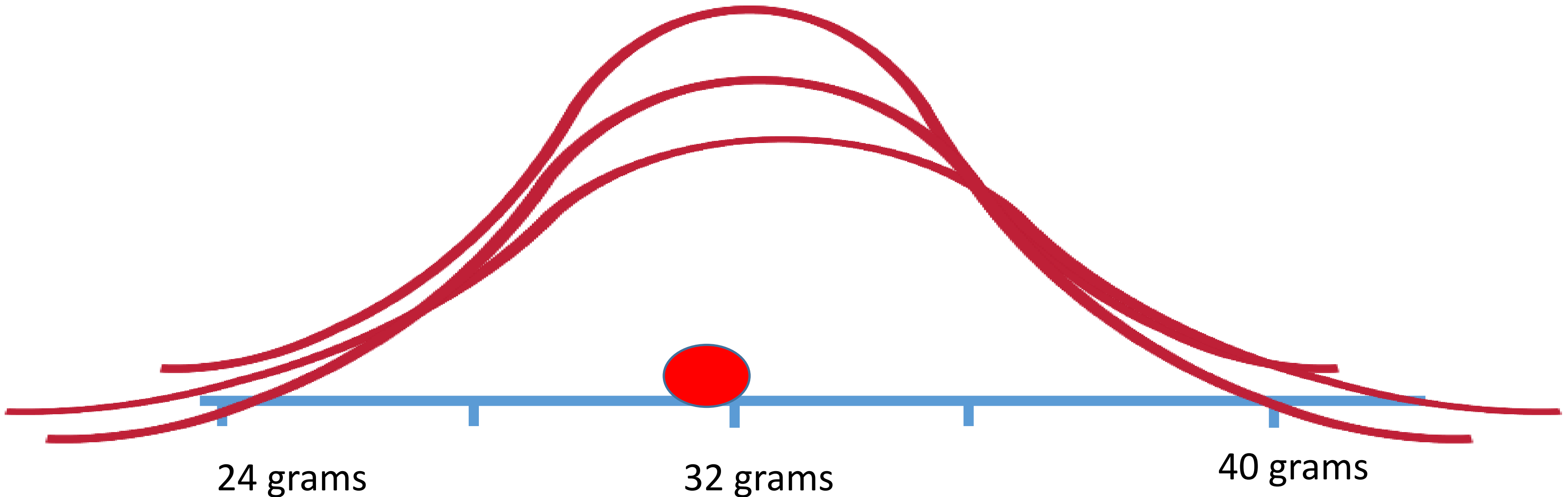
And we can plug in different values for σ to find the one that gives the maximum likelihood



Maximum Likelihood for Normal Distribution

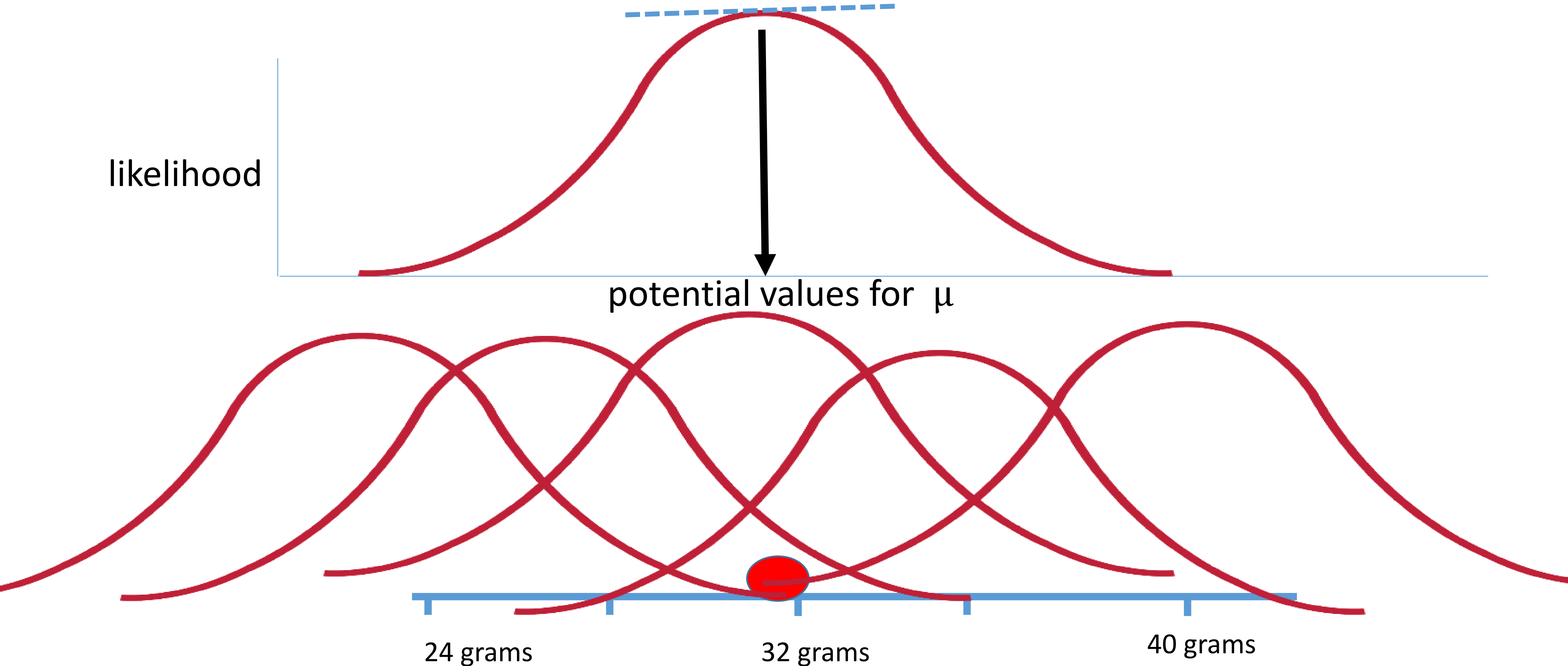
$$L(\sigma \backslash x = 32, \mu \ = 32) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(32-32)^2/2\sigma^2)}$$

This simple example shows that how to find the maximum likelihood estimates for μ and σ

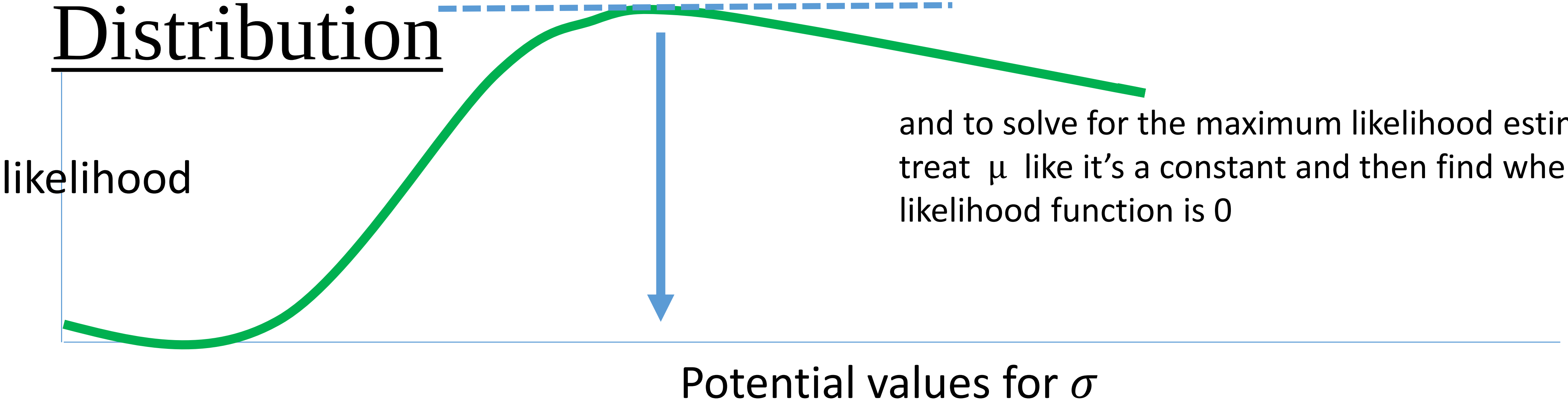


Maximum Likelihood for Normal Distribution

To solve for the maximum likelihood estimate for μ, we treat σ like it's a constant and then find where the slope of its likelihood function is 0

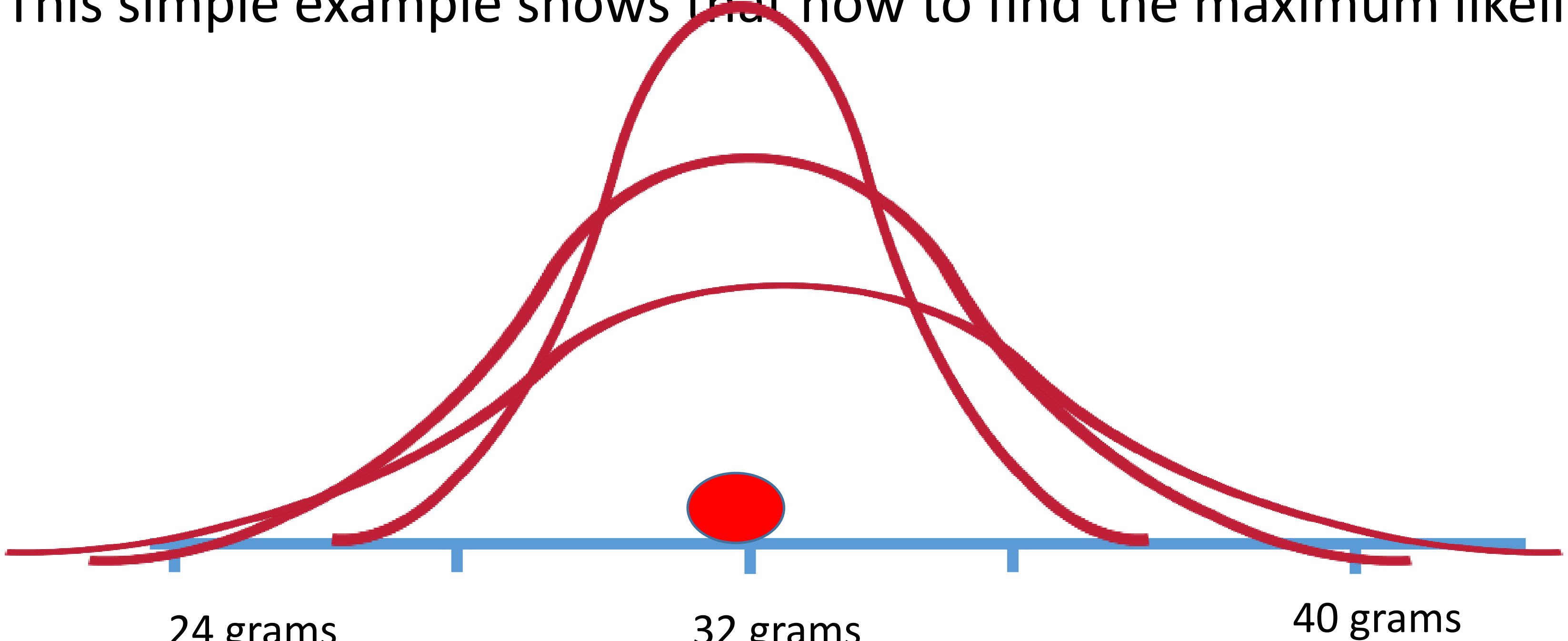


Maximum Likelihood for Normal Distribution



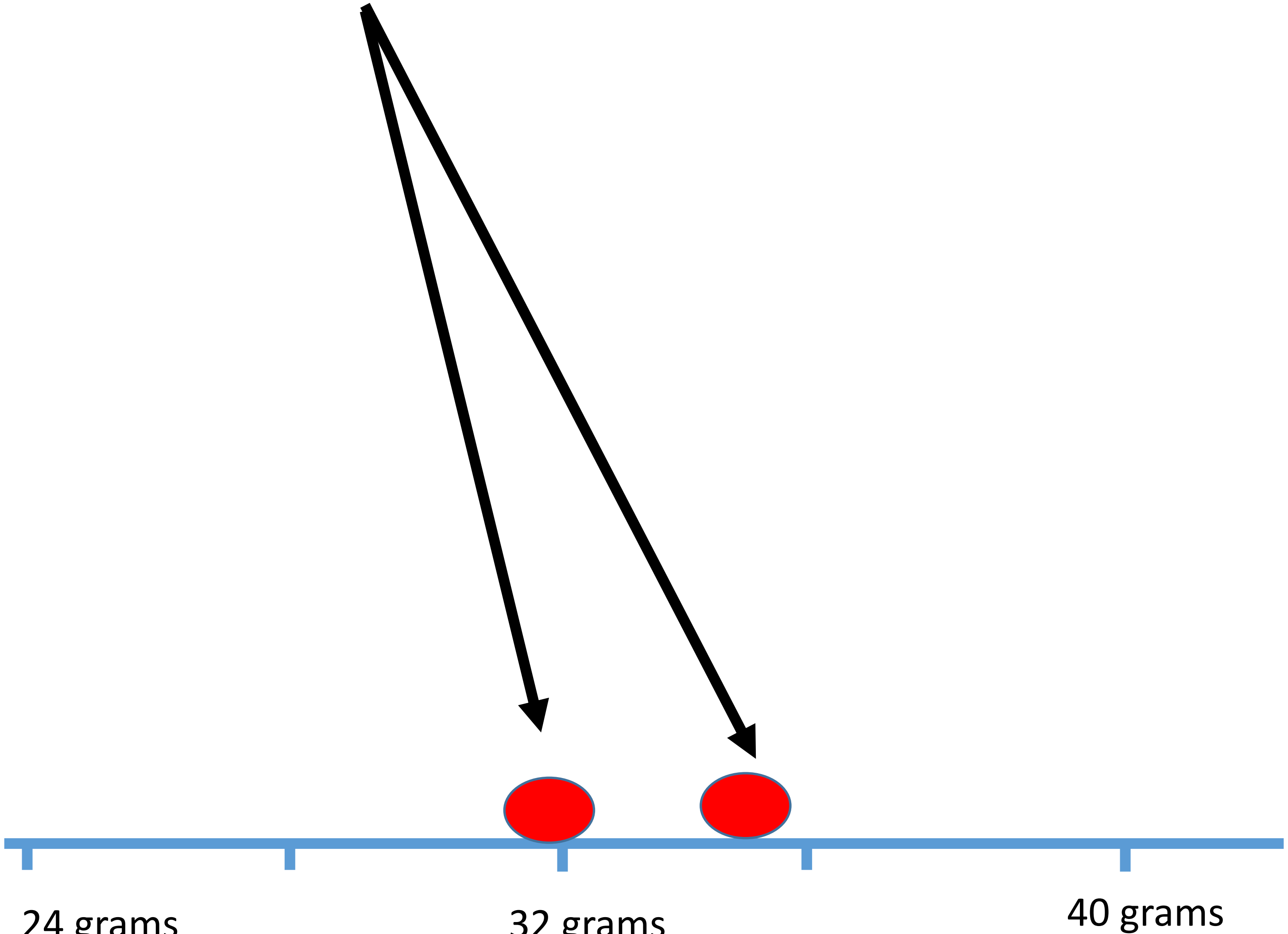
and to solve for the maximum likelihood estimate for σ, we treat μ like it's a constant and then find where the slope of its likelihood function is 0

This simple example shows that how to find the maximum likelihood estimates for μ and σ



Maximum Likelihood for Normal Distribution

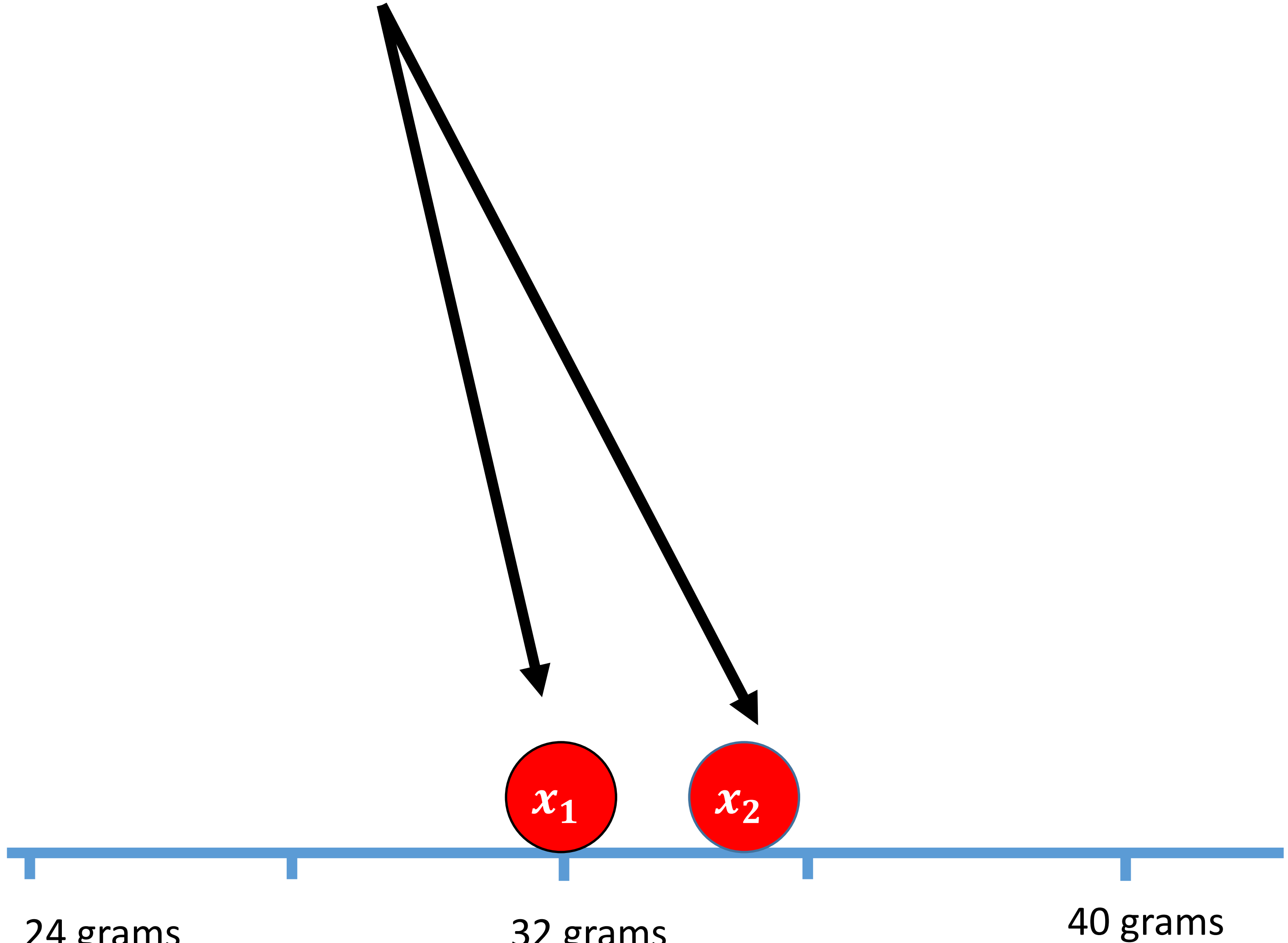
lets use a two sample dataset to calculate the likelihood of a normal distribution



Maximum Likelihood for Normal Distribution

~~lets call the~~ mouse that weight 32 grams

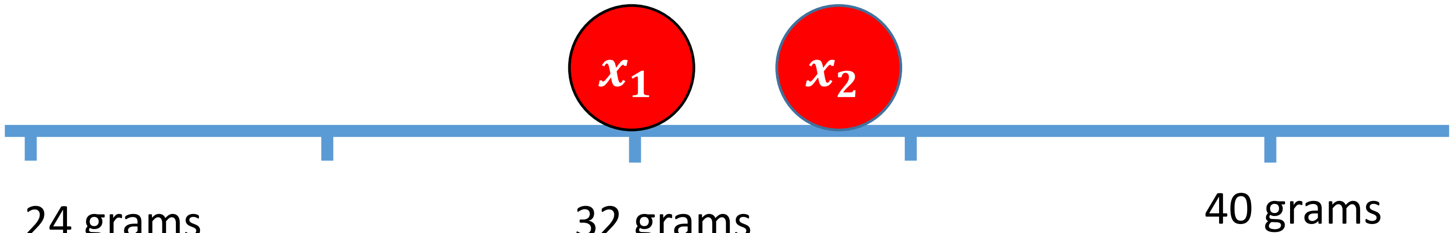
x_1 *and mouse that weighs* 34 *grams* x_2



Maximum Likelihood for Normal Distribution

We have already seen how to calculate the likelihood for this curve given $\boldsymbol{x_1}$, the mouse weights 32 grams…

$$\mathrm{L}(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32) = \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-30)^2/22^2} = 0.03$$



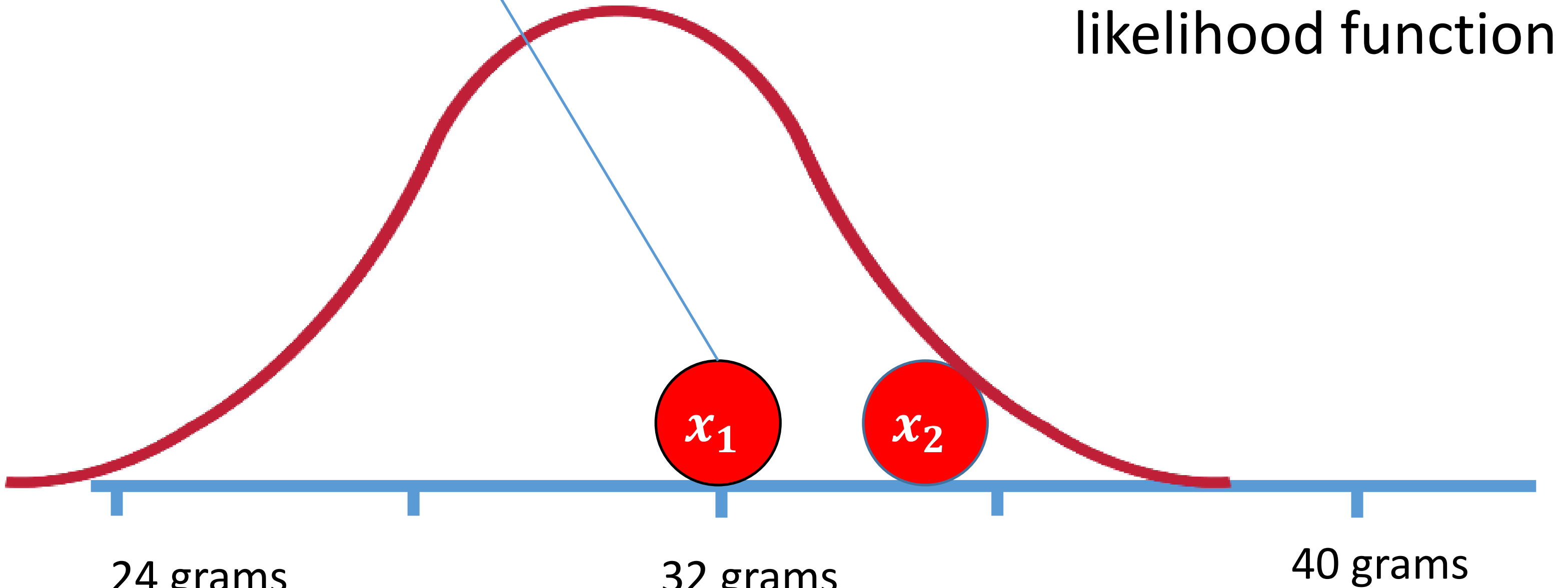
Maximum Likelihood for Normal Distribution

we have already seen how to calculate the likelihood for this curve given $\boldsymbol{x_1}$, the mouse weights 32 grams…

$$L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32)$$

$$L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_2} = 34)$$

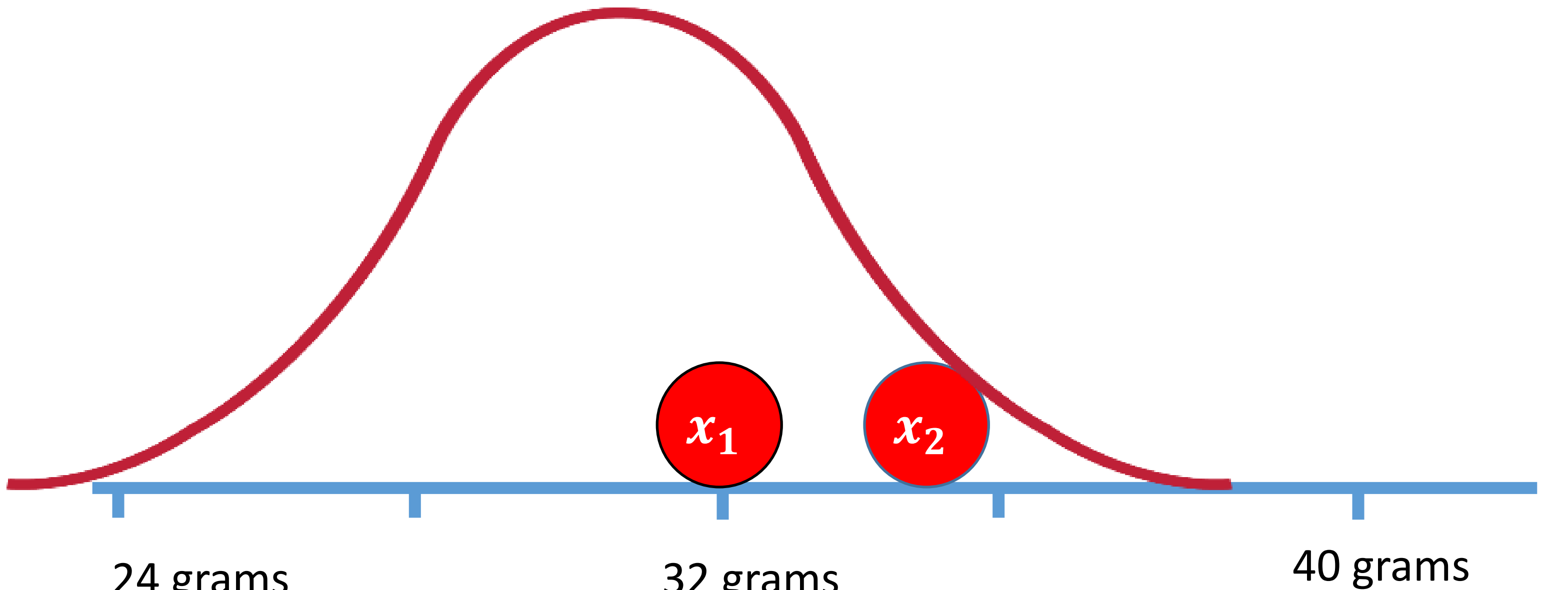
and we can calculate the likelihood for this curve given $\boldsymbol{x_2}$ by plugging the **34** into likelihood function



Maximum Likelihood for Normal Distribution

$L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32)$ $L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_2} = 34)$

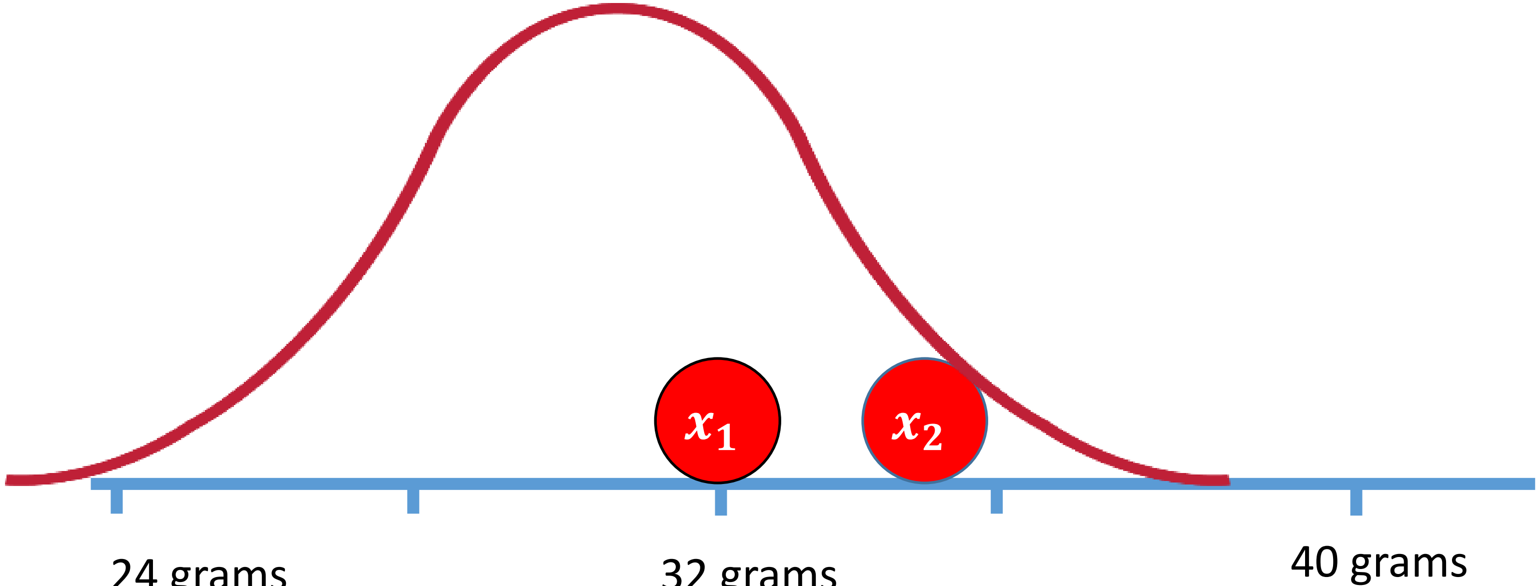
what's the likelihood of this normal curve given both $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$?



Maximum Likelihood for Normal Distribution

$$L(\mu = 28, \sigma = 2 \backslash \boxed{\boldsymbol{x_1} = 32\ and\ \boldsymbol{x_2} = 34)}$$

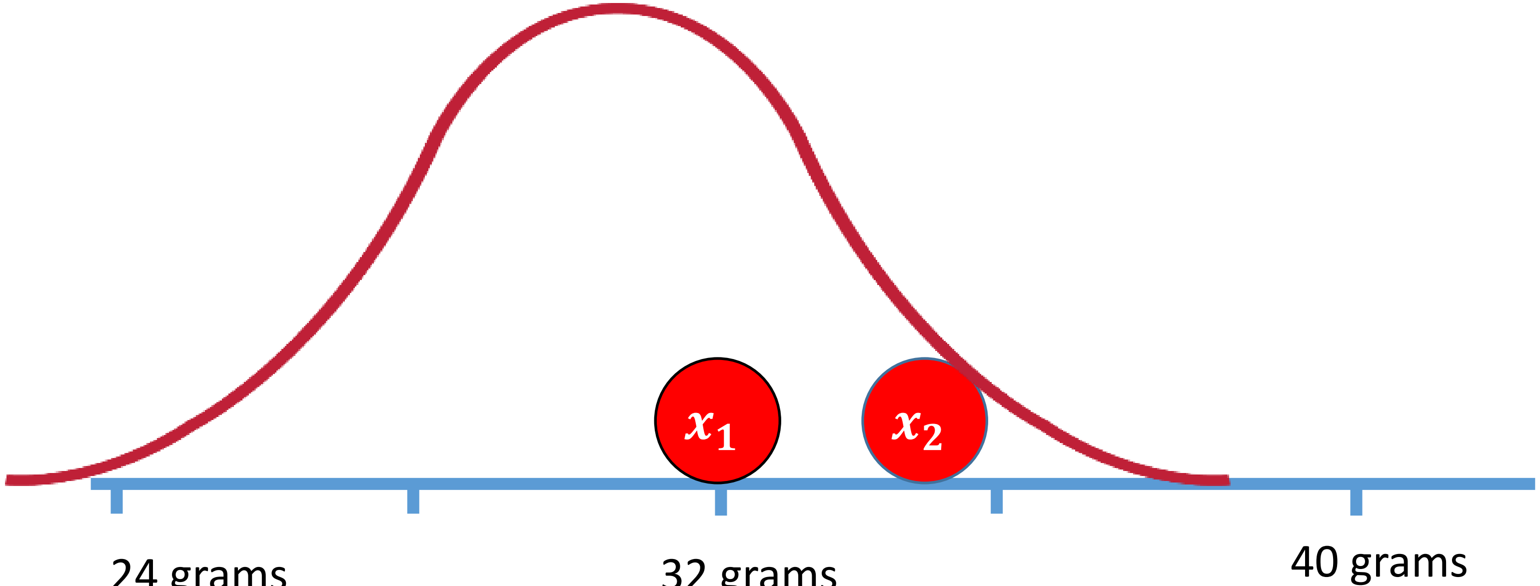
These measurements are independent ($\boldsymbol{x_1}$ did not have an effect on $\boldsymbol{x_2}$)



Maximum Likelihood for Normal Distribution

$$L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32 \; and \boldsymbol{x_2} = 34) = \boxed{L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32\,)} \times \boxed{L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_2} = 34\,)}$$

$$= \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-28)^2/22^2)} \times e^{-(34-28)^2/22^2)}$$

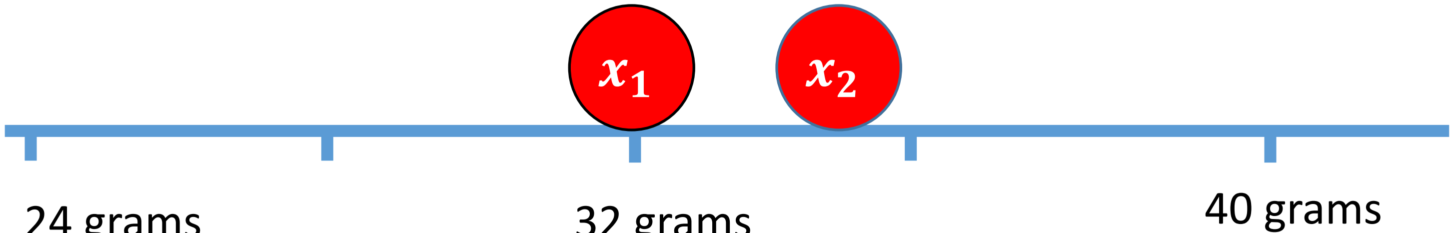


Maximum Likelihood for Normal Distribution

$$L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32 \; and \boldsymbol{x_2} = 34) = \boxed{L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32\,)} \times \boxed{L(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_2} = 34\,)}$$

$$= \frac{1}{\sqrt{2\pi 2^2}} e^{-(32-28)^2/22^2)} \times e^{-(34-28)^2/22^2)}$$

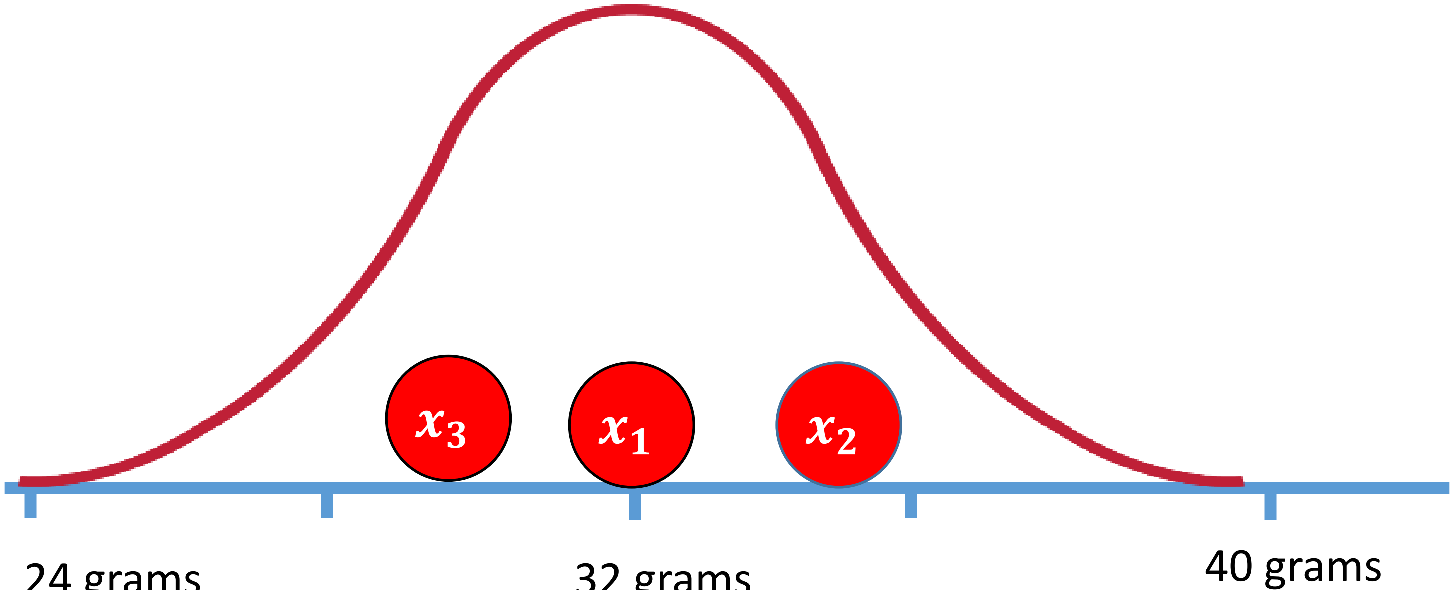
=0.0000006



Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32\ and\, \boldsymbol{x_2} = 34\ and\, \boldsymbol{x_3} = 30) =$$

$$\mathrm{L}(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_1} = 32\ ) \times \mathrm{L}(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_2} = 34\ ) \times \mathrm{L}(\mu = 28, \sigma = 2 \backslash \boldsymbol{x_3} = 30\ )$$

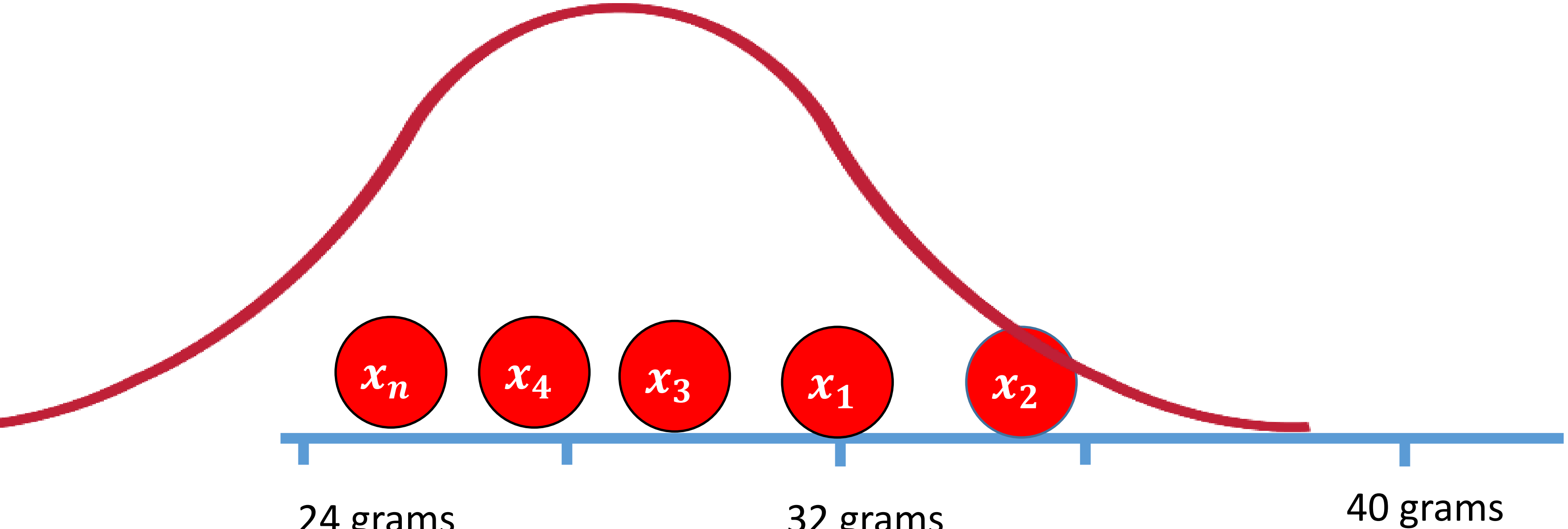


Maximum Likelihood for Normal Distribution

$$L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = L(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times L(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots . \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}$$

we multiply together all individual likelihood functions and
Lets solve the likelihood estimate for $\mu, and\ \sigma$

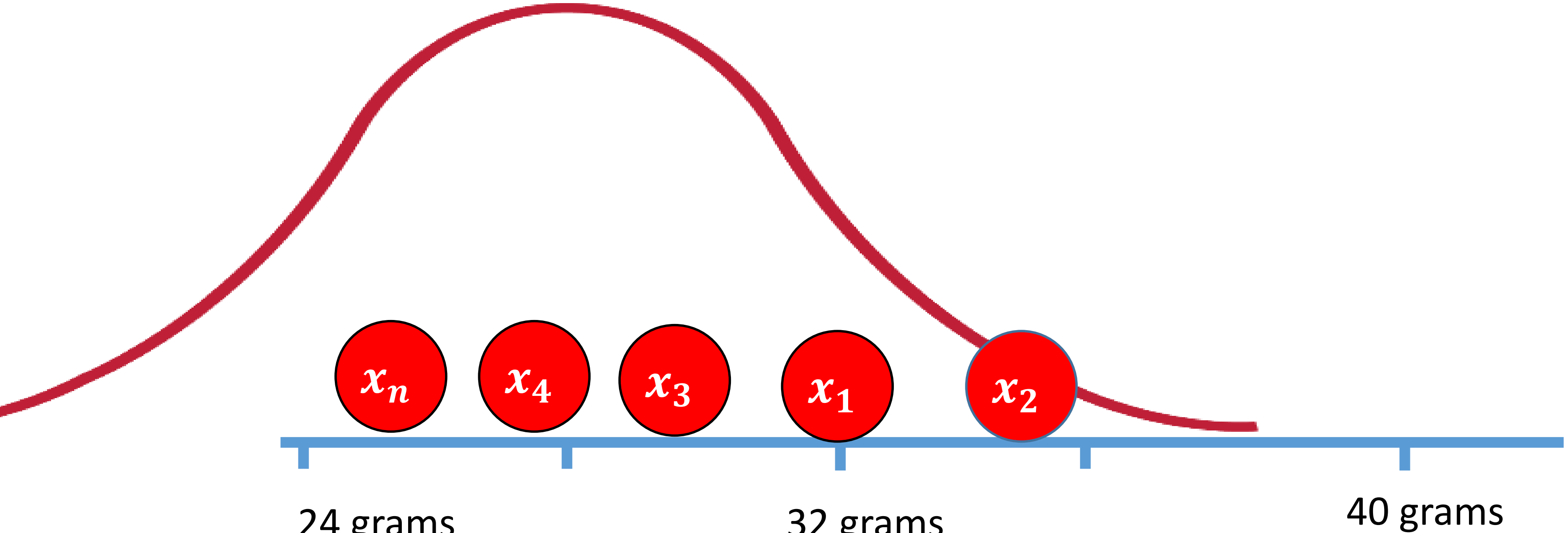


Maximum Likelihood for Normal Distribution

$$L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = L(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times L(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}$$

we multiply together all individual likelihood functions and
Lets solve the likelihood estimate for $\mu, and\ \sigma$



Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}$$

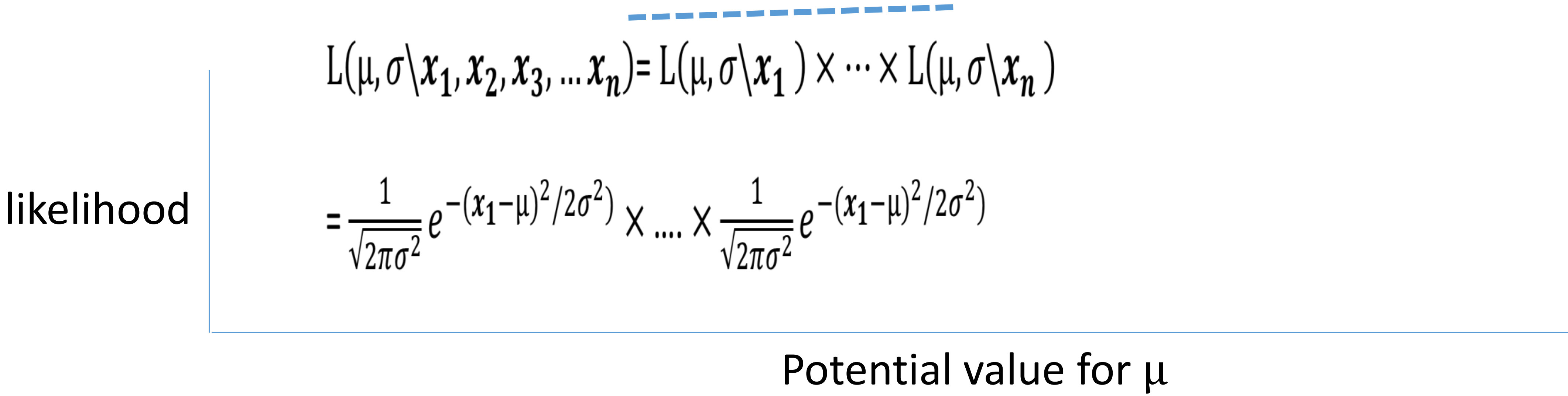
We need to do is take two different derivatives of this equation

Maximum Likelihood for Normal Distribution

$$L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = L(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times L(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1} - \mu)^2 / 2\sigma^2)} \times \ldots . \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1} - \mu)^2 / 2\sigma^2)}$$

We need to one derivative will be with respect μ, when we treat σ like its constant..

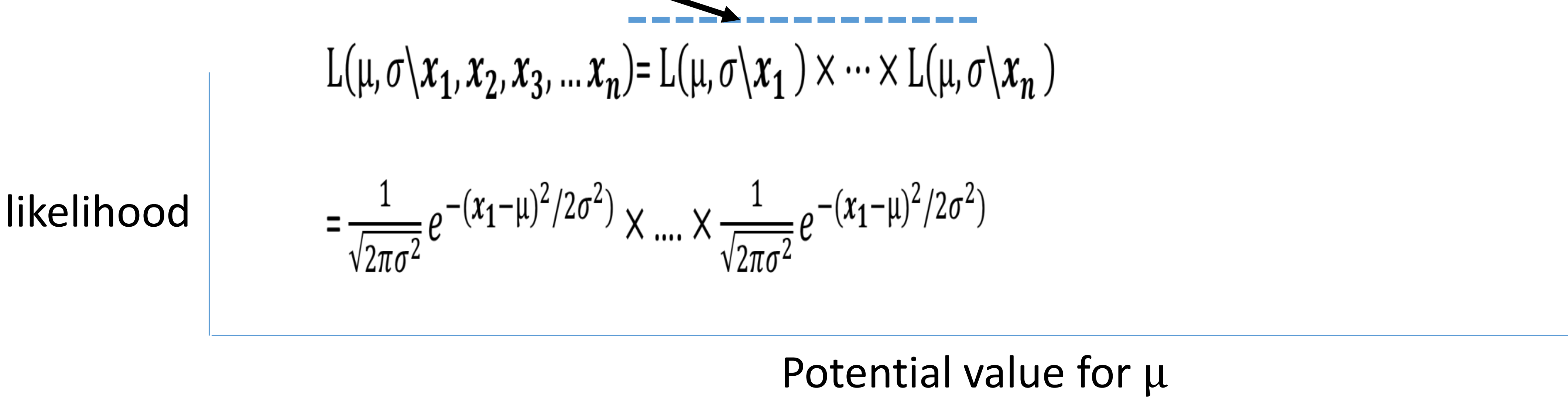


Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1} - \mu)^2 / 2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1} - \mu)^2 / 2\sigma^2)}$$

and we can find the maximum likelihood estimate for μ, by finding where this derivative**=0**

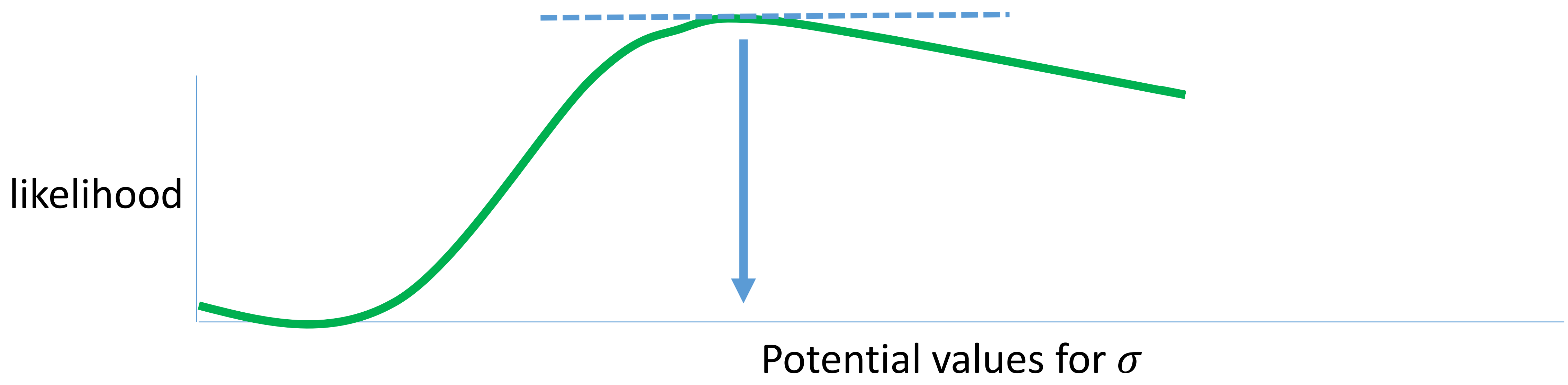


Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}$$

and other derivatives will be with respect σ when we treat μ like its a constant

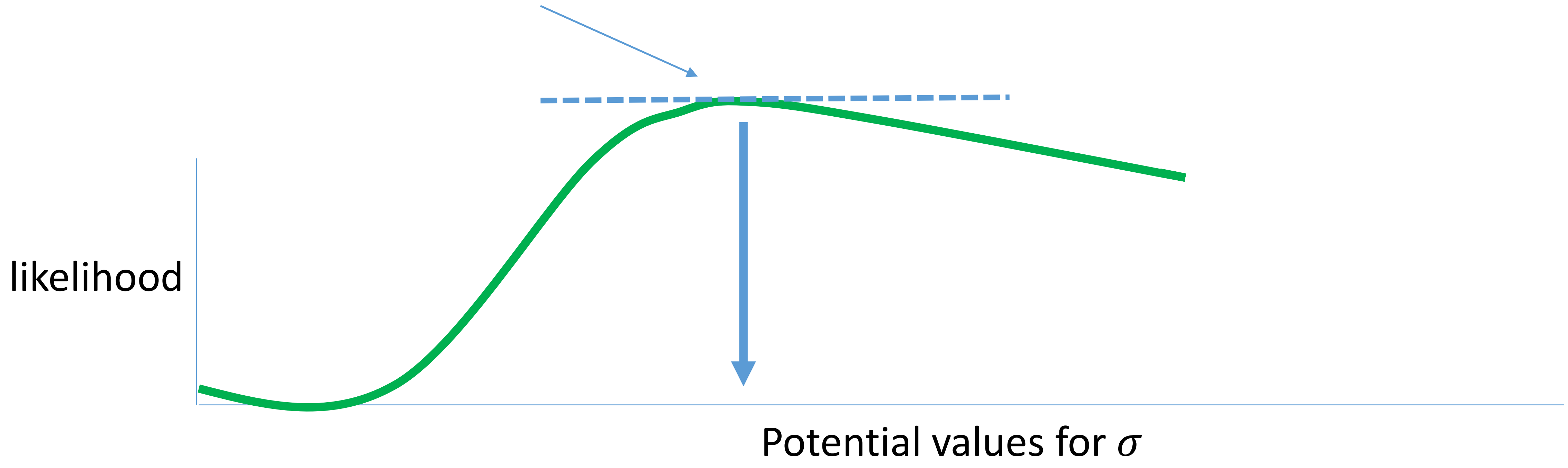


Maximum Likelihood for Normal Distribution

$$\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times \mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}$$

and we can find the maximum likelihood estimate for σ, by finding where this derivative **=0**



Maximum Likelihood for Normal Distribution

$$L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}) = L(\mu, \sigma \backslash \boldsymbol{x_1}) \times \cdots \times L(\mu, \sigma \backslash \boldsymbol{x_n})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2)}$$

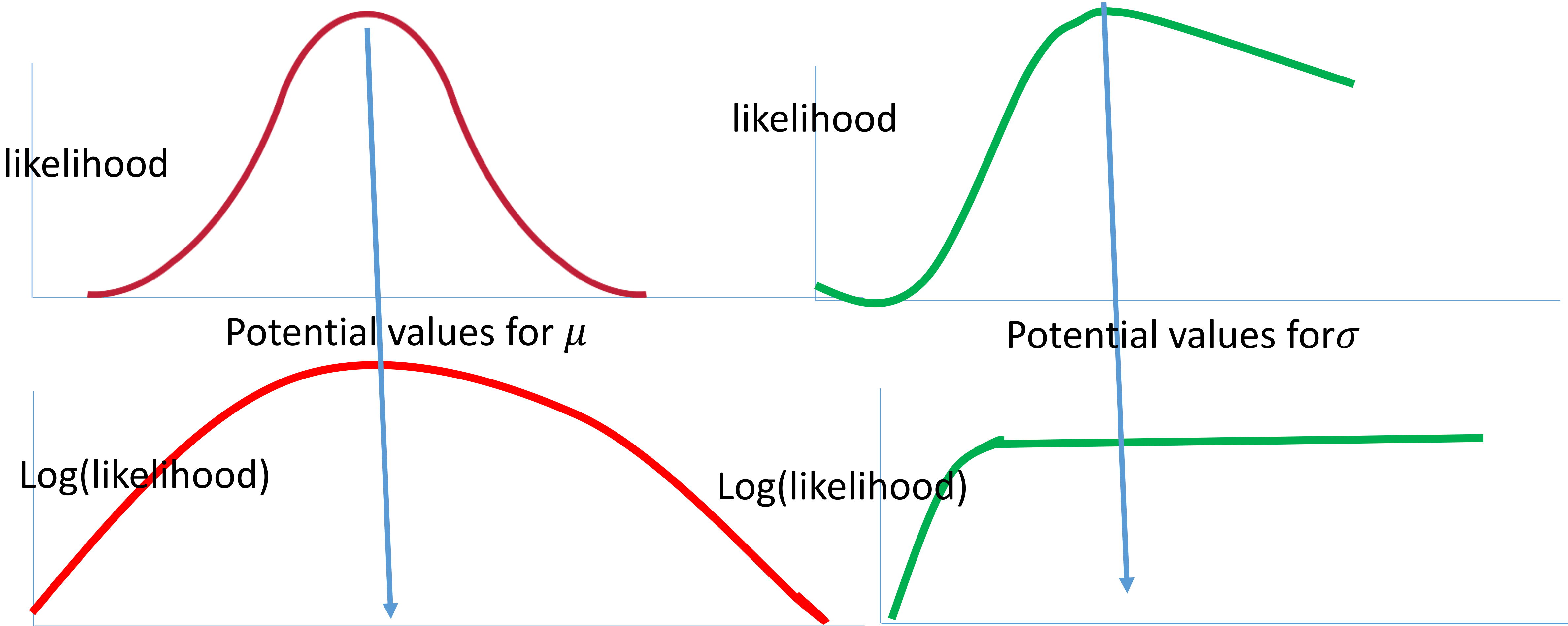
$$\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]$$
$$= \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2)}\right)$$

Before we try to take any derivatives we take the log of the likelihood function

Maximum Likelihood for Normal Distribution

Before we try to take any the likelihood function and log of the likelihood function

Both peak at the same values for μ, σ



Maximum Likelihood for Normal Distribution

$\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]$

$=\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)} \times \ldots. \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2)}\right)$

$= \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}\right) + \cdots + ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2)}\right)$

$=\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \times e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}\right)$

$=\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + ln\left(e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}\right)$

log conversion in first term and all in the following

$=\ln\left[(2\pi\sigma^2)^{-1/2}\right] - (\boldsymbol{x_1} - \mu)^2/2\sigma^2\ln(e)$

$=-1/2\ln[2\pi\sigma^2] - \frac{\boldsymbol{x_1}-\mu)^2}{2\sigma^2}$

$\ln(e) = 1$

$=-1/2\ln(2\pi)\ -1/2\ln(\sigma^2) - \frac{\boldsymbol{x_1}-\mu)^2}{2\sigma^2}$

$=-1/2\ln(2\pi)\ -\ln(\sigma) - \frac{\boldsymbol{x_1}-\mu)^2}{2\sigma^2}$

Maximum Likelihood for Normal Distribution

$$\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})] = \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2)}\right) + \cdots + ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2)}\right)$$

$$\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})] = \boxed{-1/2\ln(2\pi) - \ln(\sigma) - \frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2}}$$

$$-\ldots \boxed{-1/2\ln(2\pi) - \ln(\sigma) - \frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}}$$

we have a term for the first data point, x1…

we have a term for the first data point, $\boldsymbol{x_n}$ …

Maximum Likelihood for Normal Distribution

$$\ln[L(\mu,\sigma \backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots \boldsymbol{x_n})] = \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_1}-\mu)^2/2\sigma^2}\right) + \cdots + ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\boldsymbol{x_n}-\mu)^2/2\sigma^2}\right)$$

$$\ln[L(\mu,\sigma \backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots \boldsymbol{x_n})] = \boxed{-1/2\ln(2\pi)}\ \boxed{-\ln(\sigma)}\ \boxed{-\frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2}}$$

$$\boxed{-\ldots\ -1/2\ln(2\pi)}\ \boxed{-\ln(\sigma)}\ \boxed{-\frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}}$$

$$= -n/2\ln(2\pi)\ -n\ \ln(\sigma) - \frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2} - \cdots \ldots - - \frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}$$

Maximum Likelihood for Normal Distribution

ln$[\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]$=

= -n/2ln(2π) -n ln$(\sigma) - \frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2} - \cdots \ldots - \frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}$

This is the log of the likelihood function after simplification , and it is what we will Take the derivative of

$$\frac{\partial}{\partial \mu}\ln[\mathrm{L}(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]$$

This is the we will starts by taking the derivative with respect to μ

Maximum Likelihood for Normal Distribution

$\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]=$

$= -n/2\ln(2\pi)$ -n $\ln(\sigma)$ $\boxed{-\frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2}}$ $- \cdots \ldots - \frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}$

This is the log of the likelihood function after simplification , and it is what we will Take the derivative of

$\frac{\partial}{\partial\mu}\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})]$=0-0

This is the we will starts by taking the derivative with respect to μ

Maximum Likelihood for Normal Distribution

$-\frac{(x_1-\mu)^2}{2\sigma^2}$=-2(($x_1$- μ)(-1)=2(($x_1$- μ)

ln[L(μ, $\sigma\backslash x_1, x_2, x_3, \ldots x_n$)]=

= -n/2ln(2π) -n ln(σ) $\boxed{-\frac{(x_1-\mu)^2}{2\sigma^2}}$ $-\cdots\ldots-\frac{(x_n-\mu)^2}{2\sigma^2}$

This is the log of the likelihood function after simplification , and it is what we will Take the derivative o

$\frac{\partial}{\partial\mu}$ln[L(μ, $\sigma\backslash x_1, x_2, x_3, \ldots x_n$)]=0-0+ 2($x_1$ - μ)/2 σ^2

$\frac{\partial}{\partial\mu}$ln[L(μ, $\sigma\backslash x_1, x_2, x_3, \ldots x_n$)]=0-0+ (($x_1$ - μ)/σ^2+…+ (x_n - μ)/σ^2

$\frac{\partial}{\partial\mu}$ln[L(μ, $\sigma\backslash x_1, x_2, x_3, \ldots x_n$)]= 1/$\sigma^2$ [((x_1 − μ)+…+ (x_n − μ)]

Maximum Likelihood for Normal Distribution

$$\frac{\partial}{\partial \mu}\ln[L(\mu, \sigma \backslash \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n})] = 1/\sigma^2\ [((\boldsymbol{x_1} - \mu) + \ldots + (\boldsymbol{x_n} - \mu)]$$

$$= 1/\sigma^2\ [((\boldsymbol{x_1} + \ldots + \boldsymbol{x_n}) - n\mu]$$

Maximum Likelihood for Normal Distribution

$$\frac{\partial}{\partial\sigma}\ln[L(\mu,\sigma\backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots\boldsymbol{x_n})]$$

$$-\frac{(\boldsymbol{x_1}-\mu)^2}{2}\sigma^{-2} = -\frac{(\boldsymbol{x_1}-\mu)^2}{2}(-2)\sigma^{-3}$$
$$= (\boldsymbol{x_1}-\mu)^2\sigma^{-3}$$

Now lets take the derivative of the log-likelihood function with respect to σ

Take Derivative of function

$$\frac{\partial}{\partial\sigma}\ln[L(\mu,\sigma\backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots\boldsymbol{x_n})] = -n/2\ln(2\pi) - n\ln(\sigma)$$
$$-\frac{(\boldsymbol{x_1}-\mu)^2}{2\sigma^2} - \cdots\ldots - \frac{(\boldsymbol{x_n}-\mu)^2}{2\sigma^2}$$
$$= 0 - n/\sigma + (\boldsymbol{x_1}-\mu)^2\sigma^{-3} + \ldots + (\boldsymbol{x_n}-\mu)^2\sigma^{-3}$$
$$= -n/\sigma + (\boldsymbol{x_1}-\mu)^2\sigma^{-3} + \ldots + (\boldsymbol{x_n}-\mu)^2\sigma^{-3}$$
$$= -n/\sigma + \frac{1}{\sigma^3}[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

Maximum Likelihood for Normal Distribution

$$\frac{\partial}{\partial\mu}\ln[L(\mu,\sigma\backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots \boldsymbol{x_n})] = 1/\sigma^2\ [((\boldsymbol{x_1}-\mu)+\ldots+(\boldsymbol{x_n}-\mu)]$$

$$\frac{\partial}{\partial\sigma}\ln[L(\mu,\sigma\backslash \boldsymbol{x_1},\boldsymbol{x_2},\boldsymbol{x_3},\ldots \boldsymbol{x_n})] = -n/\sigma + \frac{1}{\sigma^3}[(\boldsymbol{x_1}-\mu)^2+\cdots+(\boldsymbol{x_n}-\mu)^2]$$

Maximum Likelihood for Normal Distribution

Find the peak, take derivative equal to 0

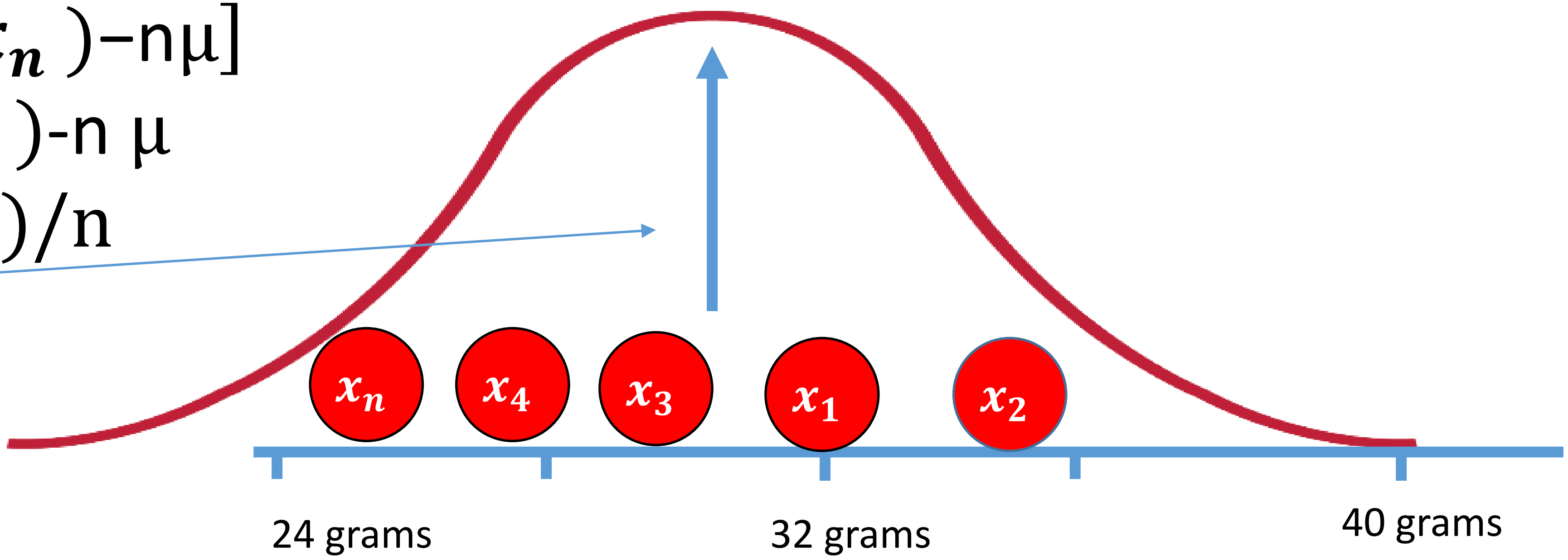
$$0= 1/\sigma^2\ [((\boldsymbol{x_1} +\ldots+ \boldsymbol{x_n})-n\mu]$$

$$0= [((\boldsymbol{x_1} +\ldots+ \boldsymbol{x_n})-n\mu]$$

$$0= (\boldsymbol{x_1} +\ldots+ \boldsymbol{x_n})-n\ \mu$$

$$\mu=(\boldsymbol{x_1} +\ldots+ \boldsymbol{x_n})/n$$

Find the peak, take derivative equal to 0 we can see that Maximum likelihood estimate . For μ is the standard deviation of the measurements



$$0=-n/\ \sigma + \frac{1}{\sigma^3}[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

Maximum Likelihood for Normal Distribution

Find the peak, take derivative equal to 0

$$0=-n/\sigma + \frac{1}{\sigma^3}[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

$$0=-n+\frac{1}{\sigma^2}[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

$$n=\frac{1}{\sigma^2}[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

$$\sigma^2 n=[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]$$

$$\sigma^2 =[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]/n$$

$$\boxed{\sigma = \sqrt{[(\boldsymbol{x_1}-\mu)^2 + \cdots + (\boldsymbol{x_n}-\mu)^2]/n}}$$

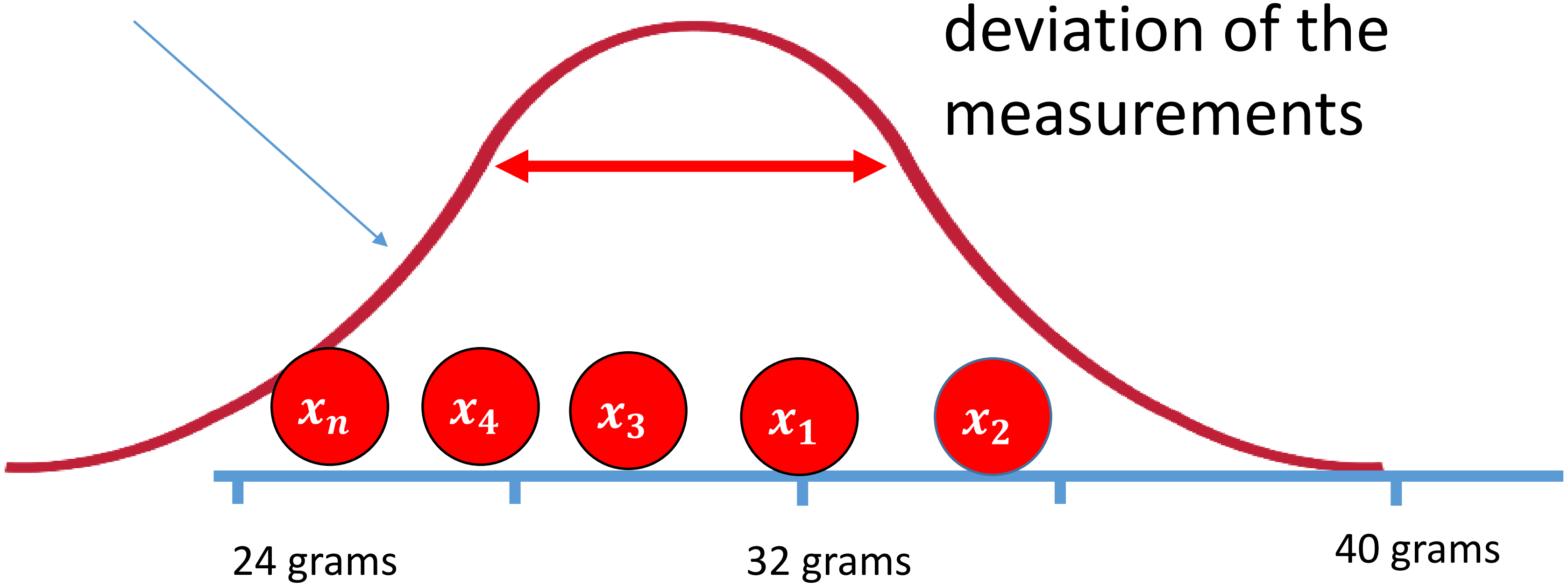
Find the peak, take derivative equal to 0 we can see that Maximum likelihood estimate for σ is the standard deviation of the measurements

Maximum Likelihood for Normal Distribution

Find the peak, take derivative equal to 0

$$\sigma = \sqrt{[(\boldsymbol{x_1} - \mu)^2 + \cdots + (\boldsymbol{x_n} - \mu)^2]/\, n}$$

we can see that Maximum likelihood estimate for σ is the standard deviation of the measurements




Q&A