# Topics to be covered

- Classifiers: (Bayesian, Maximum a posteriori, parameter estimation, decision tree, SVM, bag of words, N-gram models, association rules, nearest neighbor, locally weighted regression)
- Clustering (mixture models, k-means clustering, hierarchical clustering, distributional clustering )
- Ensemble techniques

# 1. Classifiers



## Things you can do.

### Classification

| Age | Likes Pinapple on Pizza |
|---|---|
| 42 | 1 |
| 65 | 1 |
| 50 | 1 |
| 76 | 1 |
| 96 | 1 |
| 50 | 1 |
| 91 | 0 |
| 58 | 1 |
| 25 | 1 |
| 23 | 1 |
| 75 | 1 |
| 46 | 0 |
| 87 | 0 |
| 96 | 0 |
| 45 | 0 |
| 32 | 1 |
| 63 | 0 |
| 21 | 1 |
| 26 | 1 |
| 93 | 0 |
| 68 | 1 |
| 96 | 0 |

### Regression

| Height(Inches) | Weight(Pounds) |
|---|---|
| 65.78 | 112.99 |
| 71.52 | 136.49 |
| 69.40 | 153.03 |
| 68.22 | 142.34 |
| 67.79 | 144.30 |
| 68.70 | 123.30 |
| 69.80 | 141.49 |
| 70.01 | 136.46 |
| 67.90 | 112.37 |
| 66.78 | 120.67 |
| 66.49 | 127.45 |
| 67.62 | 114.14 |
| 68.30 | 125.61 |
| 67.12 | 122.46 |
| 68.28 | 116.09 |

# 1. Classifiers



## Bag of Words (BoW)

- Bag of Words is a feature extraction method ( Way of converting raw data into feature vector )
- A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:
    - A vocabulary of known words.
    - A measure of the presence of known words.
- Information about the order and structure lost
- Each word count is a feature.

# 1. Classifiers

## Bag of Words (BoW)

**Rules we follow when using bag of words.**

Ignoring case ( Applicable to English )
Ignoring punctuation (Assuming they dont carry vital information)
Ignoring frequent words that don't contain much information, called stop words, like "a," "of," etc. (Also known as, stopwords )
Fixing misspelled words. ( Mapping disturbing samples back to 'authentic' vocab)
Reducing words to their stem (e.g. "play" from "playing") using stemming algorithms ( What are your thoughts about stemming in Bangla? )

# 1. Classifiers

## Bag of Words (BoW)

- Sample Text
  - Line 1: মোরা ঝঞ্ঝার মত উদ্দ্যম
  - Line 2: মোরা ঝর্ণার মত চঞ্চল,
  - Line 3: মোরা বিধাতার মত নির্ভয়
  - Line 4: মোরা প্রকৃতির মত স্বচ্ছল।।
- Vocabulary: The unique words are(skipping space and punctuation) :
  'মোরা' 'ঝঞ্ঝার' 'মত' 'উদ্দ্যম' 'ঝর্ণার' 'চঞ্চল' 'বিধাতার' 'নির্ভয়' 'প্রকৃতির' 'স্বচ্ছল'
- The vocabulary contains 10 unique words.
- Using the unique vocab as reference, the first line would be
  - [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
- And the second line will be
  - [1, 0, 1, 0, 1, 1, 0, 0, 0, 0]

This way, we have converted our texts into feature vectors.

# 1. Classifiers

## Bag of Words (BoW)

Different Scoring Methods for Bag of Words

- **Vocabulary:** The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.
- **Sparsity**: Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space.
- **Meaning**: Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modeled could tell the difference between the same words differently arranged ("this is interesting" vs "is this interesting"), synonyms ("old bike" vs "used bike"), and much more.

# 1. Classifiers

## N-Grams (Vocabulary of grouped words)

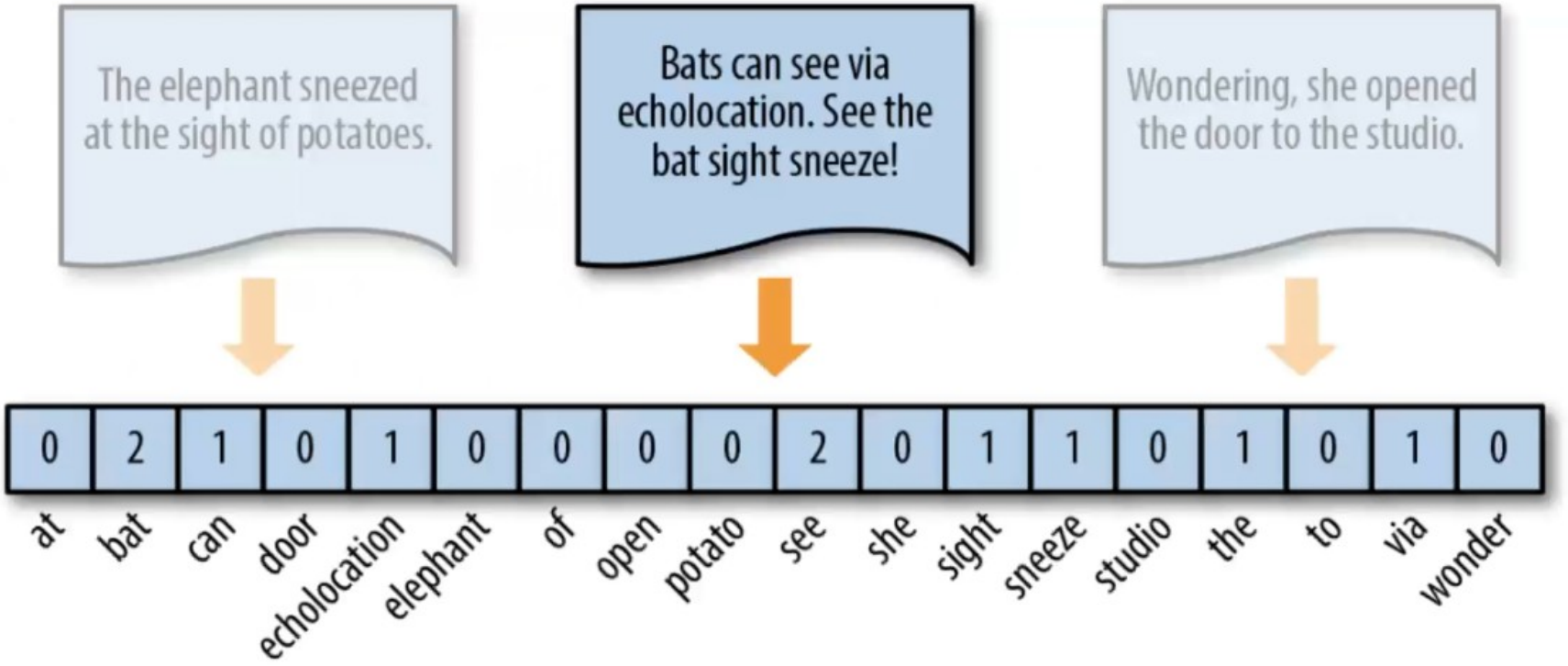
A bag-of-N-Grams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

**Figure 1 *n*-gram examples from various disciplines**

| Field | Unit | Sample sequence | 1-gram sequence | 2-gram sequence | 3-gram sequence |
|---|---|---|---|---|---|
| Vernacular name | | | unigram | bigram | trigram |
| Order of resulting Markov model | | | 0 | 1 | 2 |
| Protein sequencing | amino acid | … Cys-Gly-Leu-Ser-Trp … | …, Cys, Gly, Leu, Ser, Trp, … | …, Cys-Gly, Gly-Leu, Leu-Ser, Ser-Trp, … | …, Cys-Gly-Leu, Gly-Leu-Ser, Leu-Ser-Trp, … |
| DNA sequencing | base pair | …AGCTTCGA… | …, A, G, C, T, T, C, G, A, … | …, AG, GC, CT, TT, TC, CG, GA, … | …, AGC, GCT, CTT, TTC, TCG, CGA, … |
| Computational linguistics | character | … to_be_or_not_to_be… | …, t, o, _, b, e, _, o, r, _, n, o, t, _, t, o, _, b, e, … | …, to, o_, _b, be, e_, _o, or, r_, _n, no, ot, t_, _t, to, o_, _b, be, … | …, to_, o_b, _be, be_, e_o, _or, or_, r_n, _no, not, ot_, t_t, _to, to_, o_b, _be, … |
| Computational linguistics | word | … to be or not to be … | …, to, be, or, not, to, be, … | …, to be, be or, or not, not to, to be, … | …, to be or, be or not, or not to, not to be, … |

# 1. Classifiers

## Bag of Words (BoW)



# 1. Classifiers

## N-Grams (Vocabulary of grouped words)

A bag-of-N-Grams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

**Figure 1 *n*-gram examples from various disciplines**

| Field | Unit | Sample sequence | 1-gram sequence | 2-gram sequence | 3-gram sequence |
|---|---|---|---|---|---|
| Vernacular name | | | unigram | bigram | trigram |
| Order of resulting Markov model | | | 0 | 1 | 2 |
| Protein sequencing | amino acid | … Cys-Gly-Leu-Ser-Trp … | …, Cys, Gly, Leu, Ser, Trp, … | …, Cys-Gly, Gly-Leu, Leu-Ser, Ser-Trp, … | …, Cys-Gly-Leu, Gly-Leu-Ser, Leu-Ser-Trp, … |
| DNA sequencing | base pair | …AGCTTCGA… | …, A, G, C, T, T, C, G, A, … | …, AG, GC, CT, TT, TC, CG, GA, … | …, AGC, GCT, CTT, TTC, TCG, CGA, … |
| Computational linguistics | character | … to_be_or_not_to_be… | …, t, o, _, b, e, _, o, r, _, n, o, t, _, t, o, _, b, e, … | …, to, o_, _b, be, e_, _o, or, r_, _n, no, ot, t_, _t, to, o_, _b, be, … | …, to_, o_b, _be, be_, e_o, _or, or_, r_n, _no, not, ot_, t_t, _to, to_, o_b, _be, … |
| Computational linguistics | word | … to be or not to be … | …, to, be, or, not, to, be, … | …, to be, be or, or not, not to, to be, … | …, to be or, be or not, or not to, not to be, … |

# 1. Classifiers

## N-Grams (Vocabulary of grouped words)

**Skip Grams:** A generalization of n-grams in which the components (typically words) need not be consecutive in the text under consideration, but may leave gaps that are skipped over.

Comparative examples of Bi-Gram, Tri-Gram, and Skip Gram

মোরা ঝঞ্ঝার মত উদ্দ্যম মোরা ঝর্ণার মত চঞ্চল, মোরা বিধাতার মত নির্ভয় মোরা প্রকৃতির মত স্বচ্ছল।।

Bi-Grams: (মোরা ঝঞ্ঝার), (ঝঞ্ঝার মত), (মত উদ্দ্যম), (উদ্দ্যম, মোরা), (মোরা ঝর্ণার), (ঝর্ণার মত) (মত চঞ্চল), ... ...

Tri-Grams: (মোরা ঝঞ্ঝার মত), (ঝঞ্ঝার মত উদ্দ্যম), (মত উদ্দ্যম মোরা), (উদ্দ্যম মোরা ঝর্ণার) .... ...

Skip-Grams: There can be many types of skip grams. Two numbers define them. The amount of 'skipping' performed. And the N-gram length we are using.

1-skip-2-grams: (মোরা মত), (ঝঞ্ঝার উদ্দ্যম), (মত মোরা), (উদ্দ্যম, ঝর্ণার) ... ...

# 1. Classifiers

## N-Grams (Vocabulary of grouped words)

**Skip Grams:** A generalization of n-grams in which the components (typically words) need not be consecutive in the text under consideration, but may leave gaps that are skipped over.

Comparative examples of Bi-Gram, Tri-Gram, and Skip Gram

মোরা ঝঞ্ঝার মত উদ্দ্যম মোরা ঝর্ণার মত চঞ্চল, মোরা বিধাতার মত নির্ভয় মোরা প্রকৃতির মত স্বচ্ছল।।

Bi-Grams: (মোরা ঝঞ্ঝার), (ঝঞ্ঝার মত), (মত উদ্দ্যম), (উদ্দ্যম, মোরা), (মোরা ঝর্ণার), (ঝর্ণার মত) (মত চঞ্চল), ... ...

Tri-Grams: (মোরা ঝঞ্ঝার মত), (ঝঞ্ঝার মত উদ্দ্যম), (মত উদ্দ্যম মোরা), (উদ্দ্যম মোরা ঝর্ণার) .... ...

Skip-Grams: There can be many types of skip grams. Two numbers define them. The amount of 'skipping' performed. And the N-gram length we are using.

1-skip-2-grams: (মোরা মত), (ঝঞ্ঝার উদ্দ্যম), (মত মোরা), (উদ্দ্যম, ঝর্ণার) ... ...

# 1. Classifiers

## N-Grams (Vocabulary of grouped words)

**Syntactic n-grams:** Syntactic n-grams are n-grams defined by paths in syntactic dependency or constituent trees rather than the linear structure of the text.

For example, the sentence "economic news has little effect on financial markets" can be transformed to syntactic n-grams following the tree structure of its dependency relations: news-economic, effect-little, effect-on-markets-financial.

*Why do you think these work so good?

# 1. Classifiers



## Association Rules (Definition)

A bag-of-N-Grams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

Following the original definition by Agrawal, Imieliński, Swami[2] the problem of association rule mining is defined as:
Let $I=\{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items.

Let $D=\{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database.

Each transaction in D has a unique transaction ID and contains a subset of the items in I.

A rule is defined as an implication of the form:

$X \Rightarrow Y$, where $X,Y \subseteq I$

For details:
https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

# 1. Classifiers

## Association Rules (Definition)

A bag-of-N-Grams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

Following the original definition by Agrawal, Imieliński, Swami[2] the problem of association rule mining is defined as:
Let $I=\{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items.

Let $D=\{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database.

Each transaction in D has a unique transaction ID and contains a subset of the items in I.

A rule is defined as an implication of the form:

$X \Rightarrow Y$, where $X,Y \subseteq I$

For details:
https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

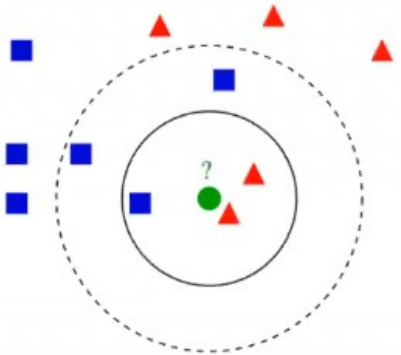
# 1. Classifiers

## KNN ( K-Nearest Neighbour)

Definition: the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

**In k-NN classification**, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

**In k-NN regression**, the output is the property value for the object. This value is the average of the values of k nearest neighbors.



# 1. Classifiers

## KNN ( K-Nearest Neighbour)

**Advantages**

- ❑ The algorithm is simple and easy to implement.
- ❑ There's no need to build a model, tune several parameters, or make additional assumptions.
- ❑ The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

**Disadvantages**

- ❑ The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

# 1. Classifiers

## KNN ( K-Nearest Neighbour)

**Depends on the quality of input feature vectors**

**CNN model reduction for k-NN classifiers**

Fig. 1. The dataset.

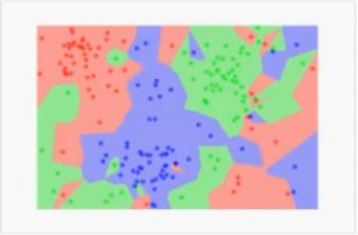


Fig. 2. The 1NN classification map.

Fig. 3. The 5NN classification map.

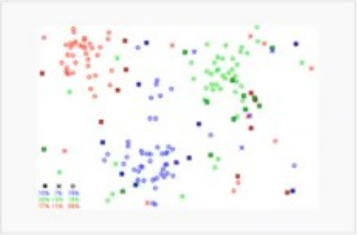


Fig. 4. The CNN reduced dataset.

Fig. 5. The 1NN classification map based on the CNN extracted prototypes.

# 2. Clustering

## What it is and the varieties

Task of dividing the population or data points into a number of groups such that similar data stay together.
Segregate groups with similar traits and assign them into clusters.

**Types of Clustering**

- **Hard Clustering:** Provides binary decision whether it belongs to a certain cluster.
- **Soft Clustering:** Provides soft decision aka a likelihood value for its belonging to a cluster.

Types of clustering algorithms

- Connectivity-based clustering (hierarchical clustering)
- Centroid-based clustering
- Distribution-based clustering
- Density-based clustering
- Grid-based clustering

# 2. Clustering

## K-Means Clustering

**The algorithm** works as follows:

- First we initialize k points, called means, randomly.
- We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.
- We repeat the process for a given number of iterations and at the end, we have our clusters.

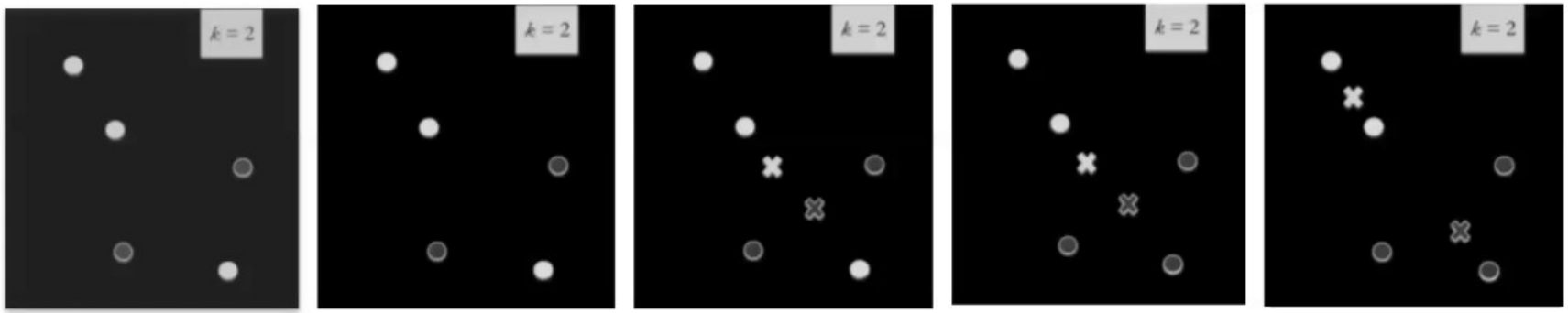
# 2. Clustering



## K Means Clustering

K is equal to 1, 2, 3, … whatever. It's your choice (well, not entirely tho)

Step 1 Step 2 Step 3 Step 4 Step 5



# 1. Clustering

## Mixture Models

At first, we need to know what is a distribution and what is a normal or a Gaussian one.

- A probability distribution is a statistical function that describes all the possible values and likelihoods that a random variable can take within a given range.

### Gaussian distribution

- Gaussian or normal distribution, 1D

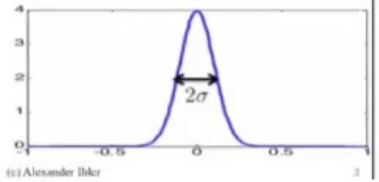
$$\mathcal{N}(x\,;\,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}(x-\mu)^2/\sigma^2\right]$$

- Parameters: mean $\mu$, variance $\sigma^2$ (standard deviation $\sigma$)

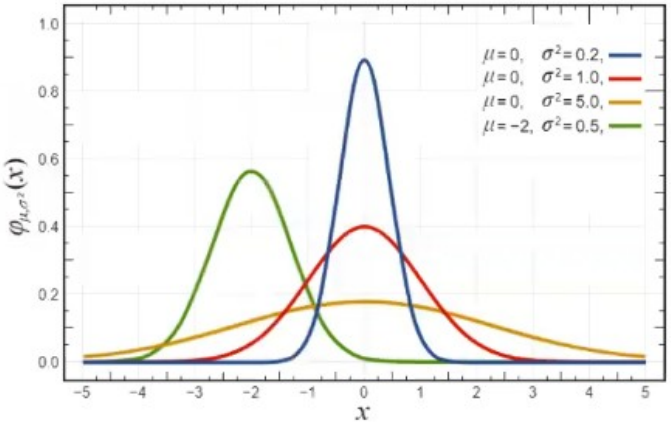
**Maximum Likelihood estimates**

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_i (x^{(i)} - \hat{\mu})^2$$




(c) Alexander Ihler




Reading Materials: http://www.cs.toronto.edu/~rgrosse/csc321/mixture_models.pdf

# 1. Clustering

## Mixture Models

- At first, we need to know what is a distribution and what is a normal or a Gaussian one.
- A probability distribution is a statistical function that describes all the possible values and likelihoods that a random variable can take within a given range.

### Gaussian distribution

- Gaussian or normal distribution, 1D

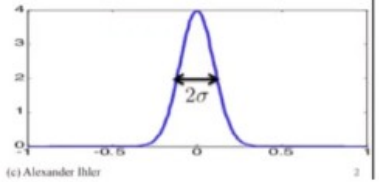
$$\mathcal{N}(x\ ;\ \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}(x-\mu)^2/\sigma^2\right]$$

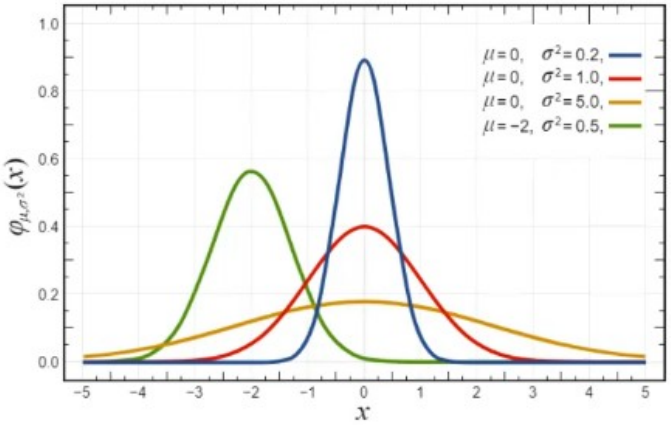
  - Parameters: mean $\mu$, variance $\sigma^2$ (standard deviation $\sigma$)

**Maximum Likelihood estimates**

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_i (x^{(i)} - \hat{\mu})^2$$



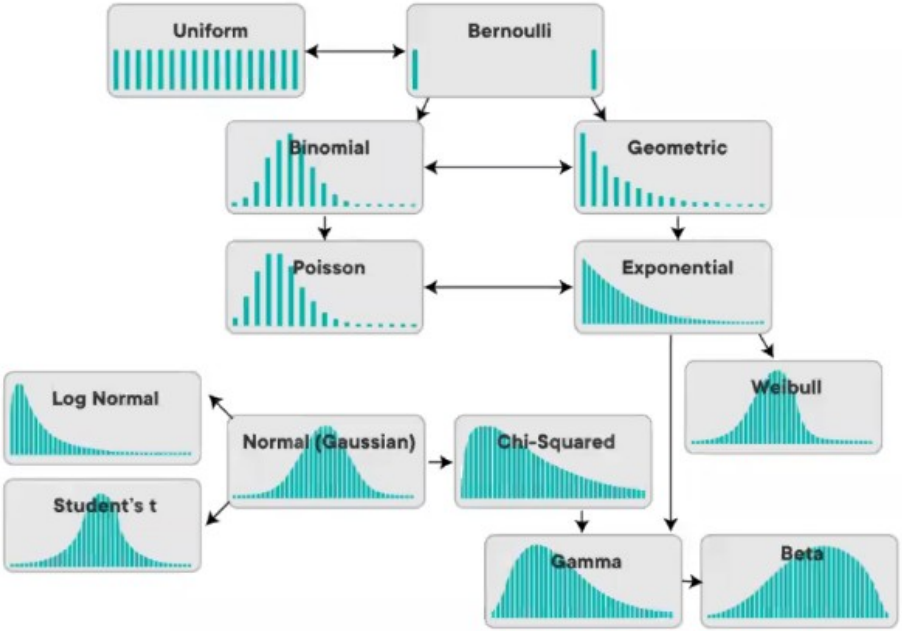

Reading Materials: http://www.cs.toronto.edu/~rgrosse/csc321/mixture_models.pdf

# 2. Clustering



## Mixture Models

Many Types of Probability Distributions are there, and they are related to each other.



# 2. Clustering

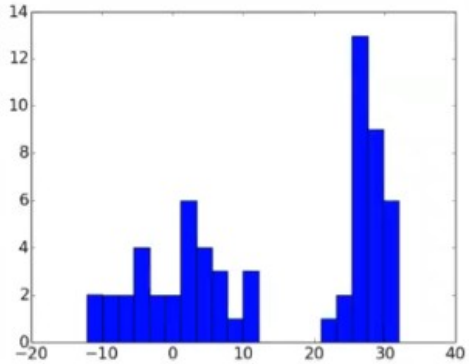


## Mixture Models

In statistics, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs.

In simpler terms, if a distribution is a mixture of more than one distributions, it's a called mixture. (duh) And trying to segregate the population into two or more sub-populations is called mixture modelling.

If all the components of them are Gaussians, its a GMM.
GMM = Gaussian Mixture Model




https://en.wikipedia.org/wiki/File:Movie.gif

# 2. Clustering

## Gaussian Mixture Models

Suppose there are K clusters (For the sake of simplicity here it is assumed that the number of clusters is known and it is K). So mean (mu) and Sigma is also estimated for each k.

- Initialize the mean $mu_k$, the covariance matrix $Sigma_k$ and the mixing coefficients $pi_k$ by some
- random values. (or other values)
- Compute the $gamma_k$ values for all k.
- Again Estimate all the parameters using the current $gamma_k$ values.
- Compute log-likelihood function.
- Put some convergence criterion (When to stop)
- If the log-likelihood value converges to some value ( or if all the parameters converge to some values ) then stop, else return to Step 2
  Read the maths from here: https://www.geeksforgeeks.org/gaussian-mixture-model/

# 2. Clustering

## Hierarchial Clustering

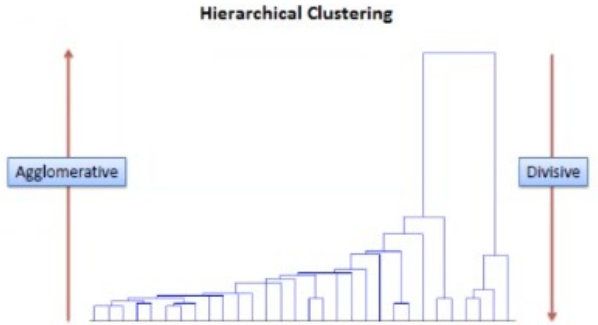
Hierarchical Clustering Algorithm also called Hierarchical cluster analysis or HCA is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom.

This clustering technique is divided into two types:

- ❏ Agglomerative Hierarchical Clustering
- ❏ Divisive Hierarchical Clustering

Divisive algorithms produce more accurate hierarchies than agglomerative algorithms in some circumstances.
What is a Dendrogram?

**A Dendrogram** is a type of tree diagram showing hierarchical relationships between different sets of data.

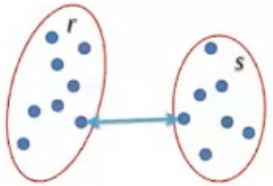


# 2. Clustering

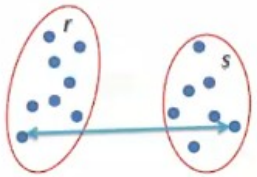


## Hierarchial Clustering

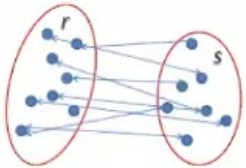
Single Linkage
Complete Linkage
Average Linkage




$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$




$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$




$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

In agglomerative or bottom-up clustering method we assign each observation to its own cluster. Then, compute the similarity (e.g., distance) between each of the clusters and join the two most similar clusters. Finally, repeat steps 2 and 3 until there is only a single cluster left.

# 2. Clustering



## Distributional Clustering

A brand new idea. Link to 2019 paper: https://arxiv.org/abs/1911.05940
Ensures cluster centers capture the distribution of the underlying data

Lets read the paper for details!

# 2. Clustering

## Distributional Clustering

A brand new idea. Link to 2019 paper: https://arxiv.org/abs/1911.05940
Ensures cluster centers capture the distribution of the underlying data

Lets read the paper for details!

# 2. Clustering



## Hierarchial Clustering

- Agglomerative Hierarchical Clustering

The Agglomerative Hierarchical Clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also known as AGNES (Agglomerative Nesting). It's a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

How does it work?

Make each data point a single-point cluster → forms N clusters
Take the two closest data points and make them one cluster → forms N-1 clusters
Take the two closest clusters and make them one cluster → Forms N-2 clusters.
Repeat step-3 until you are left with only one cluster.

https://miro.medium.com/max/257/0*iozEcRXXWXbDMrdG.gif

# 2. Clustering

## Distributional Clustering

A brand new idea. Link to 2019 paper: https://arxiv.org/abs/1911.05940
Ensures cluster centers capture the distribution of the underlying data

Lets read the paper for details!

# 3. Ensemble



## What is an Ensemble?

A collective of different models, all of which work together.
Improves prediction accuracy
Not intuitive.

Ensembles solve three problems

**Statistical Problem** The Statistical Problem arises when the hypothesis space is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!

**Computational Problem** The Computational Problem arises when the learning algorithm cannot guarantees finding the best hypothesis.

**Representational Problem** The Representational Problem arises when the hypothesis space does not contain any good approximation of the target class(es).

# 3. Ensemble



## What is an Ensemble?

Bagging algorithms:

- Random Forest

Boosting algorithms:

- AdaBoost
- Gradient Boosting Machine (GBM)
- XGBoost
- Light GBM

# 3. Ensemble



## Bagging

The name Bootstrap Aggregating, also known as “Bagging”, summarizes the key elements of this strategy. In the bagging algorithm, the first step involves creating multiple models. These models are generated using the same algorithm with random sub-samples of the dataset which are drawn from the original dataset randomly with bootstrap sampling method. In bootstrap sampling, some original examples appear more than once and some original examples are not present in the sample. If you want to create a sub-dataset with m elements, you should select a random element from the original dataset m times. And if the goal is generating n dataset, you follow this step n times.

# 3. Ensemble



## Boosting (Converting Weak Models to Strong Ones)

The term “boosting” is used to describe a family of algorithms which are able to convert weak models to strong models. The model is weak if it has a substantial error rate, but the performance is not random (resulting in an error rate of 0.5 for binary classification). Boosting incrementally builds an ensemble by training each model with the same dataset but where the weights of instances are adjusted according to the error of the last prediction. The main idea is forcing the models to focus on the instances which are hard. Unlike bagging, boosting is a sequential method, and so you can not use parallel operations here.

# 3. Ensemble



## Stacking

Stacking, also known as stacked generalization, is an ensemble method where the models are combined using another machine learning algorithm. The basic idea is to train machine learning algorithms with training dataset and then generate a new dataset with these models. Then this new dataset is used as input for the combiner machine learning algorithm.

# 3. Ensemble



## Majority Voting

Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. If none of the predictions get more than half of the votes, we may say that the ensemble method could not make a stable prediction for this instance. Although this is a widely used technique, you may try the most voted prediction (even if that is less than half of the votes) as the final prediction. In some articles, you may see this method being called “plurality voting”..

# 3. Ensemble



## Weighted Voting

Unlike majority voting, where each model has the same rights, we can increase the importance of one or more models. In weighted voting you count the prediction of the better models multiple times. Finding a reasonable set of weights is up to you.

# 3. Ensemble

## Averaging

In simple averaging method, for every instance of test dataset, the average predictions are calculated. This method often reduces overfit and creates a smoother regression model.

Weighted averaging is a slightly modified version of simple averaging, where the prediction of each model is multiplied by the weight and then their average is calculated.

# 3. Ensemble



## Random Forest

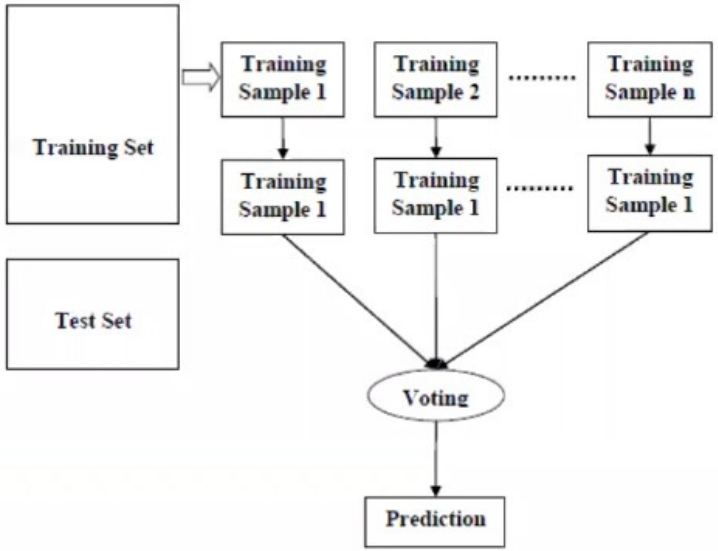
We can understand the working of Random Forest algorithm with the help of following steps −

Step 1 − First, start with the selection of random samples from a given dataset.

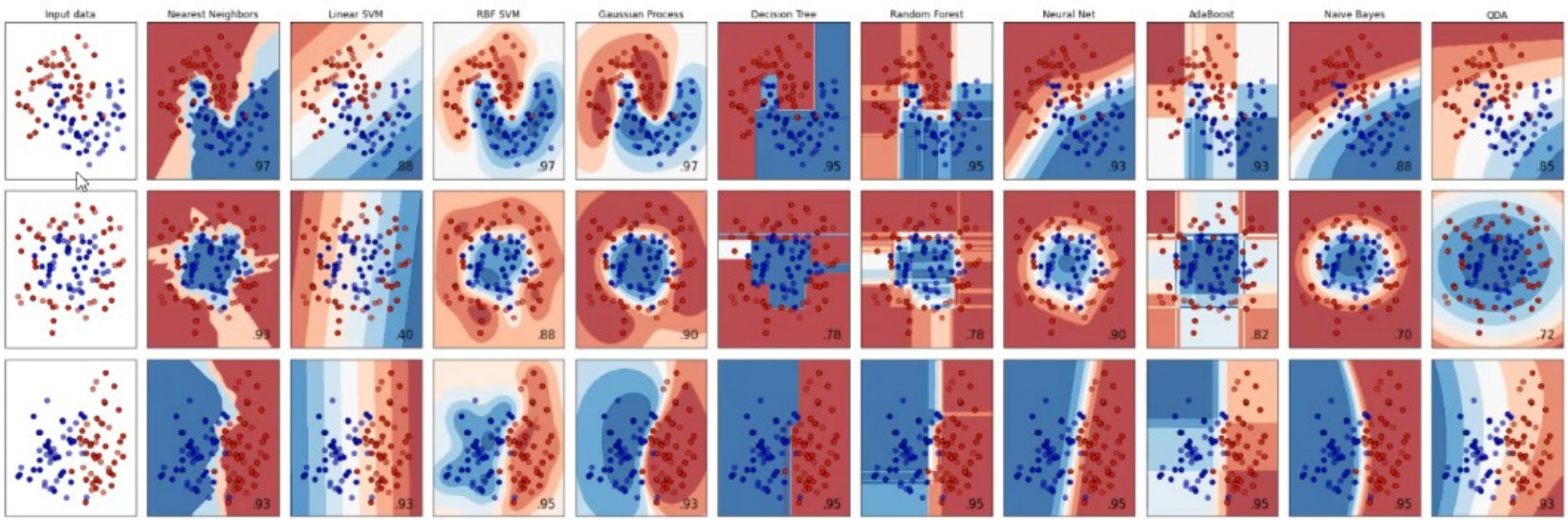
Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.



```
plt.tight_layout()
plt.show()
```




In [ ]: