Rating Aggregation

- Crowdsourcing: Applications and Techniques
  - Crowdsourcing: Findmax
- Mathematical foundations of Rating Aggregation
- Bayesian Ranking: Methodology and Real-world applications
- Ensemble Learning
- Voting Theory
- Ranking Aggregation

Imdad ullah Khan

Human Based Computation - Crowdsourcing

Crowdsourcing

Human Based Computation **Human based Data Gathering**

> Collective intelligence can be brought to bear on a wide variety of problems, and complexity is no bar . . . conditions that are necessary for the crowd to be wise: diversity, independence, and . . . decentralization.
>
> James Surowiecki– The Wisdom of Crowds

Crowdsourcing: The practice of obtaining input, ideas, or services from a large, diverse group of people, typically via the internet

- Wikipedia - Content Creation
- Amazon Mechanical Turk – Task Solving
- Amazon, IMDB, . . . Rating Aggregation

Harnesses collective intelligence for better decision-making

Offers scalability, diversity of inputs, and cost efficiency

Crowdsourcing Applications: reCAPTCHA

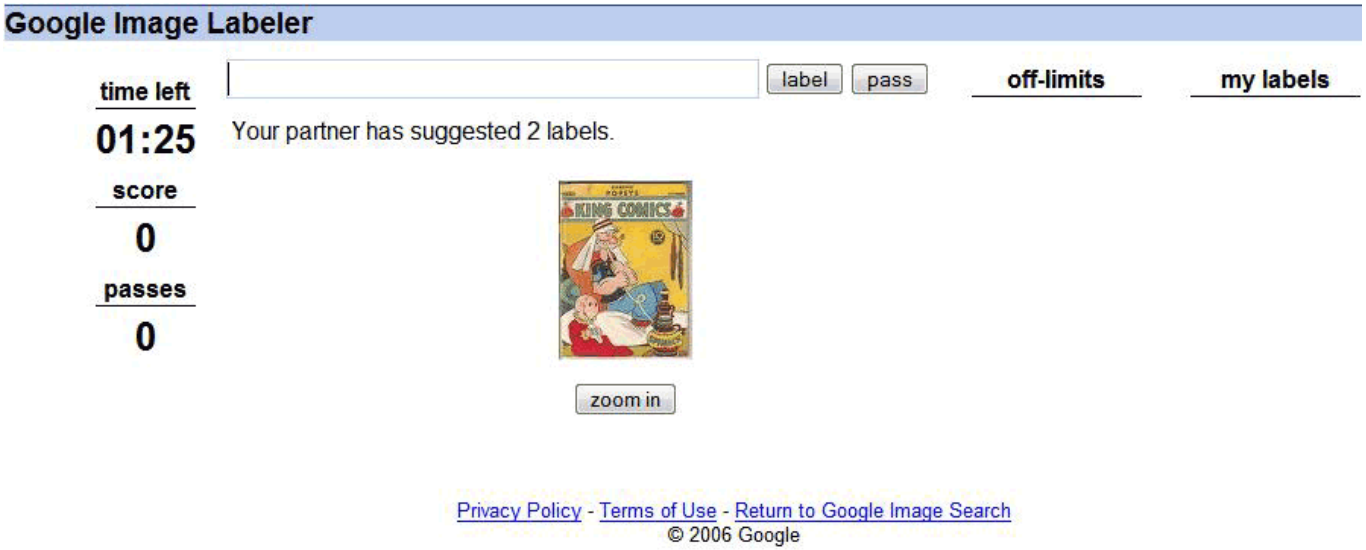
reCAPTCHA Inc. (now owned by Google) enables web host to distinguish between spambots and human website access

- Asked users to decipher hard-to-read text or match images
- Used to crowd source digitization of books illegible for scanning

Crowdsourcing Applications: ESP and GWAP

ESP game (extrasensory perception game) rebranded as GWAP (game with a purpose)

Google Image Labeler (based on ESP) allows users to label images



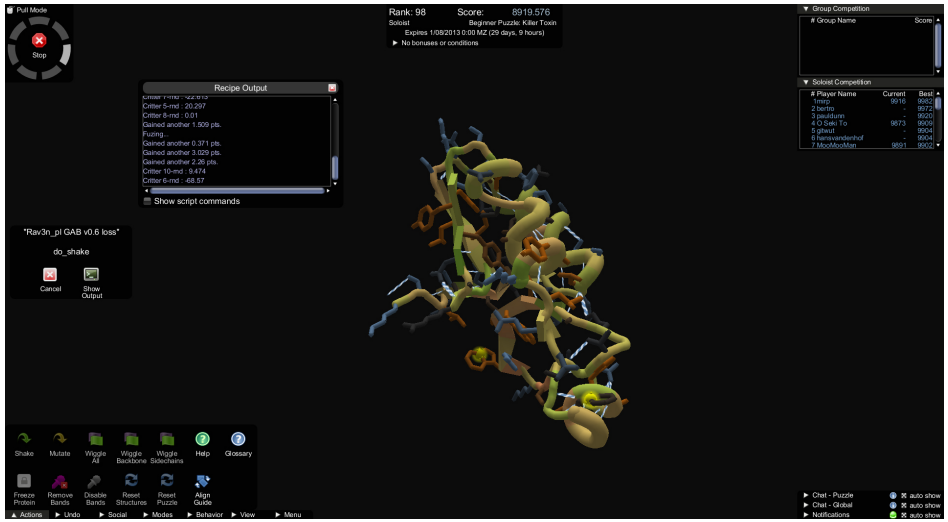
- Used to create difficult metadata for problems (image recognition)
- Used to tag or associate keywords with Google indexed images for Google Image Search

Crowdsourcing Applications: Foldit

An online puzzle to fold the structure of selected protein using game tools

Uni. of Washington center for Game Science and department of Biochemistry

Protein's function 3d structure prediction is computationally challenging



- Foldit's 57,000 players provided useful results that matched or outperformed algorithmically computed solutions

Cooper S, Khatib F, Treuille A, et al. "Predicting protein structures with a multiplayer online game". Nature (2010)

Crowdsourcing Applications: Duolingo

An online platform allowing bilingual volunteers to teach a language using another language

Translate thousands of words and sentences between the languages and arrange those words and sentences into lessons and skills

Exploring Iterative and Parallel Human Computation Processes

Greg Little[1], Lydia B. Chilton[2], Max Goldman[1], Robert C. Miller[1]

[1]MIT CSAIL
{glittle, maxg, rcm}@mit.edu

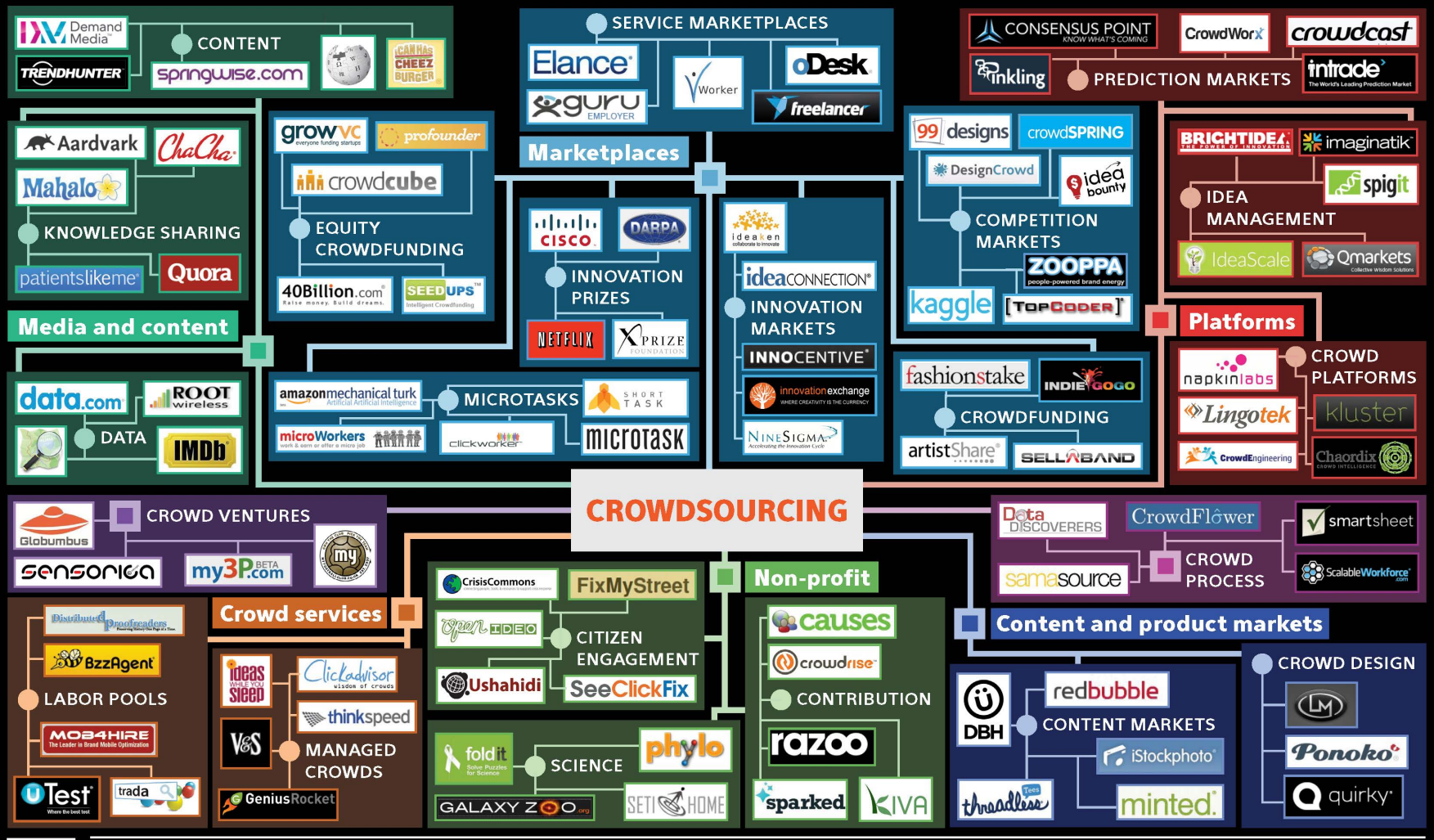
[2]University of Washington
hmslydia@cs.washington.edu

CrowdLang: A Programming Language for the Systematic Exploration of Human Computation Systems

Patrick Minder and Abraham Bernstein

Dynamic and Distributed Information Systems Group
University of Zurich, Switzerland
{lastname}@ifi.uzh.ch

Crowdsourcing landscape Beta v2



Excerpted from

Getting Results From Crowds

by Ross Dawson and Steve Bynghall

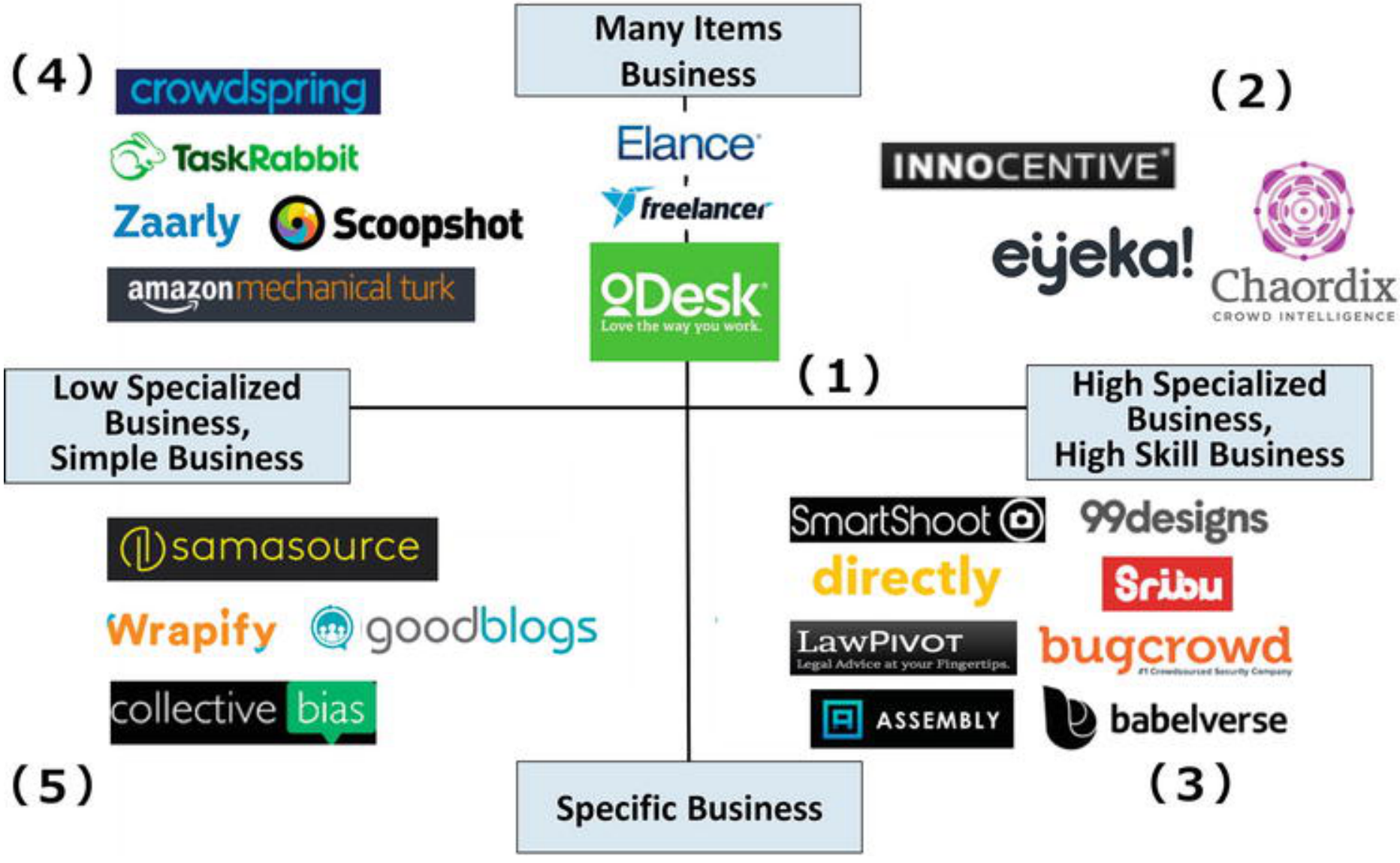
For definitions, analysis, free book chapters, and other crowdsourcing resources go to:

www.resultsfromcrowds.com

Note: examples only, see website for full list of crowdsourcing services

Advanced Human Technologies

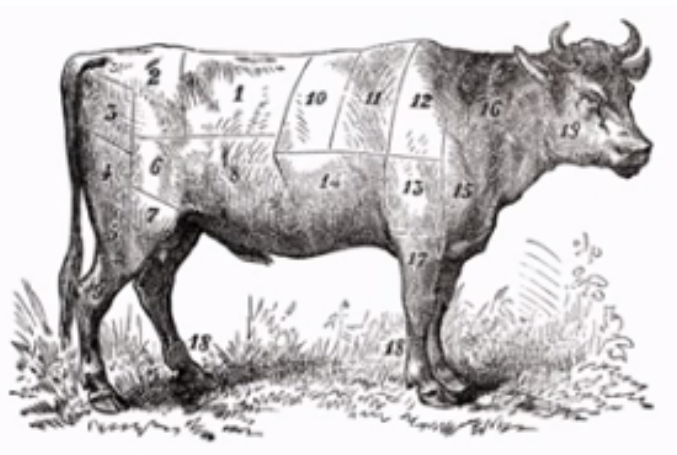
ROSSDAWSON



Tetsuro Saisho (2019), Crowdsourcing Strategy of Information Society

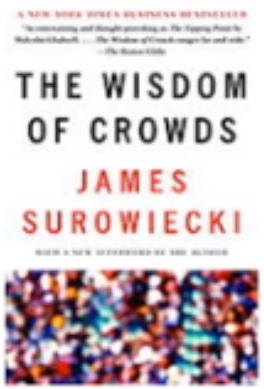
Human Based Computation - Crowd Sourcing

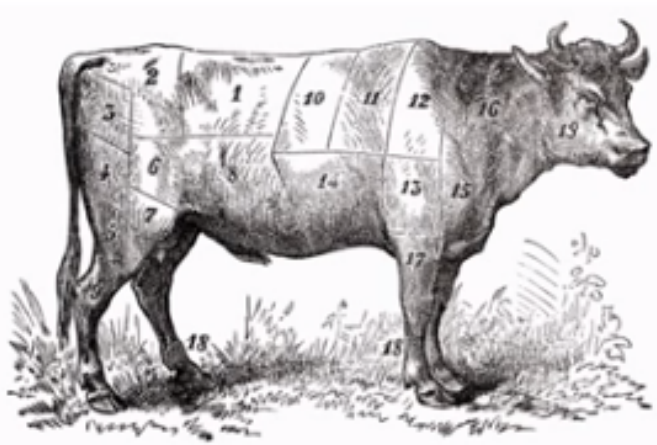
Francis Galton's **1906** demonstrated **"wisdom of the crowd"**—the idea that the collective judgment of a group can often be surprisingly accurate



Francis Galton's **1906** demonstrated **"wisdom of the crowd"**—the idea that the collective judgment of a group can often be surprisingly accurate

The Wisdom of Crowds





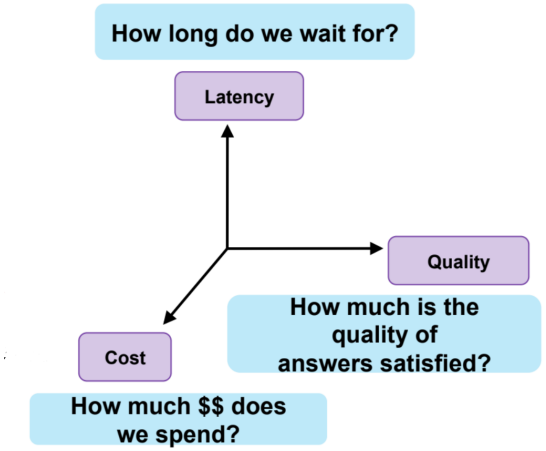
average of 800 guesses = 1,197

actual weight of the ox = 1,198

Crowdsourcing: Three Important Factors

Three main factors effecting the performance, cost, and user satisfaction of a crowdsourcing system:

1. **Latency (execution time):** Worker pool size and job attractiveness
2. **Monetary cost:** Cost per question, task volume and no. of workers, also impacts the scalability and affordability of the system
3. **Quality of answers:** Workers skills, task complexity and proportion of malicious responses; impacts system's reliability and accuracy



Human Based Comparisons - Similarity Triplets

Human Based Computation - Compute or Compare?

Humans have a hard time to

- Explain embedding coordinate
- Quantify a coordinate value
- Evaluate pairwise similarity $sim(A, B) = ?$

But humans are good at

- Differentiating things perceptually
- Comparing objects' features
- Comparing pairwise similarities $sim(A, B) > sim(A, C)$?

Human Based Computation - Compute or Compare

Humans can easily assess that

The first two images more similar than the first and the third

Humans can easily assess that

Rocky mountains

Snow-covered peak

Sea-view

Rocky mountains and snow-covered peak are similar, by scenic view

Human Based Computation - Compute or Compare

Humans can easily assess that

Icecream

Steak

Cookies

Ice cream and cookies are more similar, based on taste

Human Based Computation - Compute or Compare

Humans can easily assess that

Car

Jeep

Truck

A car is more similar to a jeep as compared to a truck, by utility

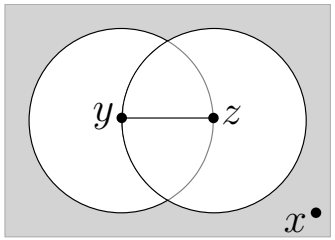
Encoding Comparison Result

Comparison of pair-wise similarities of three objects encoded as triplets

x **is the outlier among the three**

Outlier: $(x, y, z)_O$

$$(x, y, z)_O \implies d(x, y) > d(y, z) \text{ AND } d(x, z) > d(y, z)$$

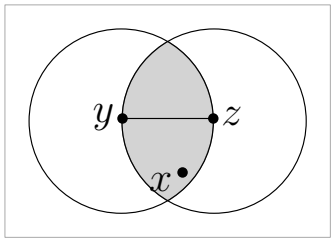


$(x, y, z)_O$

x **is the central among the three**

Central: $(x, y, z)_C$

$$(x, y, z)_C \implies d(x, y) < d(y, z) \text{ AND } d(x, z) < d(y, z)$$

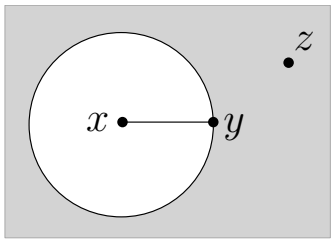


$(x, y, z)_C$

x **is the closer to** y **than** z

Anchor: $(x, y, z)_A$

$$(x, y, z)_A \implies d(x, y) < d(x, z)$$



$(x, y, z)_A$

Crowd sourcing: Find Max

Input: An array A of n distinct numbers
Output: The largest number $x \in A$ and its index

Algorithm FindMax(A)

```
max ← A[1]                          ▷ A[1] is maximum of A[1 ··· 1]
for i = 2 to n do
    if A[i] > max then
        max ← A[i]                  ▷ Update max if A[i] is larger
```

Runtime is $n - 1$ comparisons

Crowd-Sourcing Find Max: Find the "best" or "most popular" item from a pool of options based on user inputs, without comparing **all** items

▷ Commonly used in crowdsourcing platforms (e.g., Amazon ratings) to determine top-rated or most-liked products

Crowd sourcing: Find Max

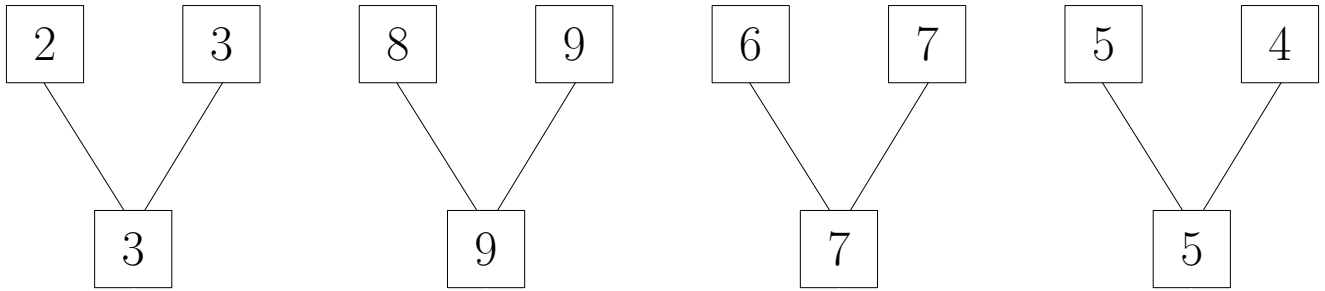
Input: An array A of n distinct numbers
Output: The largest number $x \in A$ and its index

Tournament Style Algorithm

Crowd sourcing: Find Max

Input: An array A of n distinct numbers
Output: The largest number $x \in A$ and its index

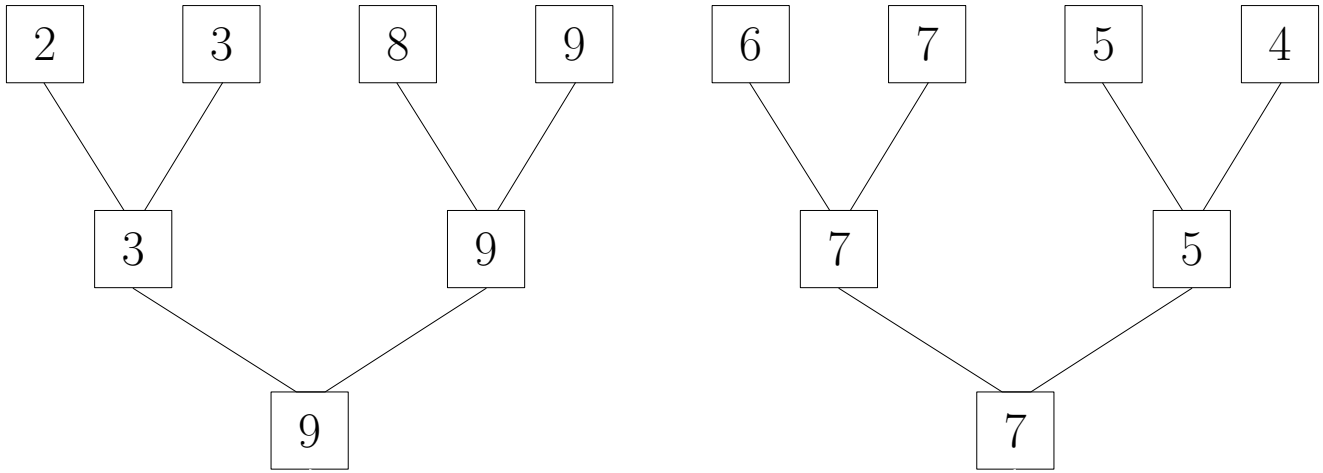
Tournament Style Algorithm



Crowd sourcing: Find Max

Input: An array A of n distinct numbers
Output: The largest number $x \in A$ and its index

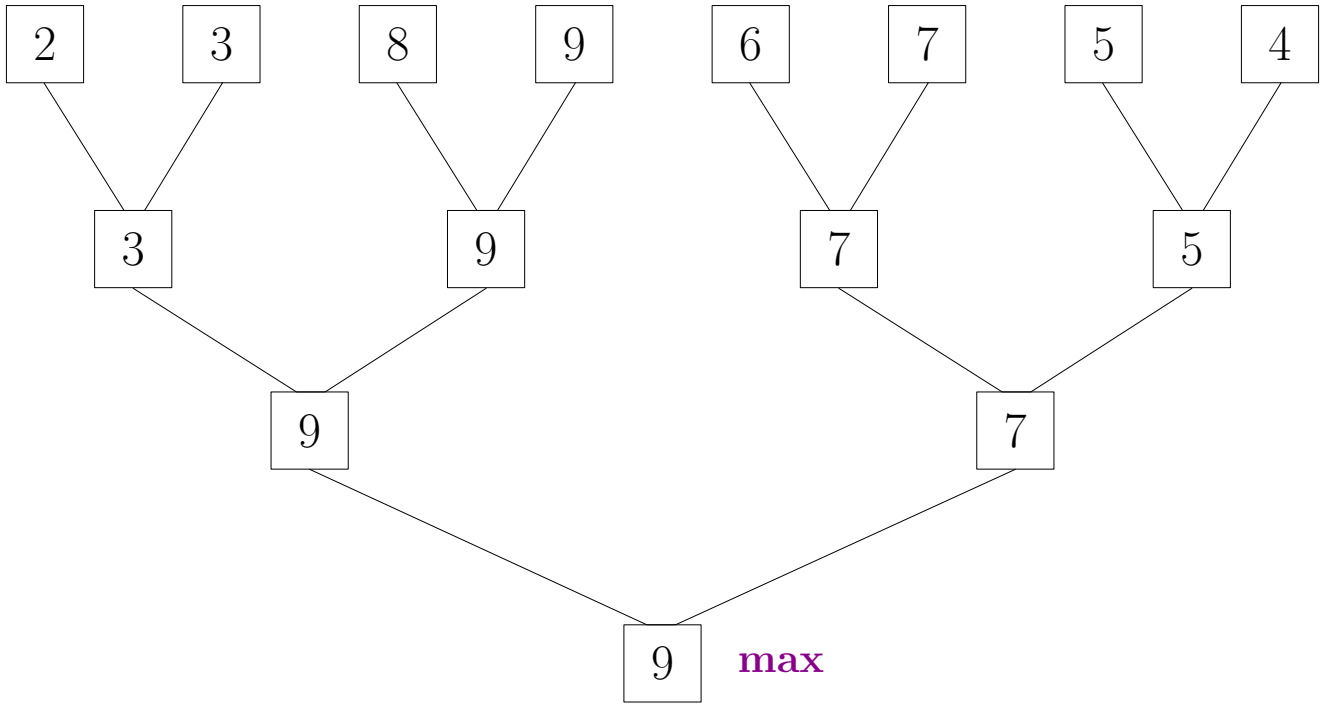
Tournament Style Algorithm



Crowd sourcing: Find Max

Input: An array A of n distinct numbers
Output: The largest number $x \in A$ and its index

Tournament Style Algorithm



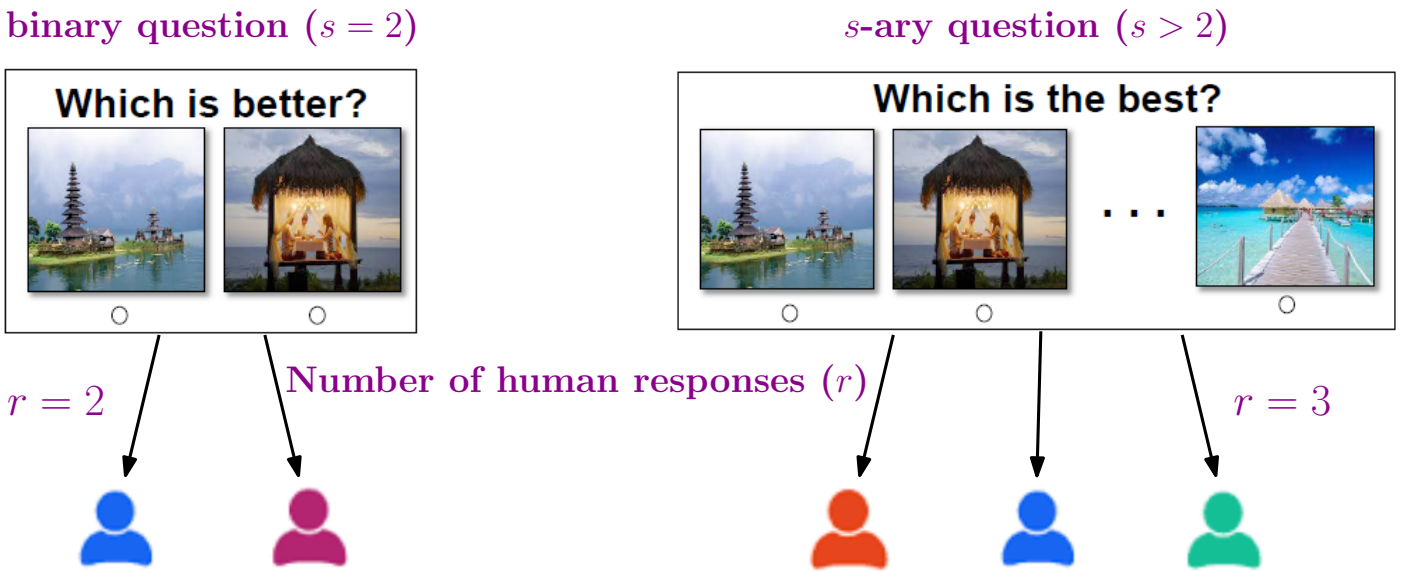
Runtime is $n - 1$ comparisons

Crowd sourcing Max: Bubble Max

Bubble Max adapts a bubble-sort-like method to find maximum by gradually narrowing down the candidate items based on human responses

Leverages human ability to compare multiple items simultaneously

Bubble Max actually is a family of algorithms with parameters controlling the number of items compared and number of questions per comparison



Diagrams adapted from Dongwon Lee @ Penn State Univeristy

It iteratively compares groups of items and eliminates weaker options

Crowd sourcing Max: Bubble Max

Input: A set E of items

Parameters: $r_1, r_2, \ldots,$ $s_1, s_2, \ldots$

- s_i : Size of subset compared in round i
- r_i : Number of human responses in round i

Output: The maximum item from E

Algorithm 2 Bubble Max Algorithm $\{s_i\}, \{r_i\}$

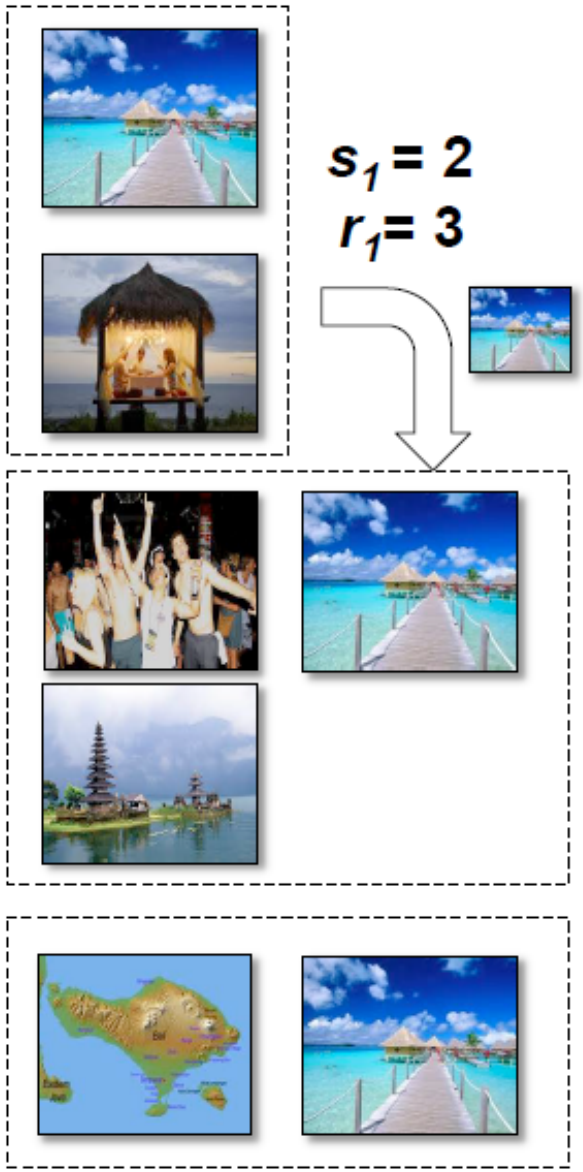
if $|E| = \{e\}$ **then**
  return e
$S_1 \leftarrow$ random subset of E of size $\min(s_1, |E|)$
$E \leftarrow E \setminus S_1$
$w \leftarrow \textsc{Comp}(S_1, r_1)$
$i \leftarrow 2$
while $E \neq \emptyset$ **do**
  $S_i \leftarrow$ random subset of E of size $\min(s_i - 1, |E|)$
  $E \leftarrow E \setminus S_i$
  $S_i' \leftarrow S_i \cup \{w\}$
  $w \leftarrow \textsc{Comp}(S_i, r_i)$
  $i \leftarrow i + 1$
return w

Crowd sourcing Max: Bubble Max

- s_i : Size of the sets compared by humans at step in round i
- r_i : Number of human responses sought in round i. This number helps manage the trade-off between accuracy and the cost of human labor
- The algorithm dynamically adjusts the comparison set size and the number of responses to optimize for both error rates and operational costs

- Cost: Determined by the number of human comparisons $r \times \text{Cost}(s)$ across all steps
- Quality: Probability of correctly identifying the maximum item, influenced by aggregation rules and error models
- Execution Time: Measured by the number of steps required, reflecting the latency in obtaining results

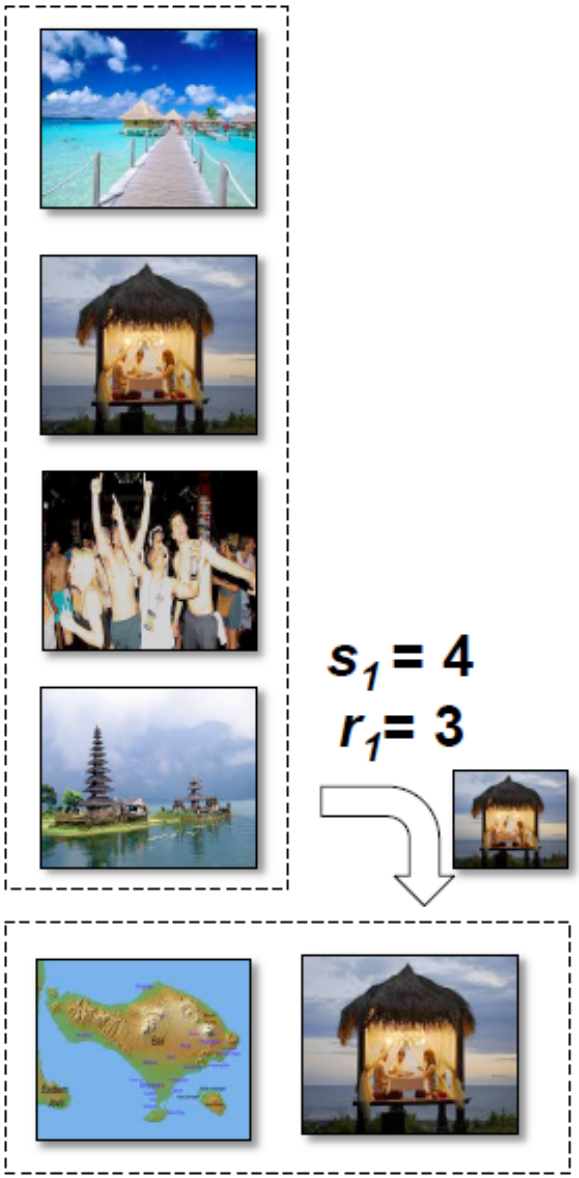
Crowd sourcing Max: Bubble Max



- $N = 5$
- $Rounds = 3$
- $\#\ of\ questions = r_1 + r_2 + r_3 = 11$

source: Dongwon Lee @ Penn State Univeristy

Crowd sourcing Max: Bubble Max



- $N = 5$
- $Rounds = 2$
- $\#\ of\ questions = r_1 + r_2 = 8$

source: Dongwon Lee @ Penn State Univeristy

Crowd sourcing Max: Tournament Max

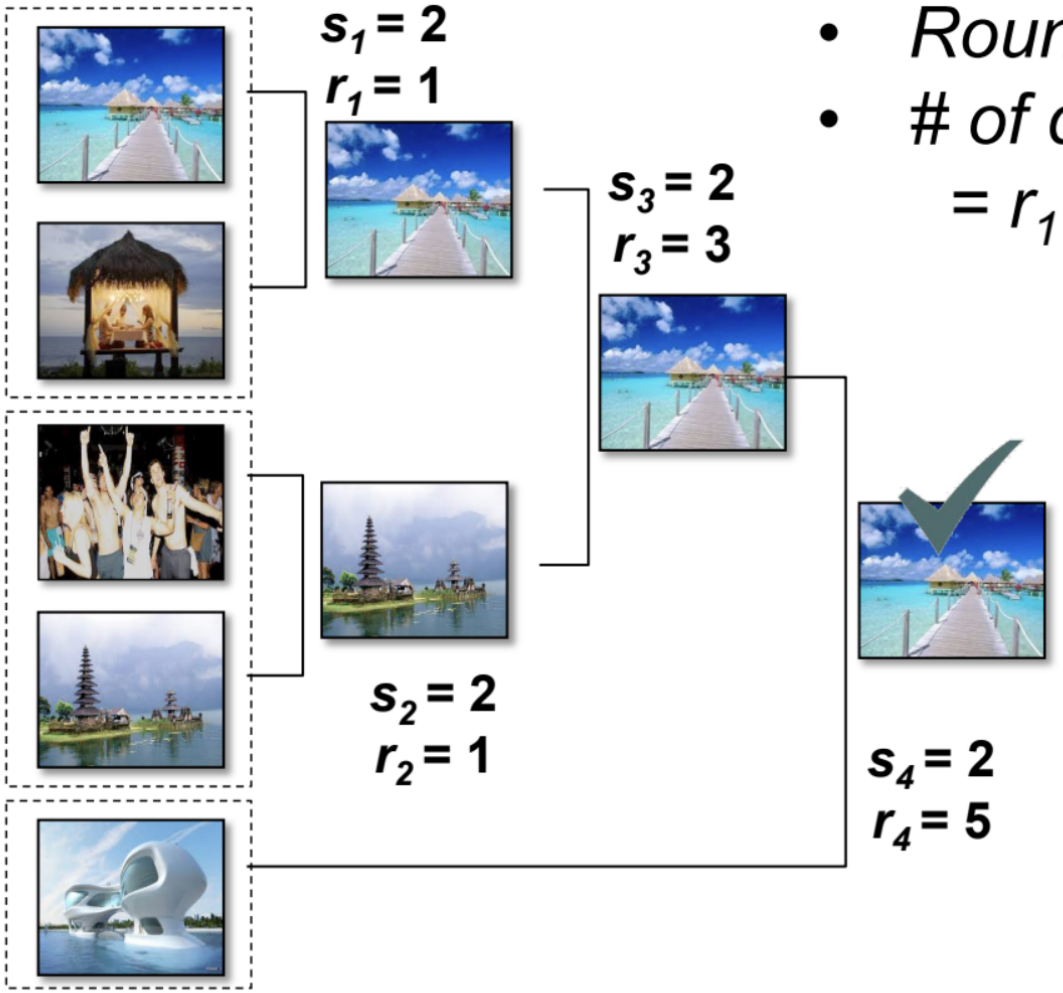
Group items into a tournament structure where winners from each group progress until one remains

- Groups of items compete in rounds
- Winner of each group progresses to the next round until the best item is determined

Algorithm 3 Tournament Max Algorithm $\{s_i\}, \{r_i\}$

$i \leftarrow 1$
$E_i \leftarrow E$
while $|E_i| \neq 1$ **do**
  Partition E_i into disjoint sets S_j, $|S_j| = s_i$ ▷ the last set may have fewer items
  $E_{i+1} \leftarrow$ results of $\textsc{Comp}(S_j, r_i)$ for all j
  $i \leftarrow i + 1$
return the remaining item in E_i

Crowd sourcing Max: Tournament Max



- $N = 5$
- $Rounds = 3$
- # of questions
  $= r_1 + r_2 + r_3 + r_4 = 10$

source: Dongwon Lee @ Penn State Univeristy

Bubble Max vs Tournament Max

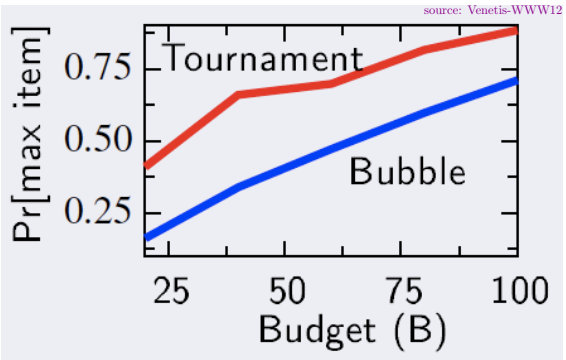
Bubble Max:

- Total comparisons $= r_1 + r_2 + ... + r_n$ (depends on the size of sets, s_i).
- Worst case, with s_i =2, O(n) comparisons needed.

Tournament Max:

- Fewer rounds than Bubble Max but may require larger comparison sets s_i per round.
- Worst case, with $s_i = 2$, O(n) comparisons needed.

Bubble Max is a special case of Tournament Max



Rating Aggregation

Rating Aggregation

Rating Aggregation Combining several individual ratings to create an overall score representing a collective insight

- Reduces many individual ratings into a simple metric – Aggregates multiple viewpoints and consolidates information into a single score
- It can highlight product quality, trust, and satisfaction levels and helps consumers make informed decisions
- Aggregation must takes into account both the number of reviews and the diversity of scores
- Trust worthiness and credibility of reviews can be impacted by aggregation mechanisms

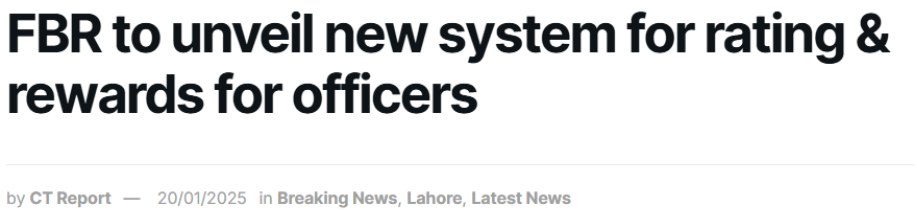
Crowdsourcing in rating aggregation provides the volume and diversity of inputs needed for reliable aggregation

Applications of Rating Aggregation

Rating Aggregation enables benchmarking, comparison, ranking,

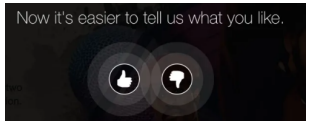
- E-Commerce (e.g., Amazon): Ratings help prioritize search results and recommend products
- Social Media: Reviews/likes are aggregated to enhance content visibility
- Crowd-Sourced Platforms (e.g., Trip Advisor): Aggregated reviews provide users with collective insight into the quality of restaurants, hotels, etc.
- Employee Performance Ratings

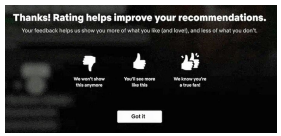
Review system is used everywhere. Graduate applications, tenure review, research papers reviews



Review System

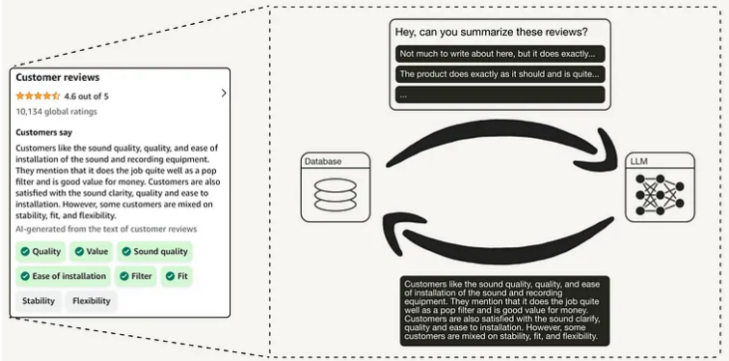
- Rating: 1–5, -3 – 3, 1–10, thumbs up thumbs down



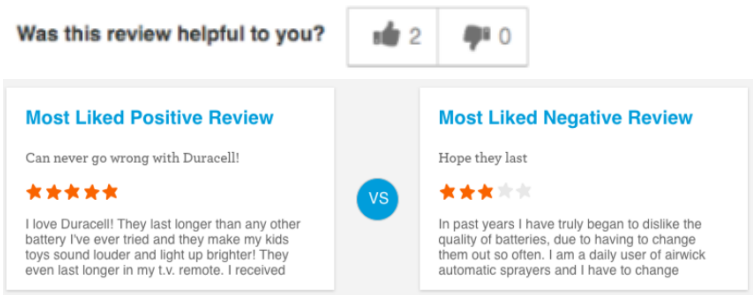


- Review: Text could be short like a line or multiple pages

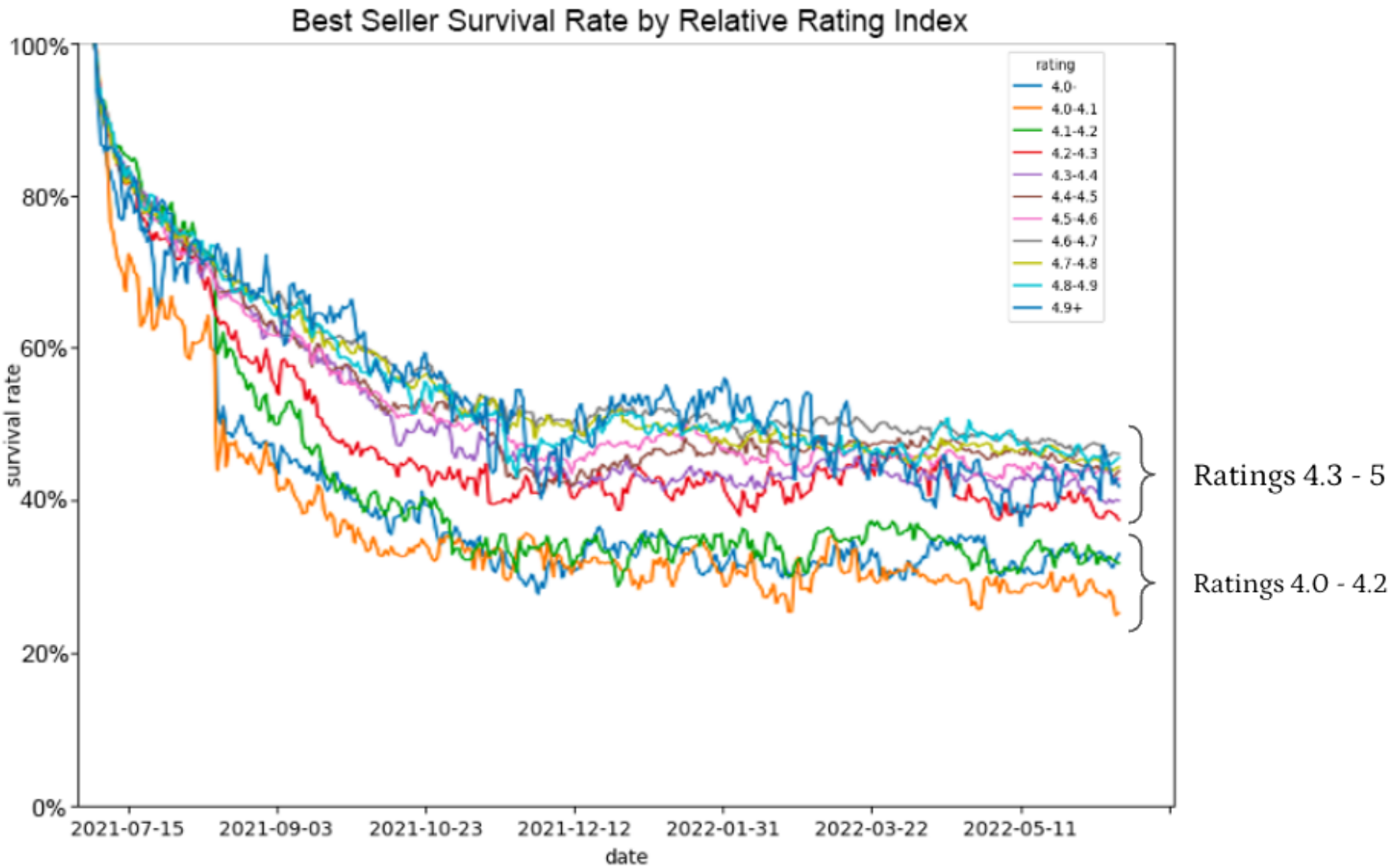
Recreating Amazon's New Generative AI Feature: Product Review Summaries (medium.com)



- Reviews of Review:



Importance of Rating



Durable Dominance on Amazon https://www.momentumcommerce.com/amazon-star-ratings-drop/

Challenges in Crowdsourced Rating Aggregation

- Diverse User Opinions: Users' varying backgrounds and preferences
- Biases and Extremes: Extreme ratings based on personal bias
- Manipulation: Deliberately inflated/deflated ratings to promote/harm
  - Can everyone review or only buyers, anonymous reviews, fake reviews
- Sparse Data: A few users provide ratings–
  - How many reviews are good enough? even if they are independent
- Subjectivity and Objectivity: Product's nature (movie vs PC)

https://www.linkedin.com/pulse/subjectivity-vs-objectivity-reviews-dane-cobain/

- Independence of Reviews: Reviews can be biased by other reviews
- Temporal Effects: Older ratings may no longer be relevant
- Rating Scale: Is 0–10 the same as −5–5
- Performance Metric: For who, Amazon, seller, buyer?

Types of Attributes: Nominal/Categorical

- Possible values are symbols, labels or names of things, categories
  - gender, major, state, color

  Describe a feature **qualitatively** and **values have no order**
- **Not quantitative**, arithmetic operations can't be performed on them

  $\text{male} - \text{female} = ??$ $\text{green} + \text{blue} = ??$
- Can code by numbers (numeric symbols) e.g. postal codes, roll num

Can compute
- frequency of values and the most frequent value
- ~~middle value~~
- ~~average value of an attribute~~

Binary Attribute: - special case of nominal TRUE/FALSE, Pass/Fail, 0/1

- **Symmetric:** Both symbols carry the same weight e.g. gender
- **Asymmetric:** Both symbols are not equally important, e.g. Pass/Fail

Types of Attributes: Ordinal Attributes

- Possible **values have meaningful order**
  - Grades : A,B,C,D
  - Serving Sizes : Small, Medium, Large
  - Ratings : poor, average, excellent
- **No quantified difference between two levels**
  - A is higher/better than B but
  - Cannot quantify how much higher is A than B, or
  - if the difference between A and B the same as the difference between B and C
- Can be obtained by discretizing numeric quantities (data reduction)

Can compute

- frequency of values and the most frequent value
- middle value
- ~~average value of an attribute~~

Types of Attributes: Numeric Attributes

- Quantitative and measurable
- can **quantify the difference between two values**
  - temperature, age, number of courses, height, years of experience

Can compute

- frequency of values and the most frequent value
- middle value
- average value of an attribute

- **Discrete Numeric Attributes**
  - values come from a finite or countably infinite sets
- **Continuous Numeric Attributes**
  - values are real (continuous)
- **Interval-Scaled:** No point 0, ratios have no meaning
  - e.g. Temperature in Celsius. 30° is not double as hot as 15°
- **Ratio-Scaled:** Well-Defined point 0, ratios are meaningful
  - e.g. Temperature in Kelvin. 30° is double as hot as 15°

Mapping Vectors to Scalars

- Let $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T \in \mathbb{R}^n$
- compare its competing estimates $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^T$ and $\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix}^T$
- Error vectors $\mathbf{e}_y = \mathbf{x} - \mathbf{y} = \begin{bmatrix} x_1 - y_1 \\ \vdots \\ x_n - y_n \end{bmatrix}$ and $\mathbf{e}_z = \begin{bmatrix} x_1 - z_1 \\ \vdots \\ x_n - z_n \end{bmatrix}$
- e.g. $\mathbf{e}_y = \begin{bmatrix} 10 & -10 & 10 & 20 \end{bmatrix}$ and $\mathbf{e}_z = \begin{bmatrix} 20 & -5 & 0 & 20 \end{bmatrix}$
- Need to map $\mathbf{e}_y$ and $\mathbf{e}_z$ to real numbers and compare
- Compare lengths $\|\mathbf{e}_y\| = \sqrt{10^2 + (-10)^2 + 10^2 + 20^2} = 26.45$, $\|\mathbf{e}_z\| = 28.72$
- Since smaller are better, $\mathbf{y}$ is a better estimate of $\mathbf{x}$
- One can argue that with a different mapping, $\mathbf{z}$ is better
  $\|\mathbf{e}_y\|_1 = |10| + |-10| + |10| + |20| = 50$, $\|\mathbf{e}_z\|_1 = |20| + |-5| + |0| + |20| = 45$
- Note the absolute value sign $\because$ error on either side is bad
- No universally good mapping of vectors to numbers
- Amazon product ratings is not a vector, it is a time series

Measures of Central Tendencies: Mean

For a dataset $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$

- **(Arithmetic) Mean** is the average of the data set
  - ▷ This definition readily extend to higher dimensional data

$$\overline{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- **Harmonic Mean**

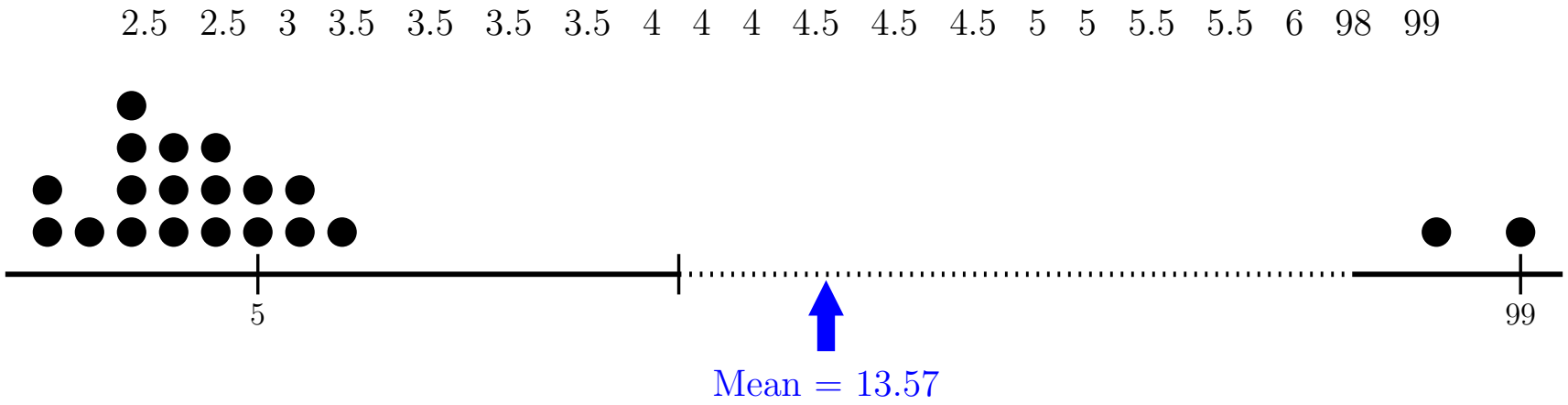
$$\overline{x} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

- **Geometric Mean**

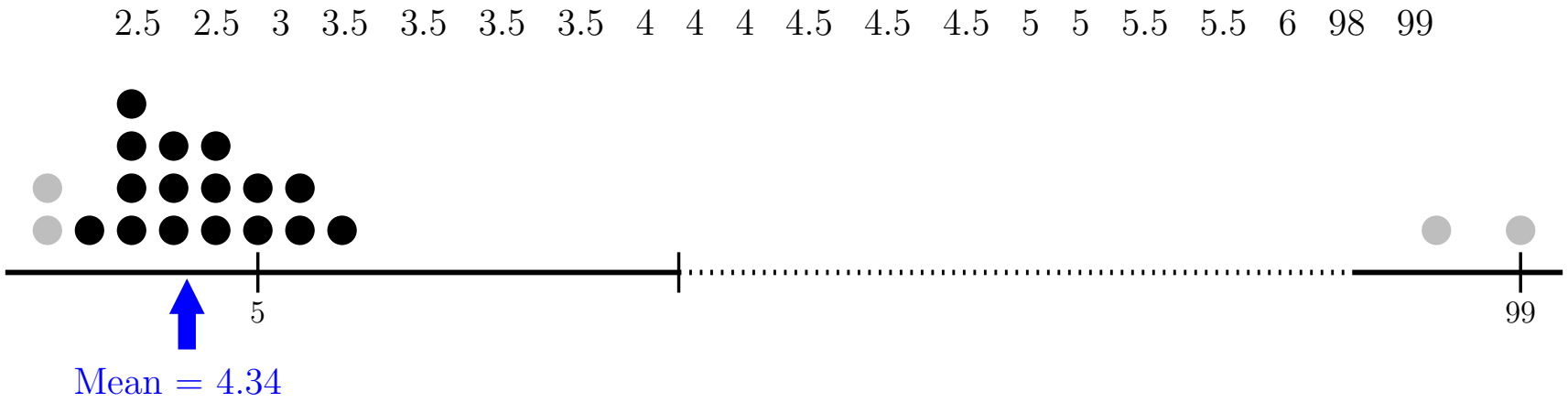
$$\overline{x} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

Trimmed or Truncated Mean

- Arithmetic mean is sensitive to outliers ▷ unstable statistic
- Just one very high/low value (think $\pm\infty$) makes mean very high/low



Trimmed Mean: Ignore $k\%$ of values at both extremes to compute mean



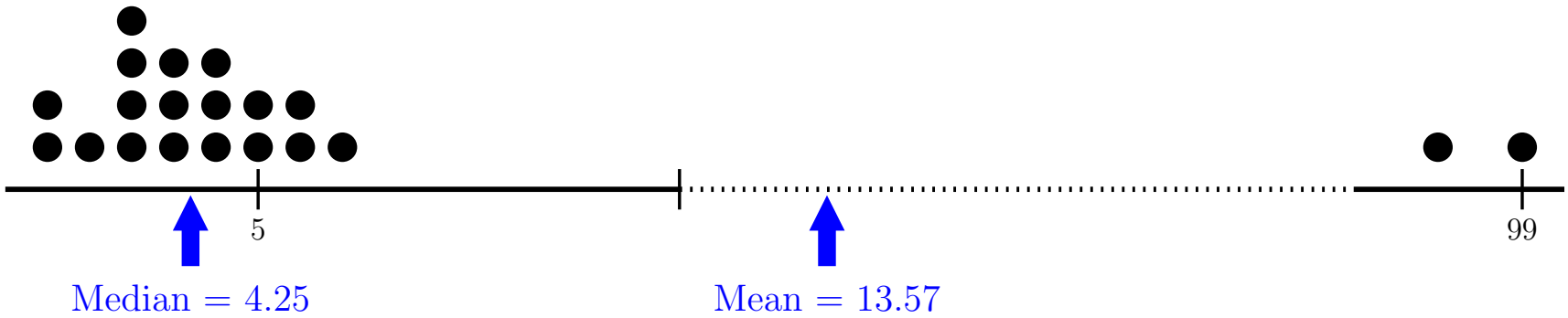
Median Aggregation

Alternatives to simple and weighted mean, esp. for robustness to outliers

Median: The value that divides the ratings into two equal halves

$$\text{Median} = \text{middle value of the sorted ratings}$$

- Median is less sensitive to outliers as compared to mean
- Median is good for asymmetric distributions and where data has outliers



- Various possible definitions for median of higher dimensional data
- Mean together with variance (see below) has nice properties

Mode Aggregation

Alternatives to the simple and weighted means, esp. for nominal data

Mode: The most frequent element in the dataset. Best suited for categorical or discrete data-

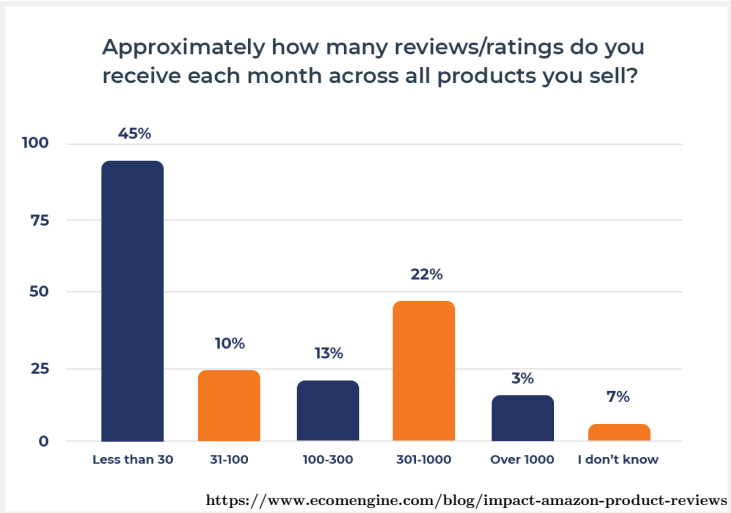
- **Mode** is the most frequent element
- Can have more than one modes
  - unimodal (one mode in data)
  - multi-modal (bimodal, trimodal): more than one modes in data

Not the same as majority element (a value with frequency $> 50\%$)

Naive Rating Aggregation: Mean

Why simple mean worked in the Galton's experiment?

1. Task Definition:
   - There was a correct/objective answer
   - In many cases there is no ground truth and the task is subjective
2. Unbiased and Independent Estimates:
   - Everybody guessed independently (without looking at others' guesses)
   - Dependence of participants' action breaks the wisdom of crowds and turn it into an information cascade
3. Review Population: 787 was a good enough large number



Naive Rating Aggregation: Mean

Average number of stars

n ratings scaled 1-star to 5-stars $\quad$ r_i ratings give i-stars

$$\text{Naive Average} = \frac{1}{n}\sum_{i=1}^{5} i \times r_i$$

- Product A: Average stars: 4.5, Number of reviews 2
- Product B: Average stars 4, Number of reviews 1000
  - ▷ Evaluate the change if one more 5 stars review appears for A and B
- Product C: Average stars: 4.3, Number of reviews 100
- Product D: Average stars 4.1, Number of reviews 200

Naive Rating Aggregation: Cumulative Rating

Since the number of reviews is an issue, what if we just add up the total number of stars a product has accumulated over all reviews

It favors old/popular product not highly rated products

- Product A: 100 reviews with 5 stars
- Product B: 1M reviews with 2 stars 4

The rating is unbounded in this case

Naive Rating Aggregation: Median Rating

What if we use the **median rating** a product has received over all reviews?

This approach fails when the distribution of ratings is skewed

Product A Ratings:

- 1000 × 1-star
- 1 × 3-star
- 1000 × 5-star
- **Median Rating = 3 stars**

Product B Ratings:

- 100 × 3-star
- 1 × 3-star
- 1000 × 5-star
- **Median Rating = 3 stars**

Product C Ratings:

- 200 × 2-star
- 1 × 3-star
- 200 × 3-star
- **Median Rating = 3 stars**

Product D Ratings:

- 10 × 3-star
- 1 × 3-star
- 10 × 4-star
- **Median Rating = 3 stars**

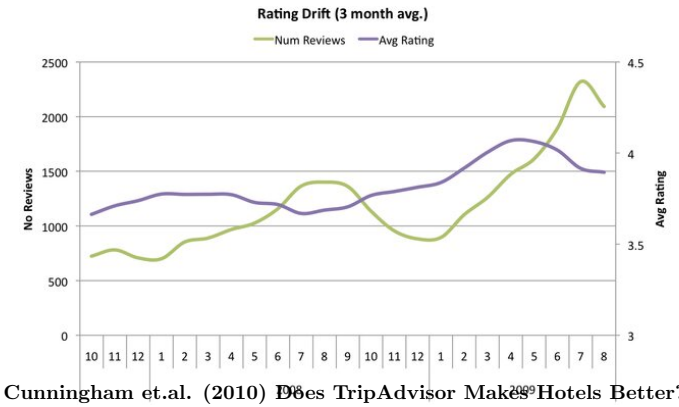
The median rating ignores rating distribution!

Visualizing Raw Rating Data

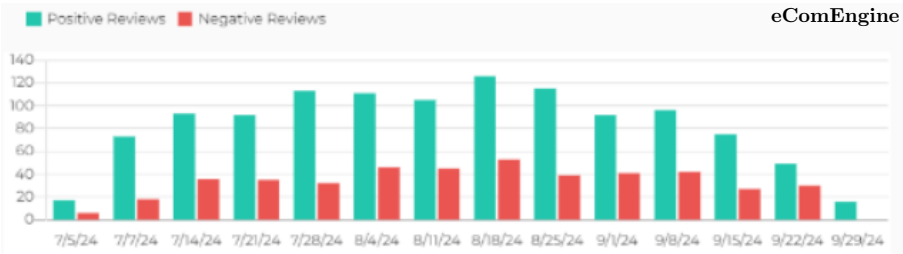
One can display ratings data - **Aggregating is a heuristic**

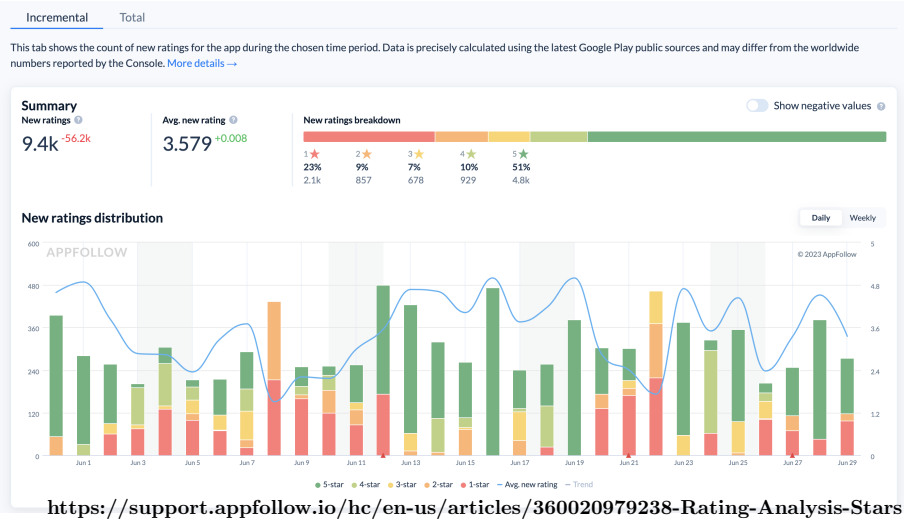
- Bar Charts: Generally used for a nominal and ordinal variables
- Height of bar represent frequencies of each symbol (value)
- Can reveal variables that have no or limited information e.g. constants
- We can use pie charts for the same purpose too
- Humans perceive difference in lengths better than in angles

Reviews Rating over Time



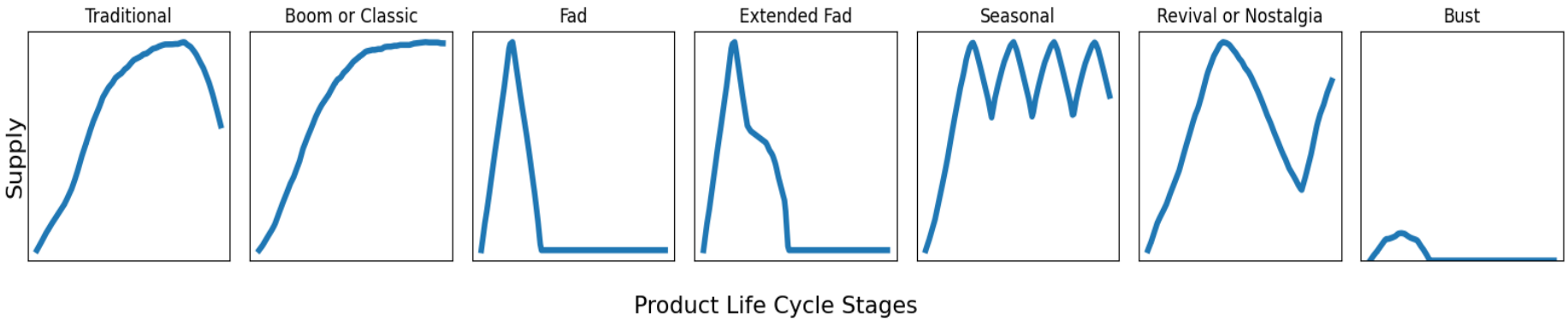
Cunningham et.al. (2010) Does TripAdvisor Makes Hotels Better?





https://support.appfollow.io/hc/en-us/articles/360020979238-Rating-Analysis-Stars

Product Life Cycle



- Traditional: Grows, matures, and eventually declines
- Boom or Classic: Sustains maturity for long time with minimal decline
- Fad: Rapidly gains popularity but fades quickly
- Extended Fad: Similar to a fad but declines more gradually
- Seasonal: Peaks and dips in demand due to seasonal trends
- Revival or Nostalgia: Declines but regains popularity later
- Bust: Fails quickly with little to no growth

Mean Aggregation

Let x be the truth/fact/actual value of a product (e.g., the ox)

- n actors in the crowd
- Estimate of actor i is y_i

$$y_i = x + \epsilon_i(x)$$

  - ϵ_i depends only on i and x, ϵ_i is independent of ϵ_j for $j \neq i$
  - For all i, ϵ_i is unbiased i.e., $E_x[\epsilon_i(x)] = 0$– This expectation is over the distribution of x

    ▷ Think of x being the true value/rating of a product chosen from a collection of products each with its true rating

- Error Measure: MSE between the aggregate and x

Mean Aggregation: Average of Errors

The average of (expected, mean squared) errors (AE) is

$$E_{AE} = \frac{1}{n}\sum_{i=1}^{n} E_x\left[(\epsilon_i(x))^2\right]$$

The expected, mean-squared error of the average (EA) is

$$E_{EA} = E_x\left[\left(\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(x)\right)^2\right] = \frac{1}{n^2}E_x\left[\left(\sum_{i=1}^{n}\epsilon_i(x)\right)^2\right]$$

Using $y_i = x + \epsilon_i(x)$ we get

$$\frac{1}{n}\left(\sum_{i=1}^{n}(y_i - x)\right) = \frac{1}{n}\left(\sum_{i=1}^{n}\epsilon_i(x)\right)$$

Mean Aggregation: Average of Errors

Recall $(a+b)^2 = a^2 + b^2 + 2ab$

$$E_{EA} = \frac{1}{n^2} E_x\left[\left(\sum_{i=1}^{n} \epsilon_i(x)\right)^2\right] = \frac{1}{n^2} E_x\left[\sum_{i=1}^{n} \epsilon_i(x)^2 + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} \epsilon_i(x)\epsilon_j(x)\right]$$

By linearity of expectation

$$E_{EA} = \frac{1}{n^2}\sum_{i=1}^{n} E_x\left[\epsilon_i(x)^2\right] + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} E_x\left[\epsilon_i(x)\epsilon_j(x)\right]$$

When $\epsilon_i(x)$ and $\epsilon_j(x)$ are independent, i.e., $E_x[\epsilon_i(x)\epsilon_j(x)] = 0$, we get

$$E_{EA} = \frac{1}{n^2}\sum_{i=1}^{n} E_x\left[\epsilon_i(x)^2\right] = \frac{1}{n} E_{AE}$$

Mean Aggregation: Average of Errors

We got

$$E_{EA} = \frac{1}{n} E_{AE} \qquad \text{RMSE}_{EA} = \frac{1}{\sqrt{n}} \text{RMSE}_{AE}$$

E_{EA} (error of average) is smaller than the average of errors, i.e., if we aggregate estimates of many actors the error decreases (wisdom of crowd)

The decrease is by a factor of n or $\sqrt{n}$, meaning with larger crowd the error is going to be even smaller

This is true only, if estimates are independent and unbiased

If estimate are completely dependent (copies of each other), then

$$E_{EA} = E_{AE}$$

The truth is somewhere in between!

Ranking - Bayesian Ranking

Ranking vs Rating Aggregation

- Aggregation: The process of combining ratings to produce a representative score, such as a mean or weighted average
- Ranking: The process of ordering items based on their aggregated scores or other relevant factors

Aggregation:

- Focuses on computing a representative score from multiple ratings
- Methods like mean, median, or weighted averages are common
- Objective: Summarize the collective opinion of users

Ranking:

- Involves sorting items in order of preference or relevance
- Ranking may consider other factors (popularity, recency or diversity)

Bayesian Ranking Approach

- If a product has "few" ratings, estimate its rating closer to a global average R (prior).
- If a product has enough ratings, its estimated rating should converge to its actual average.
- The estimated rating $\tilde{r}_i$ lies between r_i (product's average rating) and R.
- More reviews shift $\tilde{r}_i$ closer to r_i.

Bayesian Ranking

Bayesian analysis is really good for such situation, when you want to estimate something but you don't have enough data

Give scores to product as a convex combination of r_i and R

$$\tilde{r}_i = \beta r_i + (1 - \beta) R$$

Generally, $\beta = \frac{n_i}{n_i + N}$

Note that when n_i is very large (relative to N), then $\beta \to 1$, and $\tilde{r}_i \to r_i$ and vice-versa

- $\beta \in [0, 1]$
- β is monotonically increasing with increasing n_i, approaching 1 as n_i approaches ∞
- β is close to 0, when n_i is 0

Dynamic Nature of Bayesian Ratings: Sequential Updates

Sequential Updates:

- Initially: Few reviews with high ratings
- As more reviews are added, ratings tend to vary, and the Bayesian adjustment provides a stabilizing effect
- Example updates:
  - After 10 reviews: 4.5 stars
  - After 50 reviews: 4.2 stars
  - After 100 reviews: 4.1 stars
- Each update recalculates the Bayesian estimate, showing how perceptions of product quality can evolve

Why is Bayesian Ranking Needed?

- When ranking products in the same category, relying on average or cumulative ratings alone can be misleading
- The number of reviews is critical—fewer reviews make the ranking unreliable

Challenges with Simple Ranking Approaches

- Ranking only products with at least n_0 ratings is a simple approach but has limitations
- How should we choose the threshold n_0? Too low or too high can create bias

Bayesian Ranking

- Gadget A: 5 reviews, average rating 4.6 stars.
- Gadget B: 200 reviews, average rating 4.4 stars.
- Global average R is 4.0, N is 50.
- Bayesian adjusted rating for Gadget A:

$$\hat{r}_A = \frac{50 \times 4.0 + 5 \times 4.6}{50 + 5} \approx 4.05$$

- Bayesian adjusted rating for Gadget B:

$$\hat{r}_B = \frac{50 \times 4.0 + 200 \times 4.4}{50 + 200} \approx 4.32$$

- This adjustment reduces the influence of the smaller sample size for Gadget A, positioning Gadget B as the higher-quality option when considering the volume of feedback.

Analyzing Rating Patterns for a New Tech Gadget:

- Initial few ratings for a new gadget are highly positive
  - ▷ Generally, initial ratings are from passionate fans
- A Bayesian adjustment is applied to evaluate its performance against established products with similar characteristics

$$\hat{r} = \frac{50 \times 4.0 + 25 \times 4.8}{50 + 25} \approx 4.27$$

- This adjusted rating, which might be lower than the simple average, provides a more realistic expectation of the gadget's quality

Bayesian Ranking: Real world examples

Many companies use Bayesian ranking to aggregate ratings and reduce biases in rankings

- **IMDb:** Uses a Bayesian-adjusted weighted average to rank the Top 250 movies:

$$R = \frac{v}{v+m} r + \frac{m}{v+m} C$$

  Ensures movies with few ratings don't dominate rankings

  - R : final Bayesian rating
  - v : number of votes
  - m : minimum votes required

- **Beer Advocate:** Implements **Bayesian smoothing** to prevent spam and rating manipulation. Requires **minimum votes** (N_{min}) for a beer to be ranked. Caps **maximum votes** (N_{max}) to prevent dominance by a single beer

Bayesian Ranking: Real world examples

Bayesian ranking is also widely used in e-commerce, social media, and recommendation systems

- **Amazon:** Uses Bayesian inference to weigh product reviews: - Prioritizes verified purchases - Adjusts early reviews using a prior distribution - Identifies and downweighs biased reviews (e.g., fake or incentivized reviews)
- **Reddit & Stack Overflow:** Applies Bayesian averaging to prevent posts with very few votes from reaching the top too quickly. Adjusts **karma scores** based on recency and voting patterns
- **Netflix & Spotify:** Uses Bayesian hierarchical models to predict ratings based on:
  - User behavior and preferences
  - Recency of ratings and interactions
  - Similar users' choices

How Bayesian ranking can change order

Bayesian ranking can sometimes completely reverse the naive ranking order due to the varying number of ratings each product receives.

- The naive rating for each product in the table given below was calculated by simply taking the average rating.
- But when we calulate the bayesian adjusted rating depending on the Global average R, we reduce the influence of the smaller sample size.

DVD player	Num Ratings	Rank	Rating	Bayesian Rating
Panasonic	11	1	4.18	3.57
Philips	37	2	4	3.62
Sony	67	3	3.47	3.55
Toshiba	54	5	3.407	3.533

Rating and Ranking System of Amazon Products

Amazon's rating and ranking system is designed to provide customers with useful insights into product quality while mitigating the impact of rating manipulation or bias

- Rating System: Aggregates individual ratings using Bayesian methods to prevent bias due to small sample sizes
- Ranking System: Combines product ratings with other factors, e.g., sales velocity and review quality, to rank products within search results
- Features Considered:
  - Number of reviews
  - Average rating
  - Review recency and helpfulness

Variance and Bias in Machine Learning

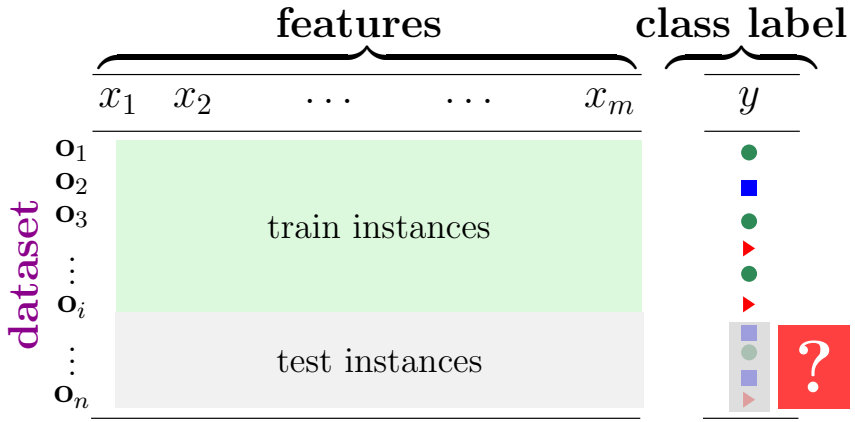
Classification

Classification is a supervised task

Input: A collection of objects

feature vectors with class labels

Output: A model for the class attribute as a function of other attributes



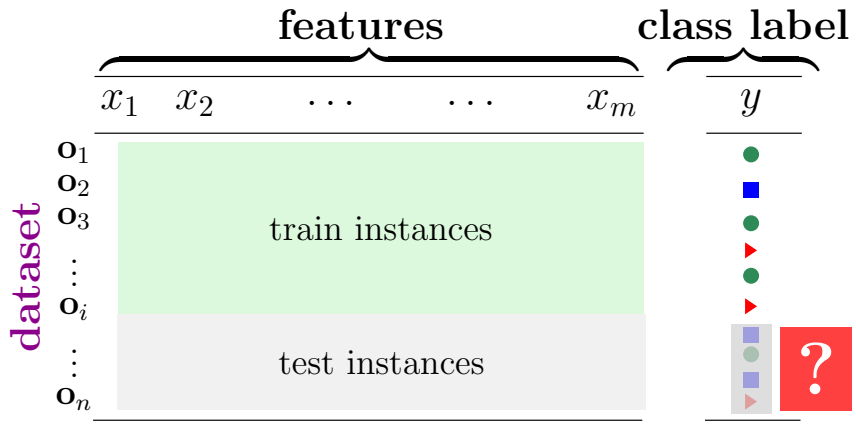
- Training Set: Instances whose class labels are used for learning
- Test Set: Instances with same attributes as training set but missing/hidden class labels
- **Goal:** Model should accurately assign class labels to unlabeled instances

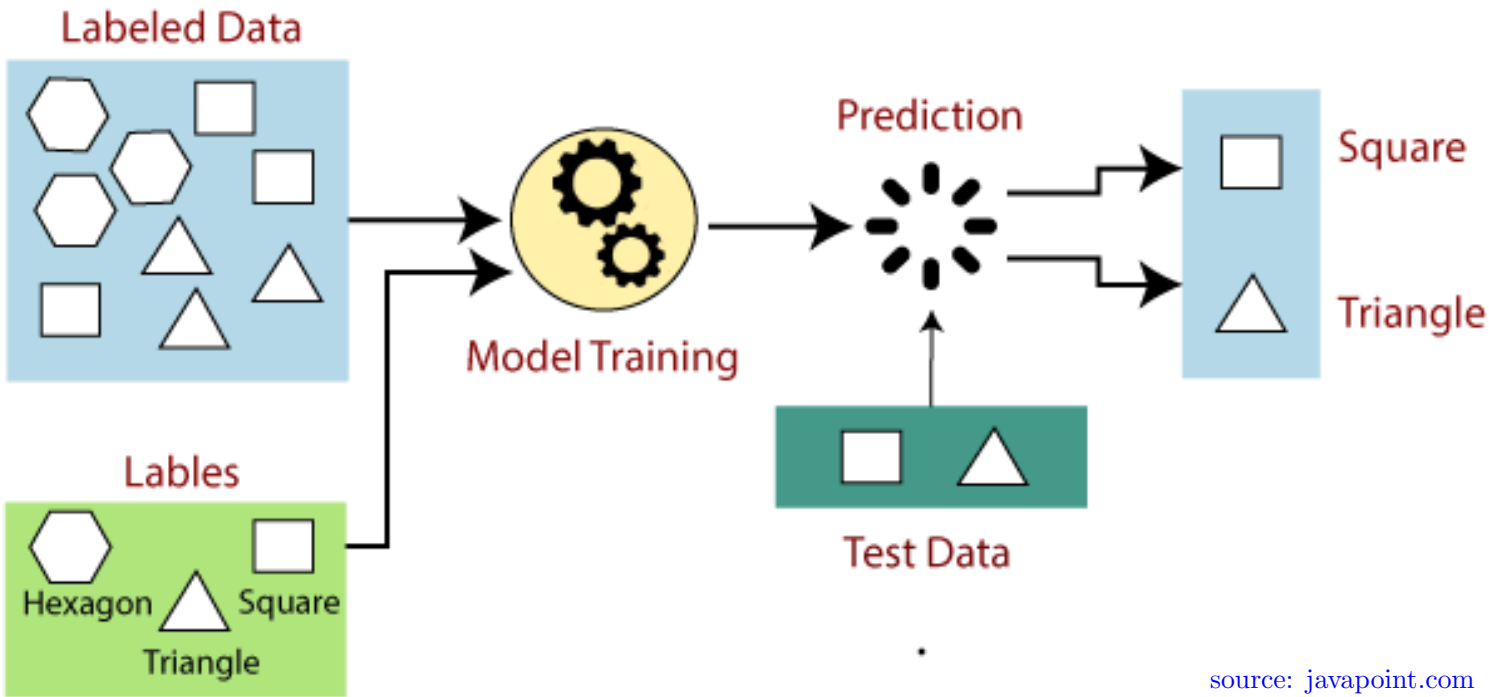
Classification

Input: A collection of objects

▷ feature vectors with class labels

Output: A model for the class attribute as a function of other attributes

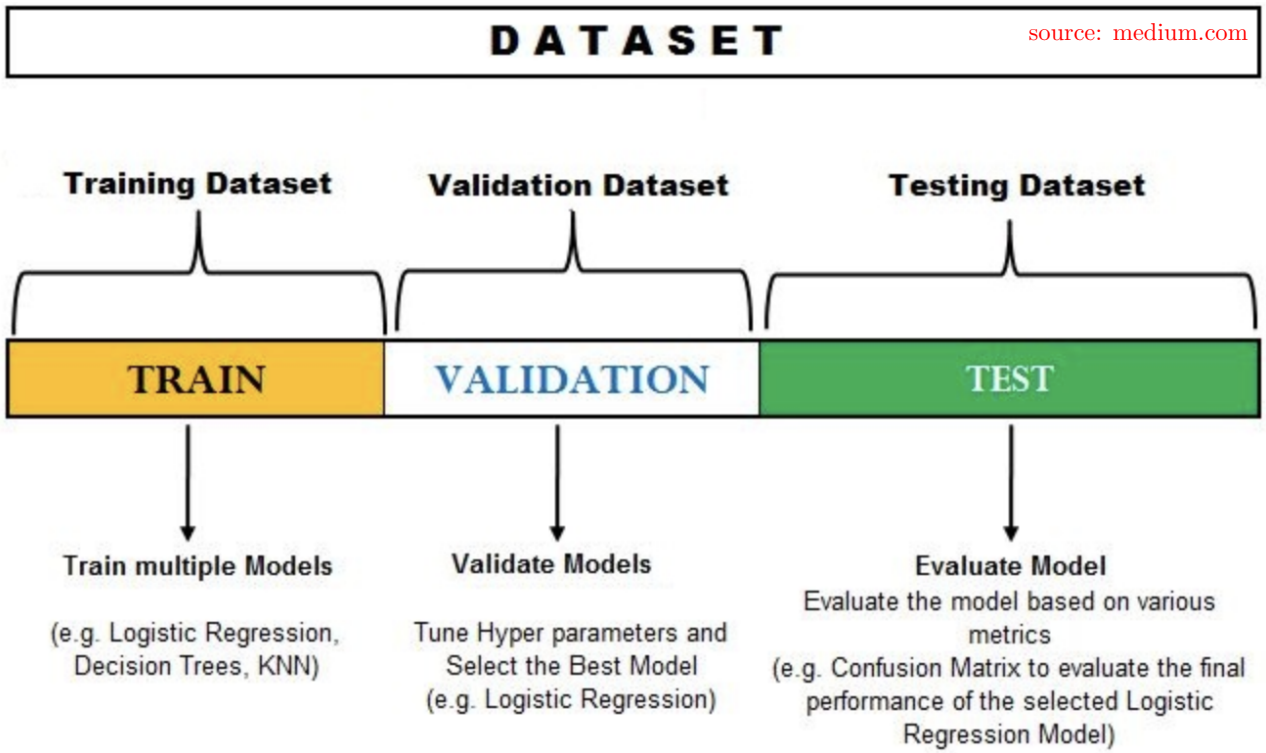




source: javapoint.com

Classification: Training-Validation split

- Generally obtained by randomly splitting the dataset
- e.g. $70-30$, $80-20$ random Train-Validation split
- Use average performance of multiple random (splits)

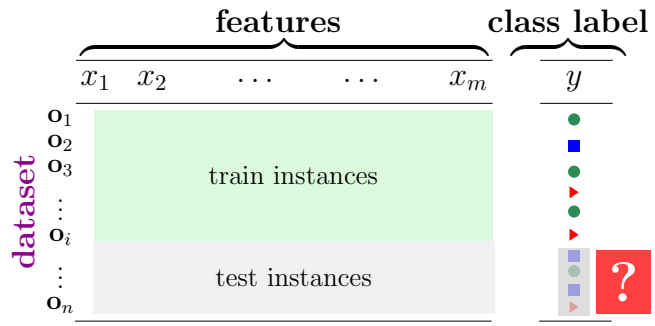


Classification

The model (binary classifier) is learned by finding patterns in training set
A classifier h approximates $y(\mathbf{o})$

h takes an object $\mathbf{o}$ and outputs the class label $\hat{y}(\mathbf{o})$

(aka model, hypothesis, discriminator, ...)



A **validation set** (a *representative* subset of training set) is used to learn parameters and tune architecture of classifier and estimate error

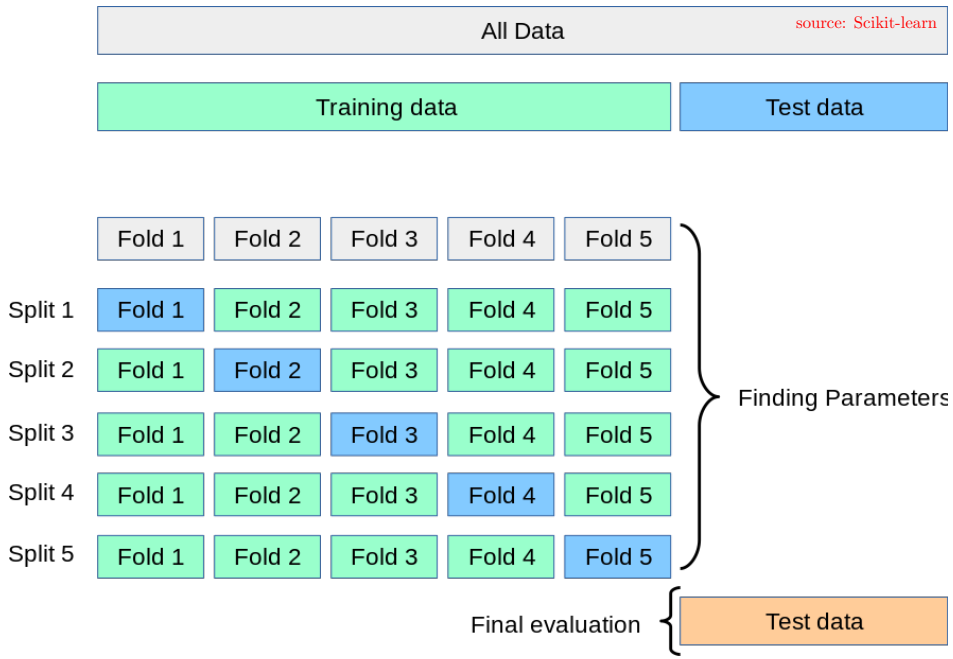
$$\text{MSE}(h) = \mathbb{E}\Big[\sum_{\mathbf{o}\in \text{ data set}} (y(\mathbf{o}) - h(\mathbf{o}))^2\Big]$$

Test set, not used for training, provides an estimate of generalization error

$h(\mathbf{o})$ can be real number (regression), labels of binary/multi classes, or class membership probability

Classification: Cross-Validation

- The dataset is randomly split into k folds
- In each of k the ith fold is used for validation and the rest for training
- Every instance is used once for validation and $k-1$ times for training
- k is usually 5 or 10



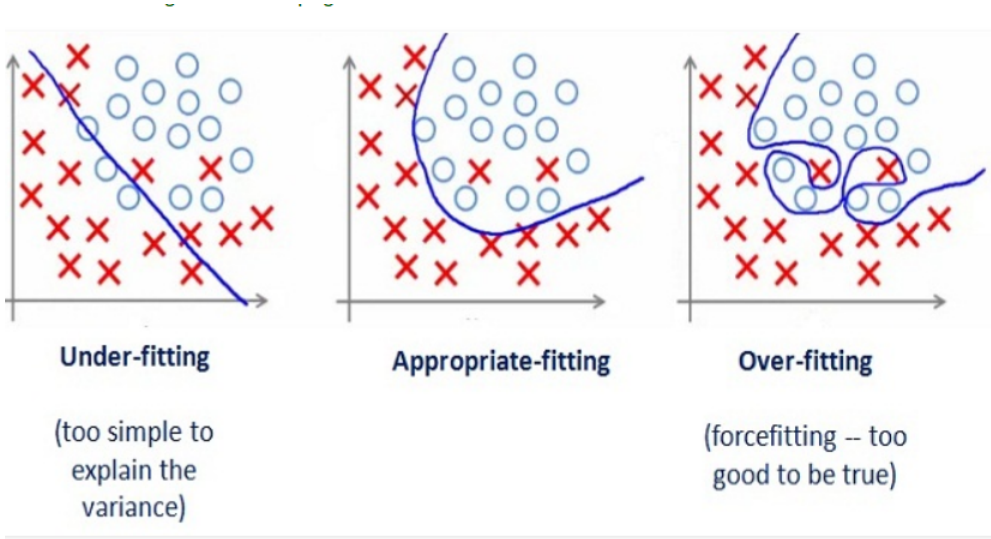
Classification: Overfitting

Overfitting: The phenomenon when model performs very well on training data but does not generalize to testing data

- The model learns the data and not the underlying function

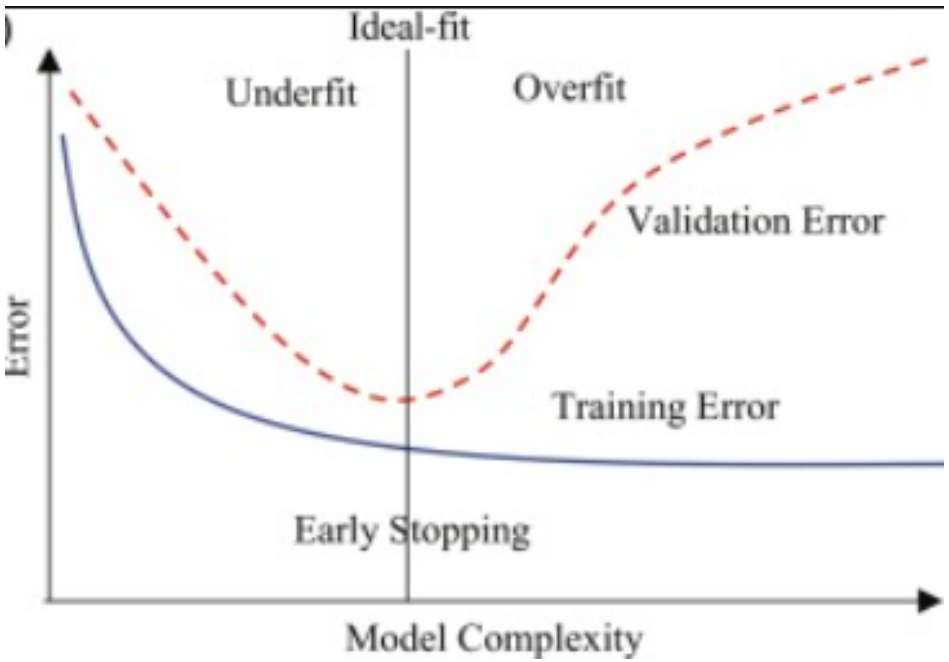
  ▷ Essentially learning by-rote

- Model has too much freedom (many parameters with wider ranges)



- Validation, Cross-validation, early stopping, regularization, model comparison, Bayesian priors help avoiding overfitting

Classification: Overfitting

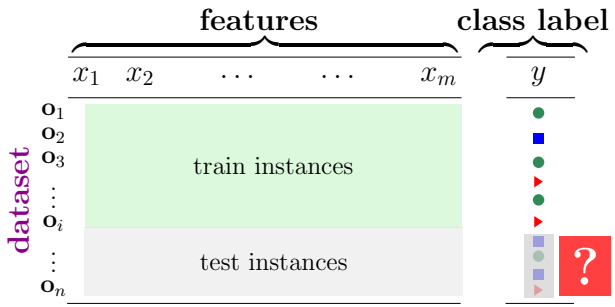


Variance and Bias in Machine Learning Models

A model h approximates $y(\cdot)$ using
$D = \{(\mathbf{o_1}, y_1), \ldots, (\mathbf{o_n}, y_n)\}$

For any unseen object $\mathbf{o}$, we want

$h(\mathbf{o}; D) = \hat{y}(\mathbf{o}) = \hat{y}$ to be equal to $y(\mathbf{o}) = y$



Bias: Error due to erroneous assumptions in the learning algorithm

$$\text{Bias}(\hat{y}) \;=\; \mathbb{E}_D[\hat{y}] - y$$

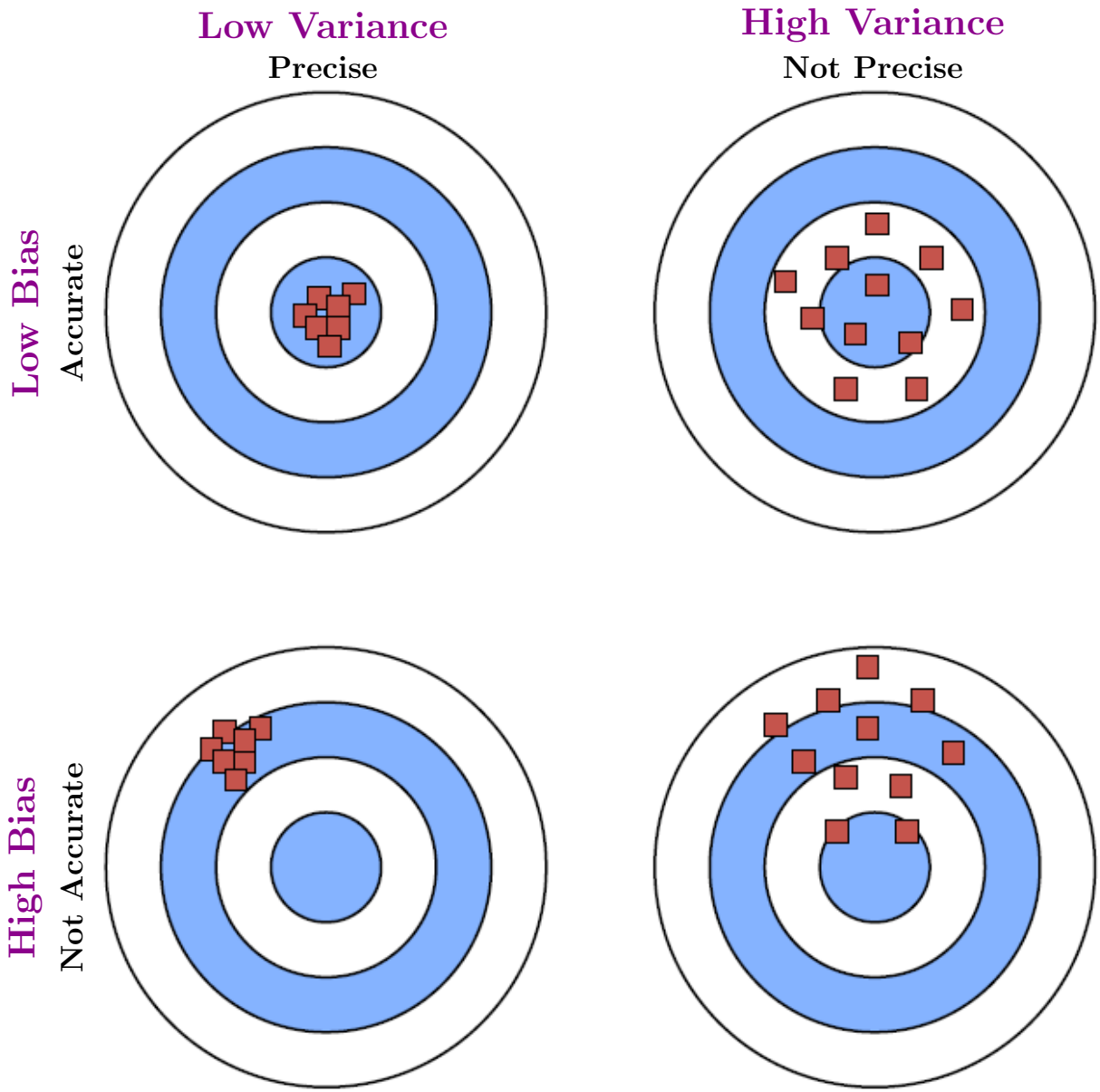
▷ High Bias: Model is oversimplified; misses the relevant relations between features and target outputs (underfitting)

Variance: Error due to sensitivity to fluctuations in training set

$$\text{Var}(\hat{y}) \;=\; \mathbb{E}_D\big[\,(\hat{y} - \mathbb{E}_D[\hat{y}])^2\,\big]$$

▷ High Variance: Model captures noise rather than output (overfitting)

Variance and Bias in Machine Learning Models



Variance and Bias in Machine Learning Models

$$\begin{aligned}\text{MSE}(h) =& \mathbb{E}[(y-\hat{y})^2] \\ =& \mathbb{E}[(y-\mathbb{E}[\hat{y}]+\mathbb{E}[\hat{y}]+\hat{y})^2] \\ =& \mathbb{E}[(y-\mathbb{E}[\hat{y}])^2]+2\mathbb{E}[(y-\mathbb{E}[\hat{y}])(\mathbb{E}[\hat{y}]-\hat{y})]+\mathbb{E}[(\mathbb{E}[\hat{y}]-\hat{y})^2]\end{aligned}$$

$$\begin{aligned}\mathbb{E}[(y-\mathbb{E}[\hat{y}])^2] =& \mathbb{E}[y^2-2y\mathbb{E}[\hat{y}]+\mathbb{E}[\hat{y}]^2] \\ =& \mathbb{E}[y^2]-2\mathbb{E}[y\mathbb{E}[\hat{y}]]+\mathbb{E}[\mathbb{E}[\hat{y}]^2] \\ =& y^2-2y\mathbb{E}[\hat{y}]+\mathbb{E}[\hat{y}]^2=(y-\mathbb{E}[\hat{y}])^2 \;=\; Bias(\hat{y})^2\end{aligned}$$

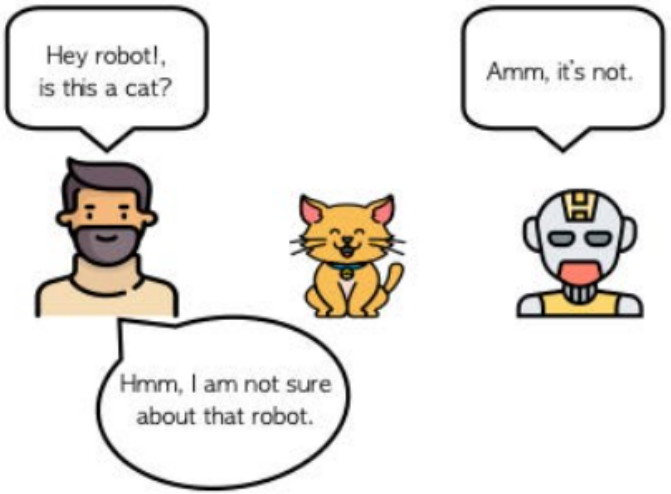
$$\begin{aligned}\mathbb{E}[(y-\mathbb{E}[\hat{y}])(\mathbb{E}[\hat{y}]-\hat{y})] &= \mathbb{E}[y\mathbb{E}[\hat{y}]-\mathbb{E}[\hat{y}]^2-y\hat{y}+\mathbb{E}[\hat{y}]\hat{y}] \\ &= y\mathbb{E}[\hat{y}]-\mathbb{E}[\hat{y}]^2-y\mathbb{E}[\hat{y}]+\mathbb{E}[\hat{y}]^2=\mathbf{0}\end{aligned}$$

$$\mathbb{E}[(\mathbb{E}[\hat{y}]-\hat{y})^2] = Var(\hat{y})$$

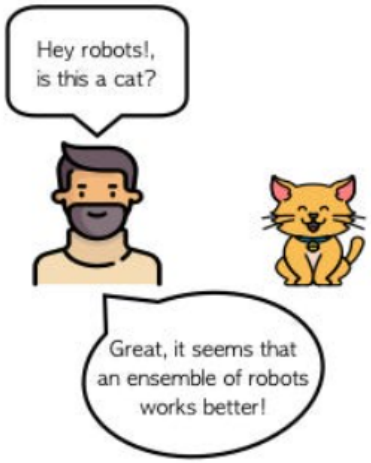
$$\text{Total Error} = \text{MSE}(\hat{y}) \;=\; \text{Bias}(\hat{y})^2 + \text{Var}(\hat{y}) + \text{Noise}$$

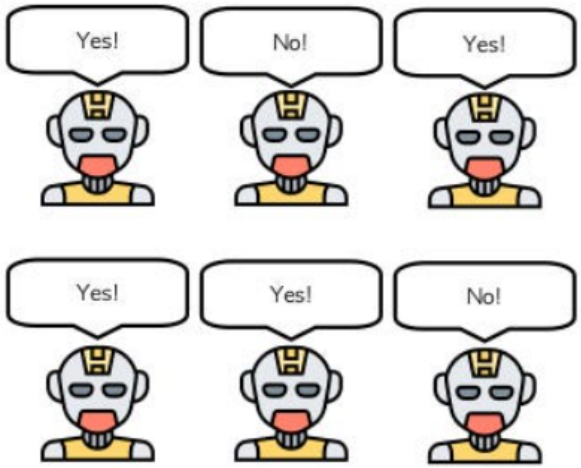
Ensemble Learning

Ensemble Learning



a) Traditional Learning



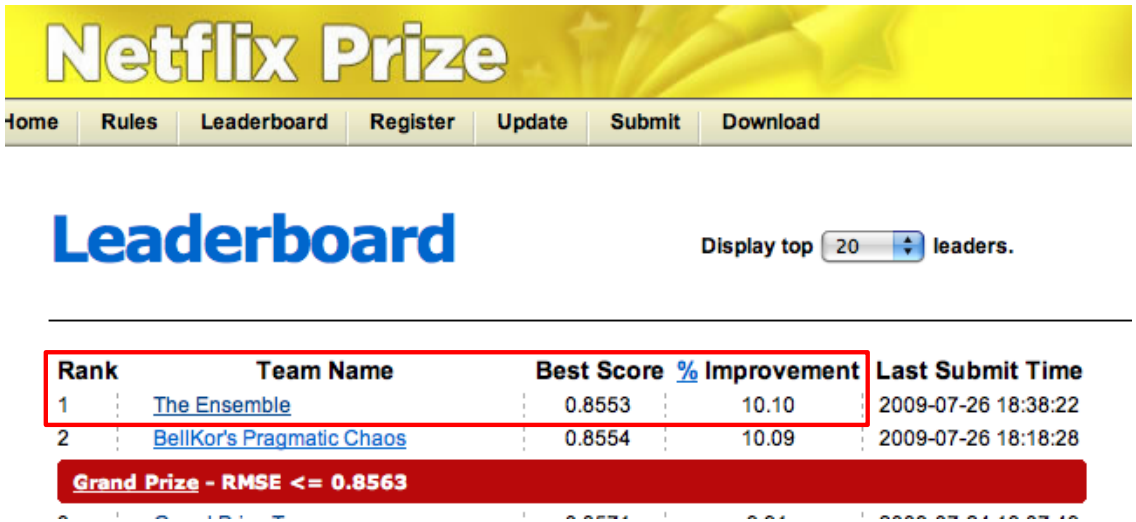


b) Ensemble Learning

source: Fernando López (towardsdatascience.com)

Ensemble Learning

Ensemble learning combines several machine learning techniques into one predictive model to decrease variance (bagging), bias (boosting), or improve predictions (stacking)



Ensemble methods

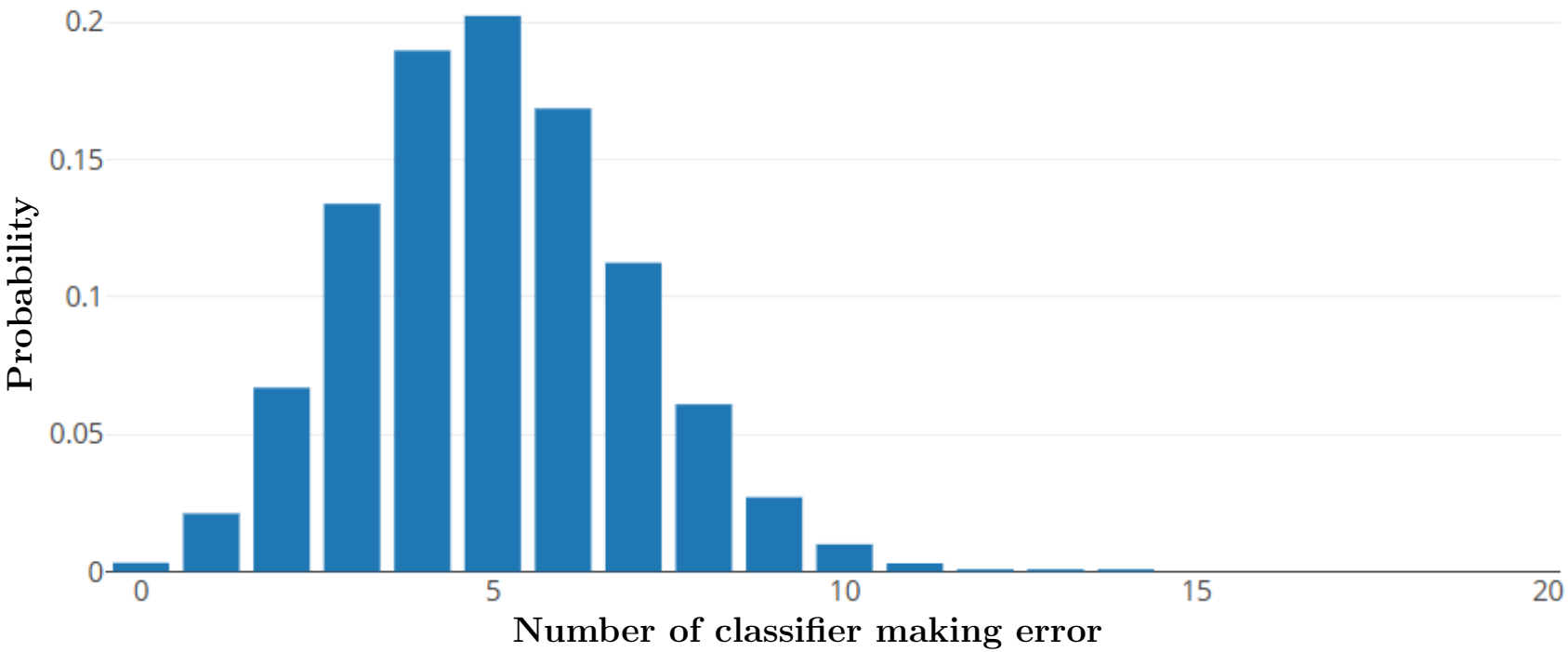
- are used when single model cannot achieve high accuracy
- typically generalize well on unseen data
- have good theoretical guarantees
- are easy to implement

Ensemble Learning: Intuition

Ensemble learning combines several machine learning techniques into one predictive model

The magic of Independent Trials: Suppose we have k classifiers with probability of error, $p_1, p_2, \ldots, p_k$

Let $p = \max_i\{p_i\}$, suppose $p = 0.25$

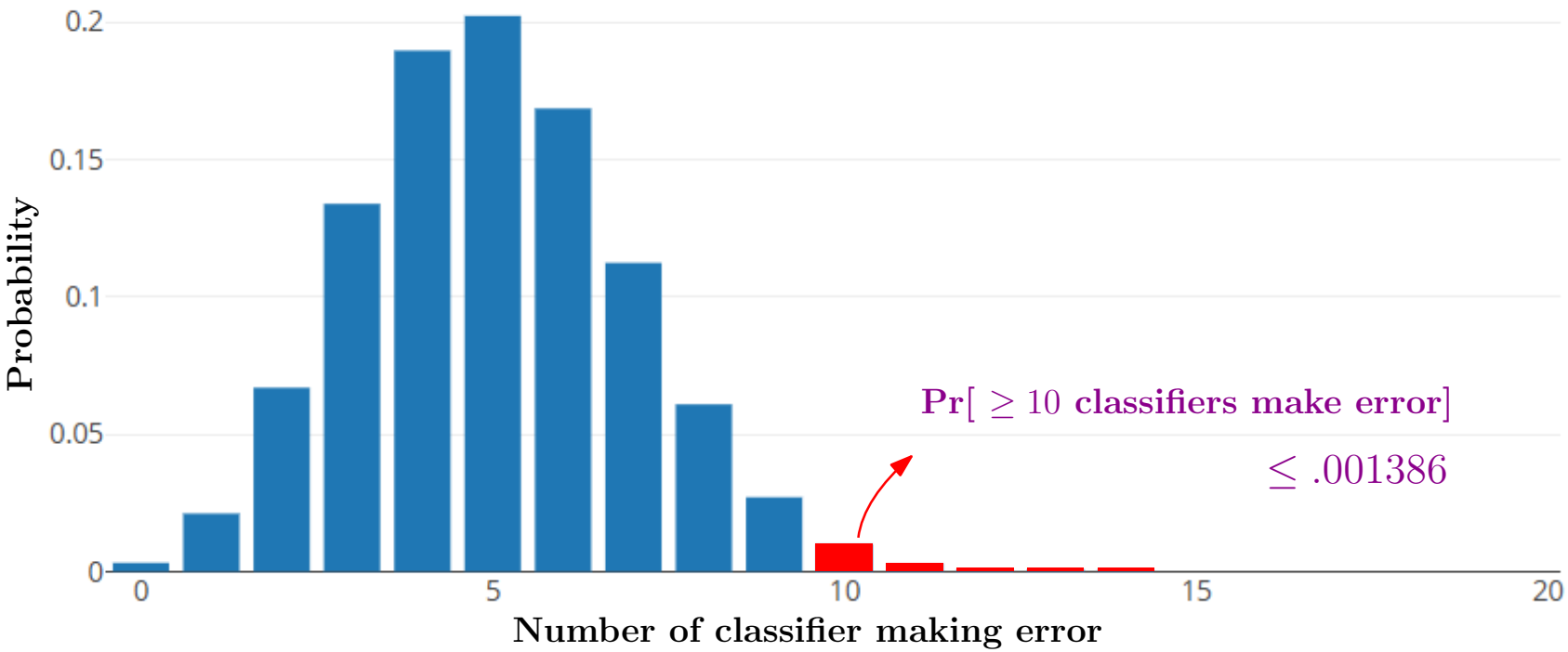


Ensemble Learning: Intuition

Ensemble learning combines several machine learning techniques into one predictive model

The magic of Independent Trials: Suppose we have k classifiers with probability of error, $p_1, p_2, \ldots, p_k$

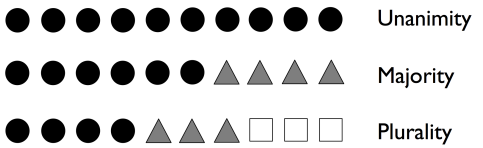
Let $p = \max_i\{p_i\}$, suppose $p = 0.25$

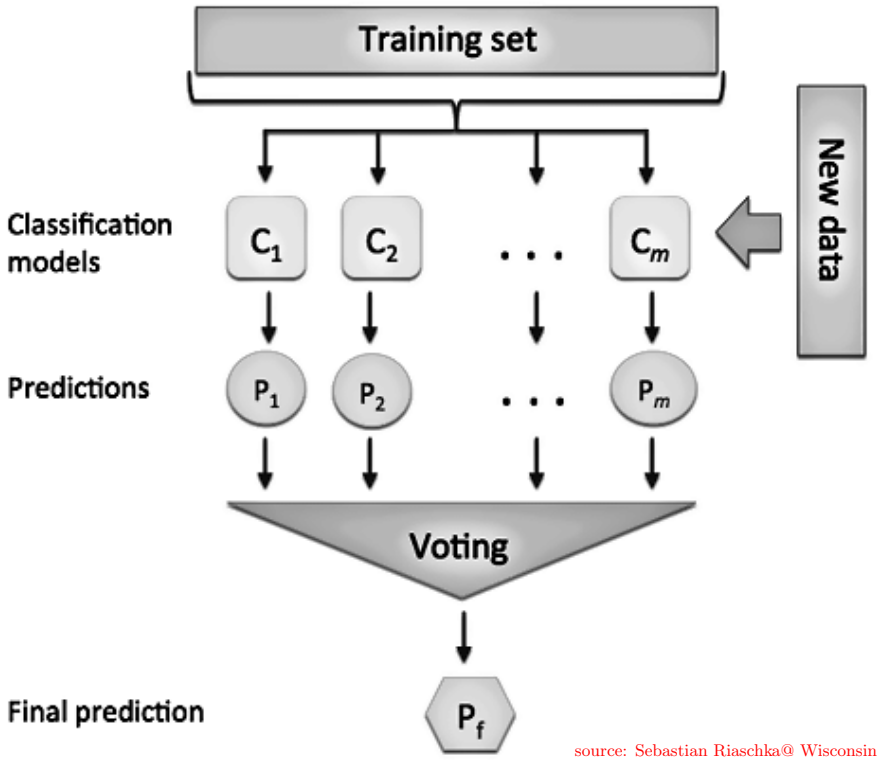


Majority Voting and Weighted Aggregation

Majority voting is a simple ensemble technique

- Votes: The predicted class of each sub-model
- Ensemble prediction: The class with **majority** votes



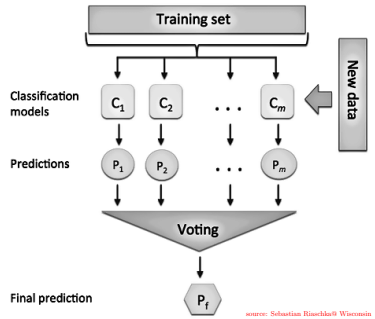


source: Sebastian Riaschka@ Wisconsin

Majority Voting and Weighted Aggregation

Majority voting is a simple ensemble technique

- Votes: The predicted class of each sub-model
- Ensemble prediction: The class with **majority** votes



Suppose we have n classifiers with probability of error, $p_1, p_2, \ldots, p_n$

Let $0.5 > p = \max_i\{p_i\}$ ▷ p : base probability, we need $p < 0.5$

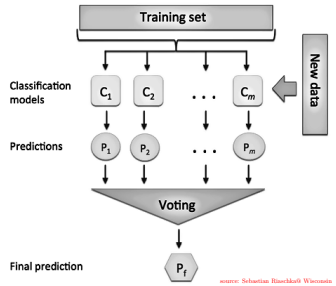
$$\text{Ensemble error: } p_{ens} < \sum_{i>\frac{n}{2}} \binom{n}{i} p^i (1-p)^{n-i}$$

$\downarrow n \quad p \rightarrow$	0.25	0.49	0.75
11	0.03433	0.47295	0.96567
21	0.00642	0.29888	0.97937
31	0.0013	0.45531	0.9987

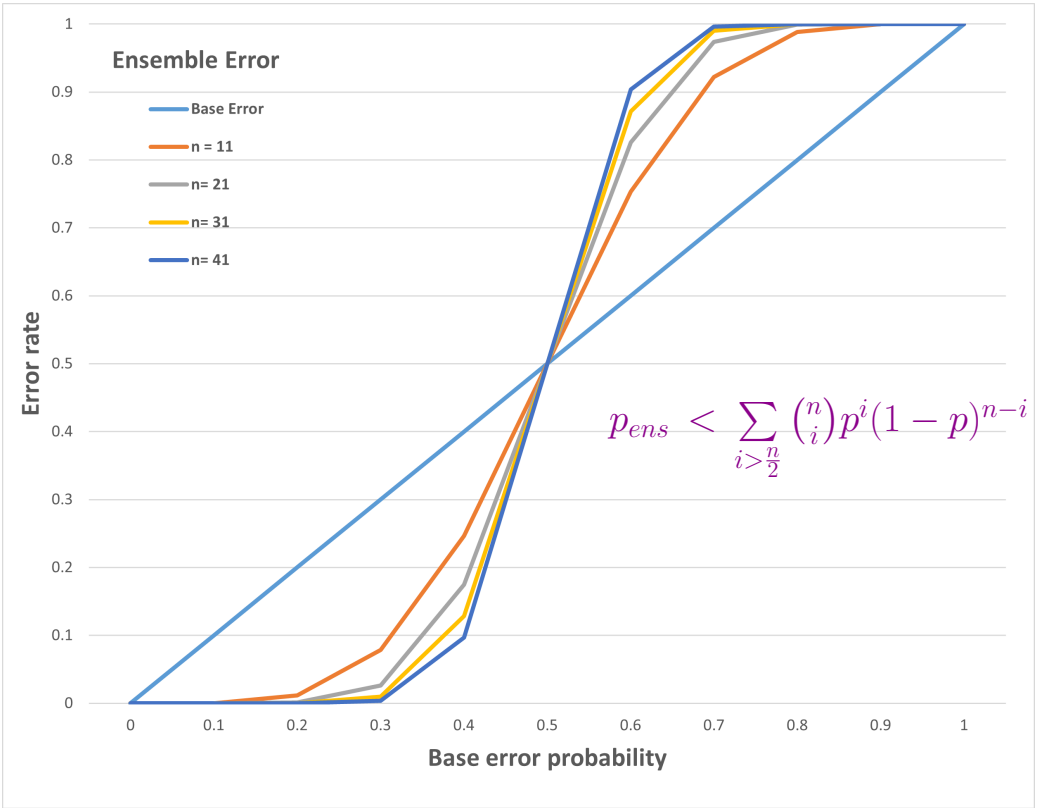
Majority Voting and Weighted Aggregation

Majority voting is a simple ensemble technique

- Votes: The predicted class of each sub-model
- Ensemble prediction: The class with **majority** votes



n classifiers, error rates $p_1, \ldots, p_n$ Base probability $p = \max_i\{p_i\}$



Soft Voting – Weighted Aggregation

Soft voting ensembles classifiers that predict class membership probabilities

- Votes: $p_{i,j}$ predicted class membership probability for class j of sub-model i
- Weights: w_i weight/trust/reputation of sub-model i

  ▷ default $w_i = \frac{1}{n}$, $\forall i \in \{w_1, \ldots, w_n\}$
- Ensemble prediction: The class j such that

$$\arg\max_j \sum_{i=1}^{n} w_i p_{i,j} = \hat{y}_f$$

Soft Voting – Weighted Aggregation

Soft voting ensembles classifiers that predict class membership probabilities

- Votes: $p_{i,j}$ predicted class membership probability for class j of sub-model i
- Weights: w_i weight/trust/reputation of sub-model i
- Ensemble prediction: The class j such that

$$\arg\max_j \sum_{i=1}^{n} w_i p_{i,j} = \hat{y}_f$$

Binary classification: *class* $j \in \{0, 1\}$ and $h_i \quad (i \in \{1, 2, 3, 4\})$:

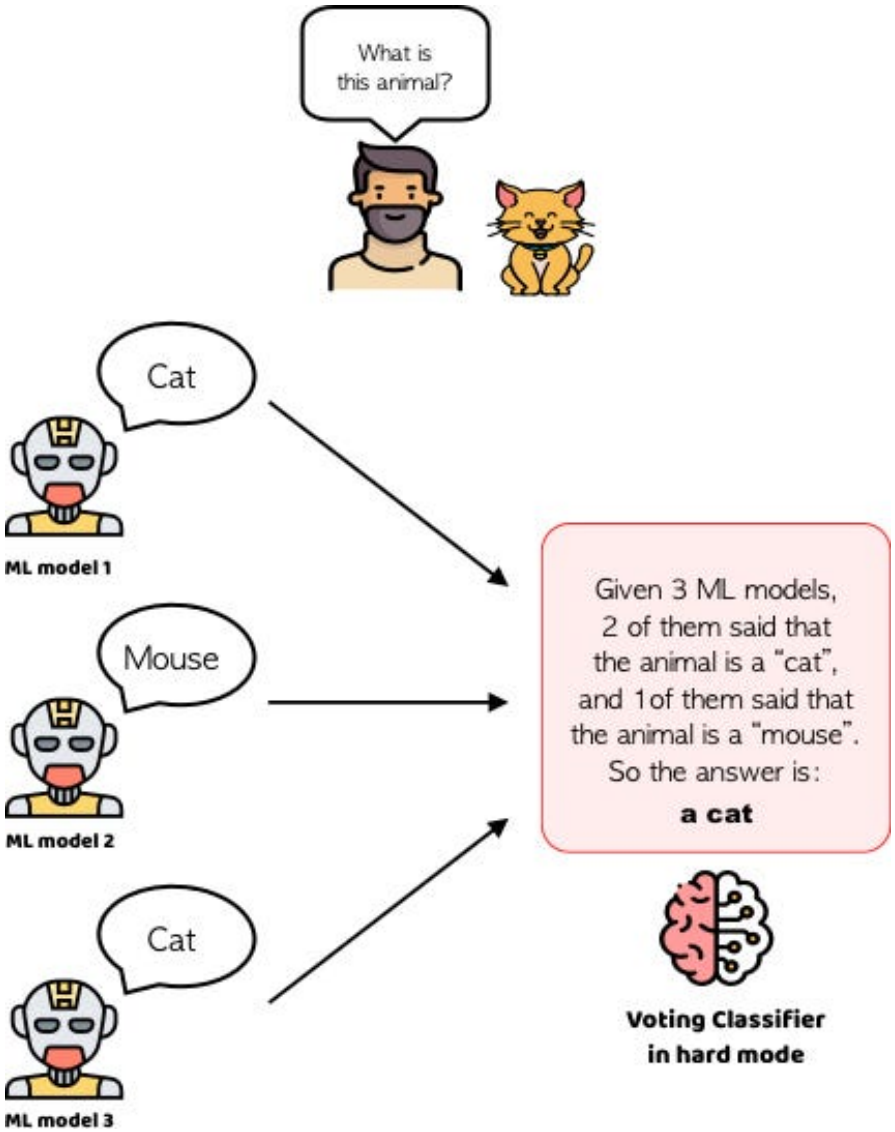
- $h_1(x) \rightarrow [0.67, 0.33]$
- $h_2(x) \rightarrow [0.3, 0.7]$
- $h_3(x) \rightarrow [0.35, 0.65]$
- $h_4(x) \rightarrow [0.4, 0.6]$

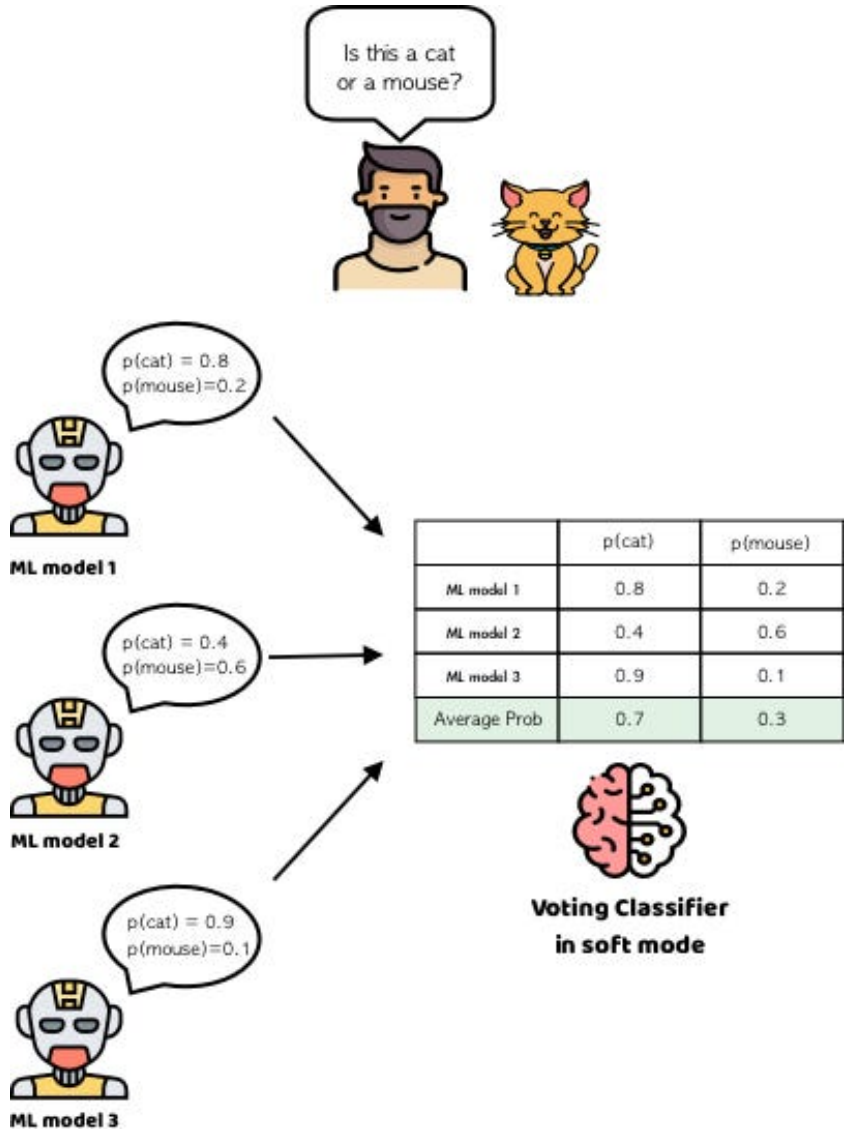
$$p(j = 0|\mathbf{o}) = 1/4(0.8 + 0.3 + 0.35 + 0.4) = 0.43$$

$$p(j = 1|\mathbf{o}) = 1/4(0.2 + 0.7 + 0.65 + 0.6) = 0.57$$

$$\hat{y}_f(\mathbf{o}) = \arg\max_j \{p(j = 0|\mathbf{o}), p(j = 1|\mathbf{o})\}$$

Ensemble Learning: Hard and Soft Voting



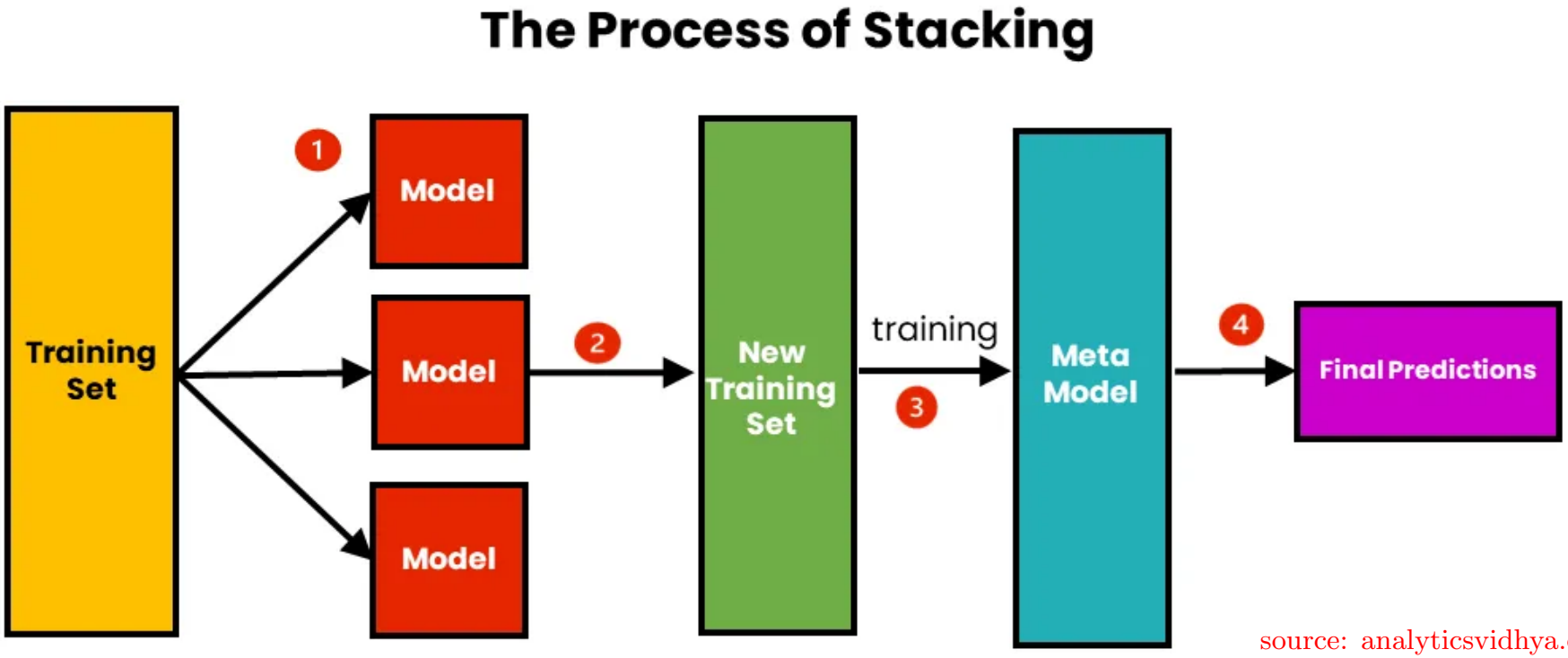


source: Fernando López (towardsdatascience.com)

Stacking

Stacking trains a new model using predictions of base classifiers as input

Base models (e.g., logistic regression, decision tree) predict on input data
Meta-model (e.g., neural network) uses the predictions for final prediction



source: analyticsvidhya.com

Stacking

Stacking trains a new model using predictions of base classifiers as input

Base models (e.g., logistic regression, decision tree) predict on input data
Meta-model (e.g., neural network) uses the predictions for final prediction

Algorithm 4 Stacking

Input: Training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathcal{Y}$
Output An ensemble classifier H

for $t = 1$ to T **do** ▷ T first-level classifiers
    Learn a base classifier h_t based on $\mathcal{D}$

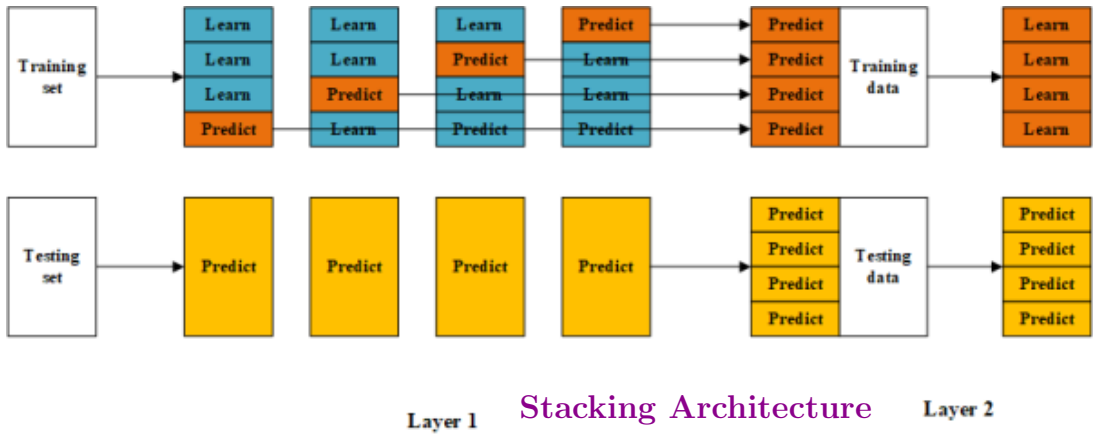
for $i = 1$ to m **do** ▷ Construct new data set from $\mathcal{D}$
    Construct a new data set $\{\mathbf{x}'_i, y_i\}$, $\mathbf{x}'_i = \{h_1(x_i), h_2(x_i), \ldots, h_T(x_i)\}$

Learn a new classifier h' based on the newly constructed data set
Return $H(x) = h'(h_1(x), h_2(x), \ldots, h_T(x))$

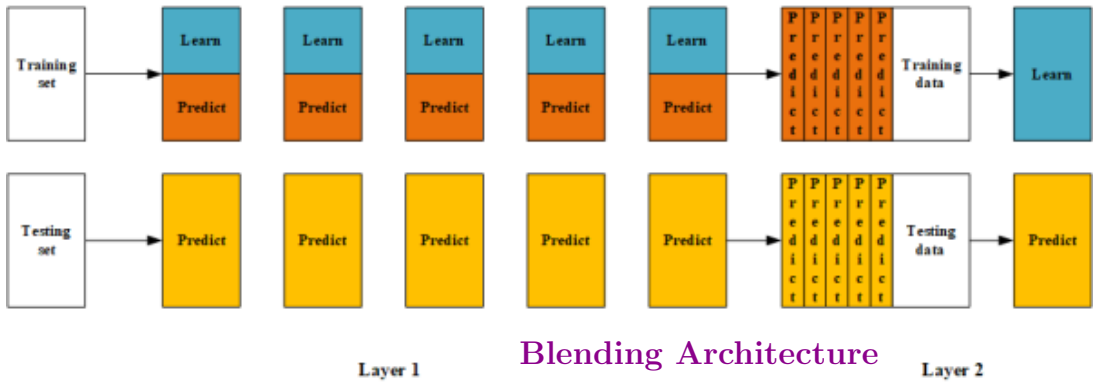
Blending

Blending is similar to stacking but uses a holdout set from the training dataset to train the combiner model (the meta-model)

E.g., base models are trained on 75% of data, and predictions are made on the remaining 25%. These predictions are used as features for meta-model



source: https://www.codeproject.com/Tips/4354591/Step-by-Step-Guide-to-Implement-Machine-Learning-V



Bagging

Bagging, or Bootstrap Aggregating, reduces variance by training multiple models on different subsets of the dataset and averaging their predictions.

Bagging helps reduce variance and helps to avoid overfitting

Algorithm 5 Bagging (n) bootstrap samples

Input: Training data $X = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathcal{Y}$

Output An ensemble classifier H

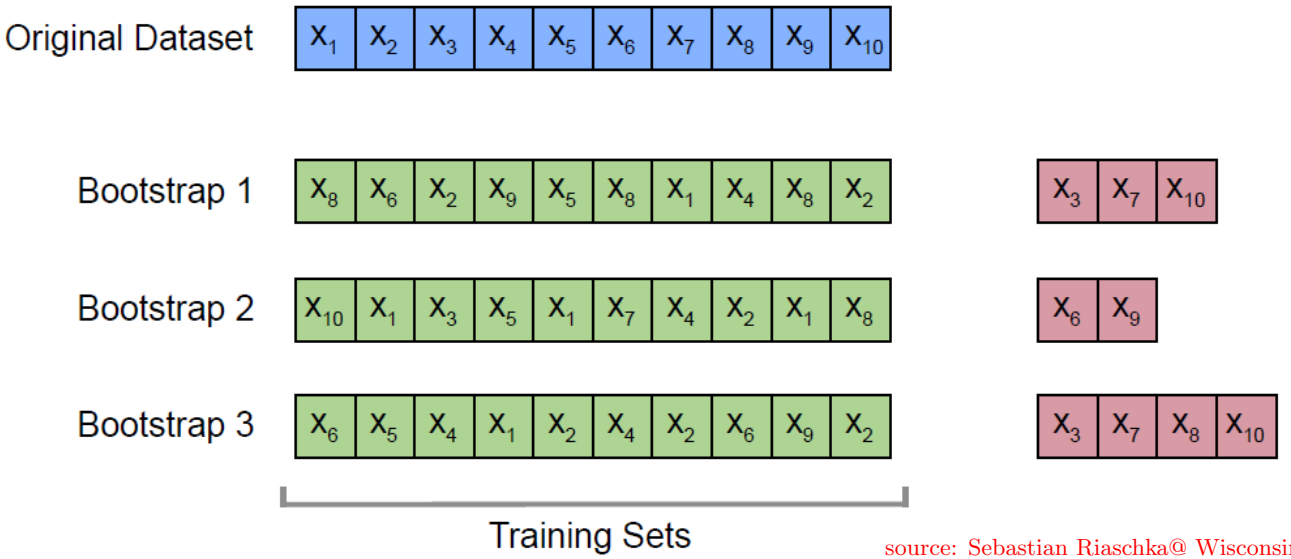
for $i = 1 \rightarrow n$ **do**

$X_i \leftarrow$ bootstrap sample of size m from training dataset X ▷ i.i.d sampling with replacement from X

Train classifier h_i on X_i ▷ sub-model

return $\hat{y}_f(\mathbf{x}) \leftarrow \textbf{mode}\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_n(\mathbf{x})\}$ ▷ $\mathbf{x}$ is test instance

Bagging: Bootstrap Sampling

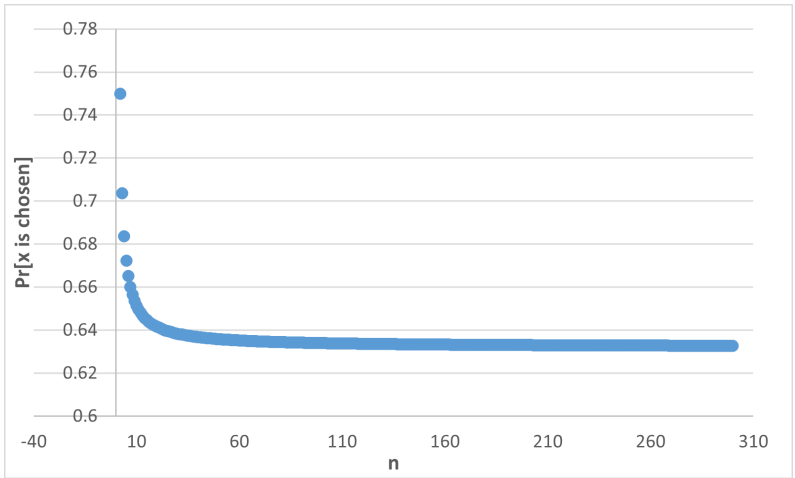


source: Sebastian Riaschka@ Wisconsin

$$Pr[\mathbf{x}_i \text{ is not chosen}] = (1 - 1/n)^n$$

$$\lim_{n \to \infty} (1 - 1/n)^n = 1/e$$

$$Pr[\mathbf{x}_i \text{ is chosen}] = 1 - (1 - 1/n)^n$$
$$\simeq 06327$$



Bagging Workflow

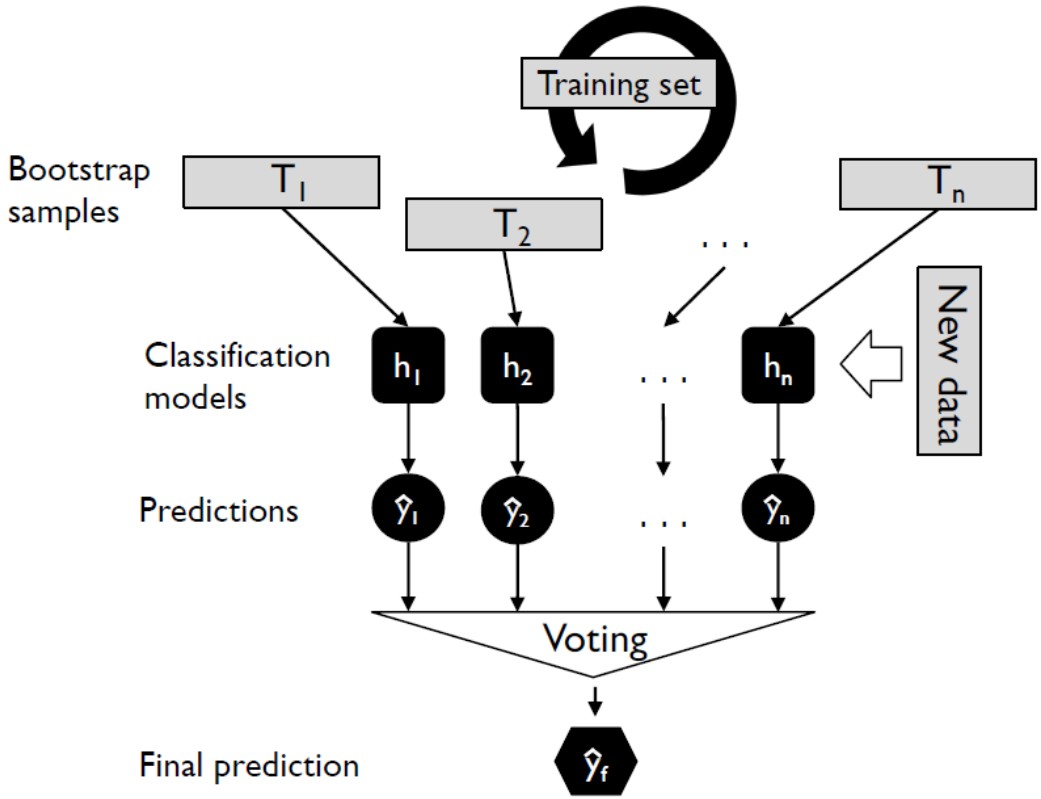
Training example indices	Bagging round 1	Bagging round 2	…
1	2	7	…
2	2	3	…
3	1	2	…
4	3	1	…
5	7	1	…
6	2	7	…
7	4	7	…

h_1 h_2 h_n

$$\hat{y}_f = \mathbf{mode}\left\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_n(\mathbf{x})\right\}$$

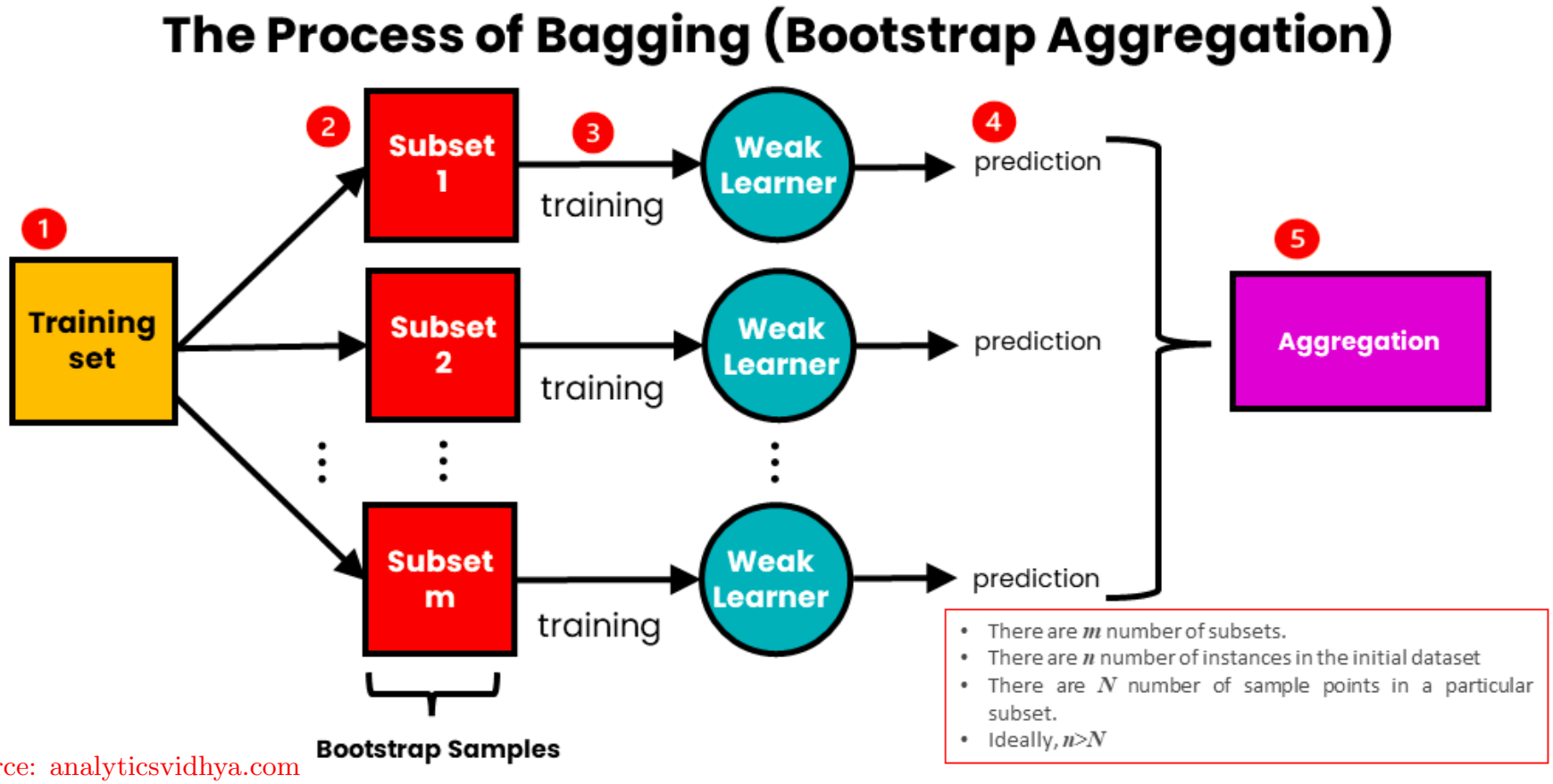
where $h_i(\mathbf{x}) = \hat{y}_i(\mathbf{x})$

Diagram adapted from Sebastian Riaschka@ Wisconsin



Benefits and Limitations of Bagging

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we also reduce Bias? – Yes **Boosting**



source: analyticsvidhya.com

Boosting

Boosting is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers

PAC Learning model (Mathematical Analysis framework for ML)

get random examples from unknown, arbitrary distribution

Strong PAC Learning Algorithm: if for any distribution D, parameters ϵ, δ, using polynomially many examples and polynomial time, we can find classifier h such that $Pr[error_D(h) \leq \epsilon] \geq 1 - \delta$

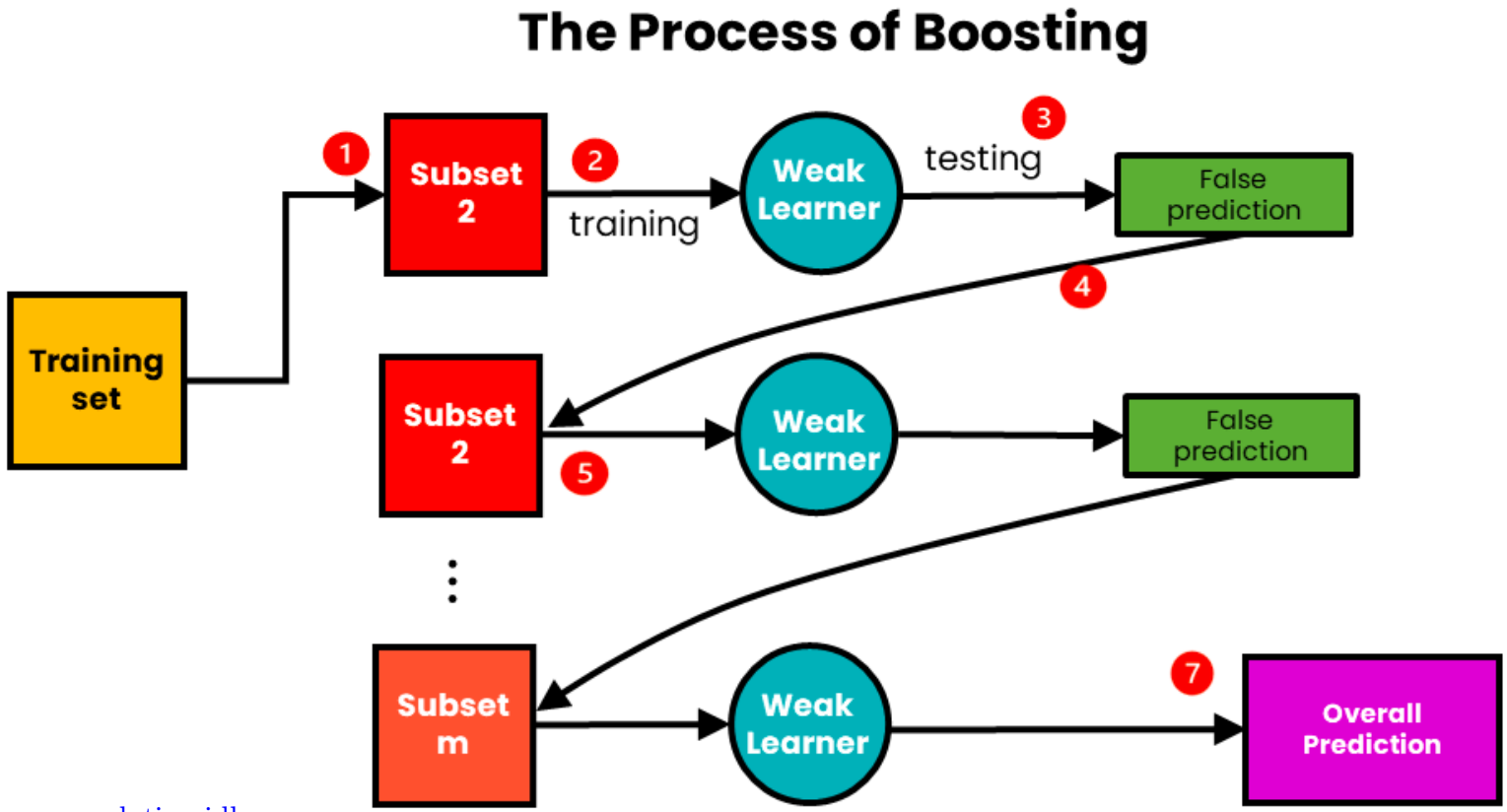
Weak PAC Learning Algorithm: Same as above, but generalization error only has to be slightly better than random guessing, i.e. $\exists\gamma$ such that $Pr[error_D(h) \leq \frac{1}{2} - \gamma] \geq 1 - \delta$

[Kearns & Valiant (1988)] Does weak learnability imply strong learnability?

Yes, boost by majority voting Freund (1990)

Boosting Workflow

Boosting reduces the overall bias by enhancing weak learners through focusing on samples that previous models misclassified



source: analyticsvidhya.com

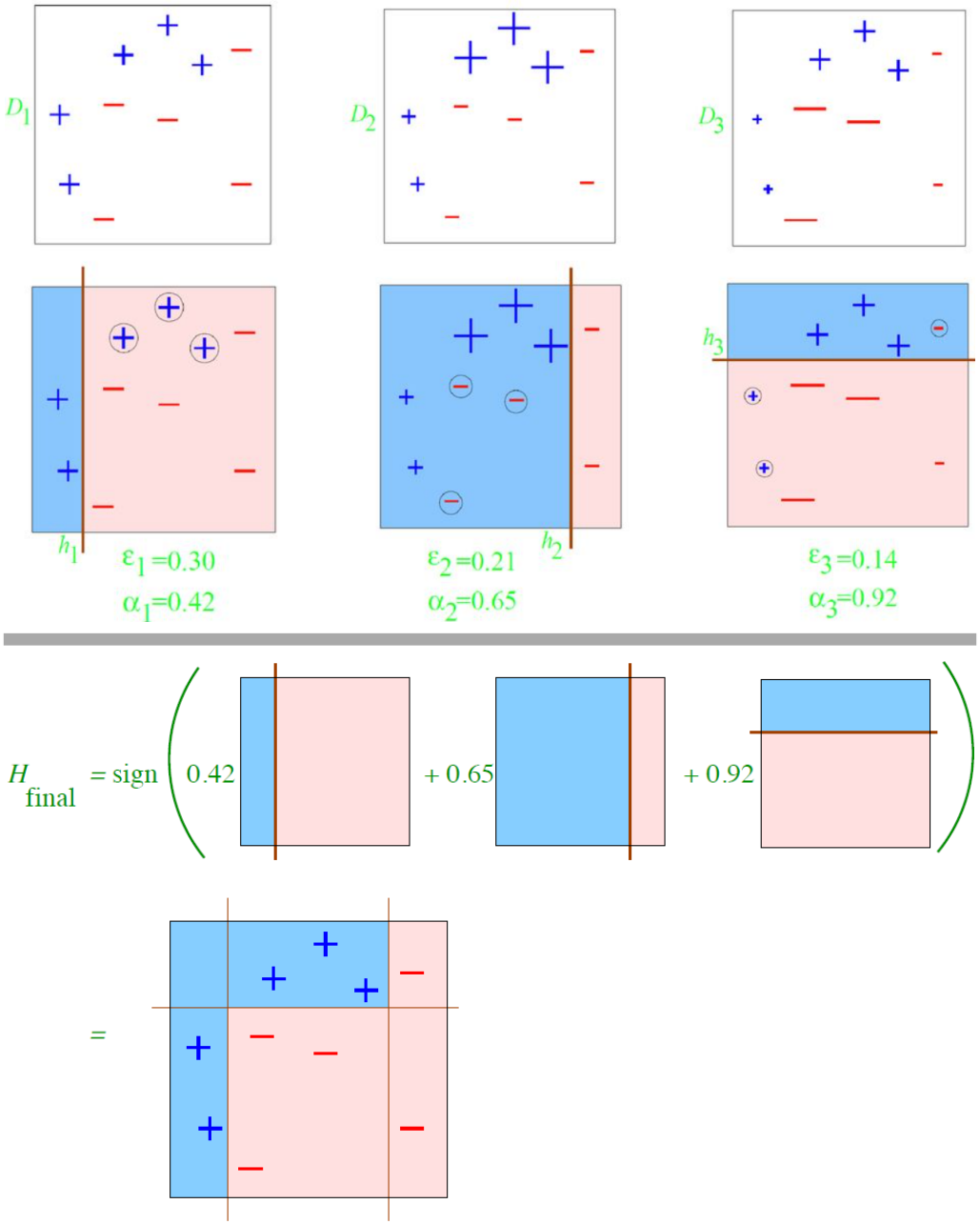
Popular Algorithms: 1) AdaBoost 2) Gradient Boosting

Boosting

1. Initialization:
   - Given Training data $X = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$
   - Initialize a distribution $D_1(i) = 1/m$ for $i = 1, \ldots, m$
2. **For** $t = 1$ to T:
   - Training:
     - Train weak classifier $h_t : X \to \{-1, +1\}$ under the distribution D_t
     - $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$ e.g., $\epsilon_t = \sum_{i=1}^{m} D_t(i) \cdot \mathbf{1}[h_t(x_i) \neq y_i]$ ▷ error of h_t
     - If $\epsilon_t \geq 1/2$, terminate or adjust the algorithm
   - Update Distribution:
     - Compute $\alpha_t = 1/2 \log\left(1-\epsilon_t/\epsilon_t\right)$ ▷ measure performance of h_t
     - $D_{t+1}(i) = D_t(i)e^{-\alpha_t y_i h_t(x_i)}/Z_t$ ▷ Updated distribution

       Z_t is a normalization factor to ensures D_{t+1} is a probability distribution
3. Final Classifier:
   - $H_f(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$ ▷ weighted majority of T weak classifiers

Boosting


D_1
D_2
D_3
h_1
h_2
h_3
$\varepsilon_1 = 0.30$
$\alpha_1 = 0.42$
$\varepsilon_2 = 0.21$
$\alpha_2 = 0.65$
$\varepsilon_3 = 0.14$
$\alpha_3 = 0.92$
$H_{\text{final}} = \text{sign}$
0.42
+ 0.65
+ 0.92
=

AdaBoost for Rating Aggregation

- AdaBoost can be adapted to aggregate product ratings
- Multiple models are trained each providing an aggregate rating, focusing on different aspects of the product and reviews - e.g.,
  - A model that predicts product ratings based on user demographics (e.g., their history of ratings)
  - A model that predicts ratings based on product features (e.g., price, category)
  - A model that predicts ratings based on the sentiment of reviews (positive or negative sentiment)
- Final aggregated rating is then a weighted average of the submodels

This approach can be employed when we have a good number of products for which the ground truth is available, i.e. we know the true rating or score of the products

AdaBoost for Rating Aggregation

Estimator	p_1	p_2	p_3	$\cdots$
Estimator 1	$y_1(p_1)$	$y_1(p_2)$	$y_1(p_3)$	$\cdots$
Estimator 2	$y_2(p_1)$	$y_2(p_2)$	$y_2(p_3)$	$\cdots$
Estimator 3	$y_3(p_1)$	$y_3(p_2)$	$y_3(p_3)$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Correct Answer	t_1	t_2	t_3	$\cdots$

- Train n estimators on m products
- $y_i(p_j)$ is the prediction from the i-th estimator
- α_i: weight assigned to estimator i (based on performance)
- Final aggregate estimate: $y(p_j) = \sum_{i=1}^{n} \alpha_i y_i(p_j)$
- We train these estimator using boosting strategy

AdaBoost for Rating Aggregation

Adjusts Training weights: w_{ij} for the i-th estimator and j-th product to focus on products earlier models performed poorly

Initial weights, $w_{ij} = \frac{1}{m}$

Then sequentially for each i, we train estimator y_i to minimize $\sum_{j=1}^{m} w_{ij} \mathbf{1}[y_i(p_j) \neq t_j]$

Error Calculation for model i: $\epsilon_i = \frac{\sum_j w_{ij} \mathbf{1}[y_i(p_j) \neq t_j]}{\sum_j w_{ij}}$

Estimator Combining Weight (importance of model i): $\alpha_i = \log\left({}^{1}/_{\epsilon_i} - 1\right)$

Update instances weights for next model: $w_{i+1,j} = w_{ij} \cdot e^{\alpha_i \mathbf{1}[y_i(x_j) \neq t_j]}$

Note that $w_{i+1,j} = \begin{cases} w_{ij} & \text{if model } i \text{ gets } p_j \text{ right} \\ w_i j \cdot \left({}^{1}/_{\epsilon_i} - 1\right) & \text{if model } i \text{ makes a mistake on } p_j \end{cases}$

Time-Sensitive Rating Aggregation

Time-Sensitive Rating Aggregation

Time-sensitive rating aggregation accounts for changes over time

Older reviews may be less relevant to current product quality, while recent reviews might provide more accurate assessments

- Review Decay: Older reviews may be down-weighted to prioritize newer feedback
- Trend Identification: (e.g., improving or declining quality) based on the temporal distribution of reviews
- Modeling Time: Algorithms incorporate timestamps into rating models to apply different weights to reviews based on recency

Time-Sensitive Rating Aggregation

Adjusting the weight of each review based on its timestamp

- ▷ The older the review, the less weight it may carry in the overall rating

Time Decay Function: Reviews weighted based on their recency

$$w(t) = e^{-\lambda t}$$

- t is the time since the review was posted
- λ is the decay rate that controls how fast the decay occurs

Time-Weighted Aggregation:

- Reviews $\{r_1, r_2, \ldots, r_n\}$, timestamps of reviews $\{t_1, t_2, \ldots, t_n\}$
- Decay rate λ

$$\hat{R}(t) = \frac{\sum_{i=1}^{n} w(t_i) \cdot r_i}{\sum_{i=1}^{n} w(t_i)}$$

Provides a evolving reflection of product or service quality

Seasonality in Rating Data

Many products and services experience periodic changes in popularity

Seasonality, holidays, and events can cause temporary spikes or drops in ratings that do not reflect long-term quality

- Fourier Analysis: Decomposes the rating data into sine and cosine components to detect underlying periodicity
- Autocorrelation Function (ACF): Measures the correlation of the ratings with their lagged values to identify repeating patterns
- Seasonal Decomposition of Time Series: Separates a time series into seasonal, trend, and residual components

  ▷ Seasonal and Trend decomposition using LOESS (Locally Weighted Regression and Scatterplot Smoothing)

Seasonality in Rating Data

Seasonal Adjustment: Adjust or compensate for seasonality/periodicity

Seasonal adjustment removes/normalizes periodic trends from rating data, leaving a "deseasonalized" time series that reflects long-term patterns

- **Seasonal Decomposition (STL):** Removes the seasonal component detected by STL to leave only the trend and residual components

$$R_{adj}(t) = R(t) - S(t)$$

  where $S(t)$ is the seasonal component of the time series
- **Ratio-to-Moving-Average Method:** Divides the rating at each time point by the corresponding seasonal factor, which is derived from a moving average of ratings over a fixed time period p

$$R_{adj}(t) = {}^{R(t)}/_{S(t)}$$

  - $S(t) = {}^{R(t)}/_{MA(t)}$ ▷ seasonal factors
  - $MA(t) = \sum_{i=t-p+1}^{t} {}^{R(i)}/_{p}$ ▷ Moving Average over period p

Aggregating Streaming Ratings

Aggregating streaming ratings involves handling a continuous flow of data from users. Ratings arrive over time and must be integrated into the aggregate score in real-time without reprocessing historical data

- Cumulative Moving Average: Incrementally updates the average rating as new ratings arrive

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} R_i$$

- Exponential Moving Average (EMA): Assigns exponentially decreasing weights to older ratings, giving more importance to recent data

$$\mu_t = \alpha R_t + (1-\alpha)\mu_{t-1}$$

- Window-Based Approaches: Aggregates ratings over a sliding window of recent data, providing real-time feedback using limiting memory

Identifying and Mitigating Bias in Ratings

Bias in ratings can distort the accuracy of aggregated results

Reviewer Selection Bias: Occurs when some users, often with extreme experiences, are more likely to review, skewing the rating distribution

▷ Leads to an unbalanced and non-representative set of ratings

Temporal Bias: Ratings fluctuate over time due to changes in product quality, update, service improvements, or upgrade/degradation

▷ Use time-decay models to down-weight older reviews

Platform-Induced Bias: Platforms can introduce biases through default rating settings or by incentivizing users to leave reviews

- Default Rating Bias: Some platforms pre-select a default rating (e.g., 5 stars), influencing users to submit ratings without much thought
- Incentivized Reviews: Platforms may offer rewards or discounts in exchange for reviews, which often leads to artificially inflated ratings

Debiasing Algorithms

Debiasing algorithms aim to mitigate the effects of various biases

Statistical Methods for Debiasing Review Aggregation often assume certain statistical properties about unbiased reviews and use those properties to adjust the ratings

- Weighted Averages: Assign different weights to reviews based on their perceived bias. For example, recent reviews may be weighted more heavily than older ones
- Bias-Correction Models: Estimate the bias in each review and apply a correction to bring it closer to the expected true rating
- Propensity Score Adjustment: Adjust reviews based on the likelihood of a review being biased

Handling Noise and Uncertainty in Rating Systems

Noise and uncertainty in rating systems can arise due to randomness, human error, or anomalous behavior. These factors can significantly impact the accuracy and reliability of aggregated ratings

Sources of Noise in Crowdsourced and User-Generated Ratings:

- Random User Behavior: Users may assign ratings arbitrarily without careful evaluation
- Inconsistent Standards: Different users have varying interpretations of rating scales, leading to inconsistent ratings
- Human Error: Accidental misclicks or misunderstandings of the product being rated
- External Factors: External biases, such as emotional state or environmental influences, can distort ratings

Anomalies in Ratings Due to Noise and Randomness

Anomalies in ratings caused by noise and randomness include outliers or sudden deviations from the usual pattern of ratings

Types of Anomalies:

- Outliers: Extreme ratings that deviate significantly from rest of data
- Fluctuations in Ratings: High variability in ratings for a product, often caused by noise
- Random or Baseless Ratings: Users assigning ratings without any valid reason

- Outlier Detection: Identify ratings that deviate significantly
- Smoothing: Apply moving average or exponential smoothing
- Weighted Averaging: Weigh noisy or suspicious ratings lower
- Noise Filtering: Model and filter out noise based on patterns

Aggregating Inconsistent or Conflicting Ratings

Inconsistent or conflicting ratings are common. Users may have divergent opinions or provide contradictory ratings for the same product or service

Aggregating such ratings requires models to produce a consensus that minimizes the impact of conflicts and reflects an accurate assessment

- Consensus Models: Aim to resolve disagreements in ratings by finding an aggregate score that best represents the group's overall opinion
  - Median-Based Consensus: Median minimizes influence of outliers and provides a central estimate
  - Weighted Consensus: Weigh ratings based on the credibility of the reviewers or the strength of the evidence behind the ratings
  - Delphi Method: Iterative approach where experts provide ratings, and the aggregation is revised until consensus is reached
- Probabilistic Models: Use distributions (e.g., Beta, Dirichlet) to account for uncertainty in the ratings

Trust and Reputation in Online Platforms

Reputation System collects, distributes, and aggregates information about behavior

Examples: BBB, Bizrate, eBay, Epinions

Reputation is mainly used to assert sellers' trustworthiness

- Explicit Trust Models: Trust ratings can be obtained via explicit feedback/review
  - ▷ Paid but didn't receive, not as advertised, slow shipping
- Implicit Trust Models: Trust is inferred from buyers' behavior (number of reviews written, accuracy of their past ratings, or reputation in community)
- Hybrid Trust Models: Combine both explicit and implicit trust signals to compute a final trust score for each user

Trust-based aggregation systems adjust the weight of ratings (e.g., a seller) based on the trustworthiness/reliability of the rater

▷ E.g., On Yelp, reviews from verified or frequent users have more influence

Trust Propagation in Reputation System

Trust propagation: Extending trust scores from known users to unknown users by considering their relationships or interactions.

Helps assign trust scores to users who have not been explicitly evaluated

- Trust Networks: Users are nodes, and trust relationships are edges. Trust propagates through the network to assign scores to all users
- Propagation Algorithms: TrustRank or personalized PageRank
- Reputation-Trust Feedback Loop: Trust and reputation influence each other, with higher reputation increasing trust in a user and trusted users contributing more reliable ratings

Review Inconsistencies, Spam Reviews

Inconsistencies due to differences in user experiences, expectations, or preferences

- Temporal Normalization: Adjust ratings based on the timing of the review (e.g., newer reviews may carry more weight)
- Contextual Adjustments: Incorporate factors like reviewer location or expertise to balance out contextual differences in reviews
- Weighting by Review History: Frequent reviewers with consistent reviewing patterns are given more weight in the aggregated rating

A hotel that receives varying reviews based on the time of year (e.g., peak season vs. off-season) will have its ratings normalized to account for these differences

Fake reviews, Spam, and Manipulation attempts

- Spam Reviews: Overly +ve/-ve reviews, generated by bots or paid users
- Review Manipulation: Inflated ratings by coordinating fake reviews
- Pattern Analysis: e.g., an unusual number of reviews in short time
- Review Filtering: Down-weights or removes suspicious reviews

Flag suspicion, if a product suddenly receives a spike in 5-star reviews from new accounts

Social Choice Theory: Voting and Ranking Aggregation

Voting Theory

Voting theory: The study of collective decision-making where individuals preferences are aggregated into a collective outcome

Rank aggregation techniques are methods used to combine multiple rankings into a single aggregated ranking

- Political systems: Elect officials or make decisions on policy matters
- Collaborative decisions: Group choices in committees/organizations
- Search Engine: Pagerank/HITS interpret links as votes
- Social Media: Upvoting and downvoting to rank posts and media
- Recommendaters: Aggregate preferences to recommend videos
- Crowdsourcing Platforms (Wikipedia): Consensus-driven voting methods to decide on content and edits

Voting Theory

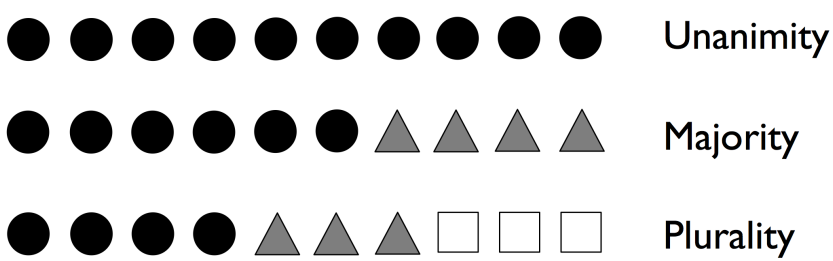
Voting: Problem Formulation

Aggregate individual preferences into a collective decision or outcome

- **Profiles:** The set of voters' preferences
- **Outcomes:** The possible decisions or alternatives
- **Voting Rule:** A function that maps profiles to an outcome

Voting

- What are the set of alternatives?
- What are the voters' preference orders or profiles?
- What is the aggregation method to determine the final outcome?



Profiles and Outcomes

$\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$: Alternatives/Candidates $\quad N = \{1, 2, \ldots, n\}$: Voters

Each voter $i \in N$ has a preference relation P_i over the alternatives in $\mathcal{A}$

▷ Typically modeled as a ranking (linear order) of the alternatives

A profile $\mathcal{P}$ is a set of rankings (preferences) for all voters:

$$\mathcal{P} = (P_1, P_2, \ldots, P_n)$$

Candidates $\mathcal{A} = \{A, B, C\}$ and voters $N = \{1, 2, 3\}$:

Profile $\mathcal{P} = (A \succ_1 B \succ_1 C,\ B \succ_2 C \succ_2 A,\ C \succ_3 A \succ_3 B)$

- Voter 1 ranking: $A \succ_1 B \succ_1 C$
- Voter 2 ranking: $B \succ_2 C \succ_2 A$
- Voter 3 ranking: $C \succ_3 A \succ_3 B$

An outcome is an alternative $W \in \mathcal{A}$ (winner) selected based on the profile

The function f maps a profile $\mathcal{P}$ to an outcome ▷ Single-winner outcome

Social welfare outcome: f produces a ranking (ordering) of all alternatives

Accuracy and Representation in Preference Aggregation

Accuracy in Preference Aggregation: Ensure that the outcome accurately reflects the true preferences of voters

- Strategic voting: Voters may misrepresent their preferences to influence the outcome in their favor
- Incomplete preferences: Not all voters may rank all alternatives, leading to incomplete profiles
- Majority bias: Some aggregation methods may favor the majority, ignoring minority preferences

Representation:

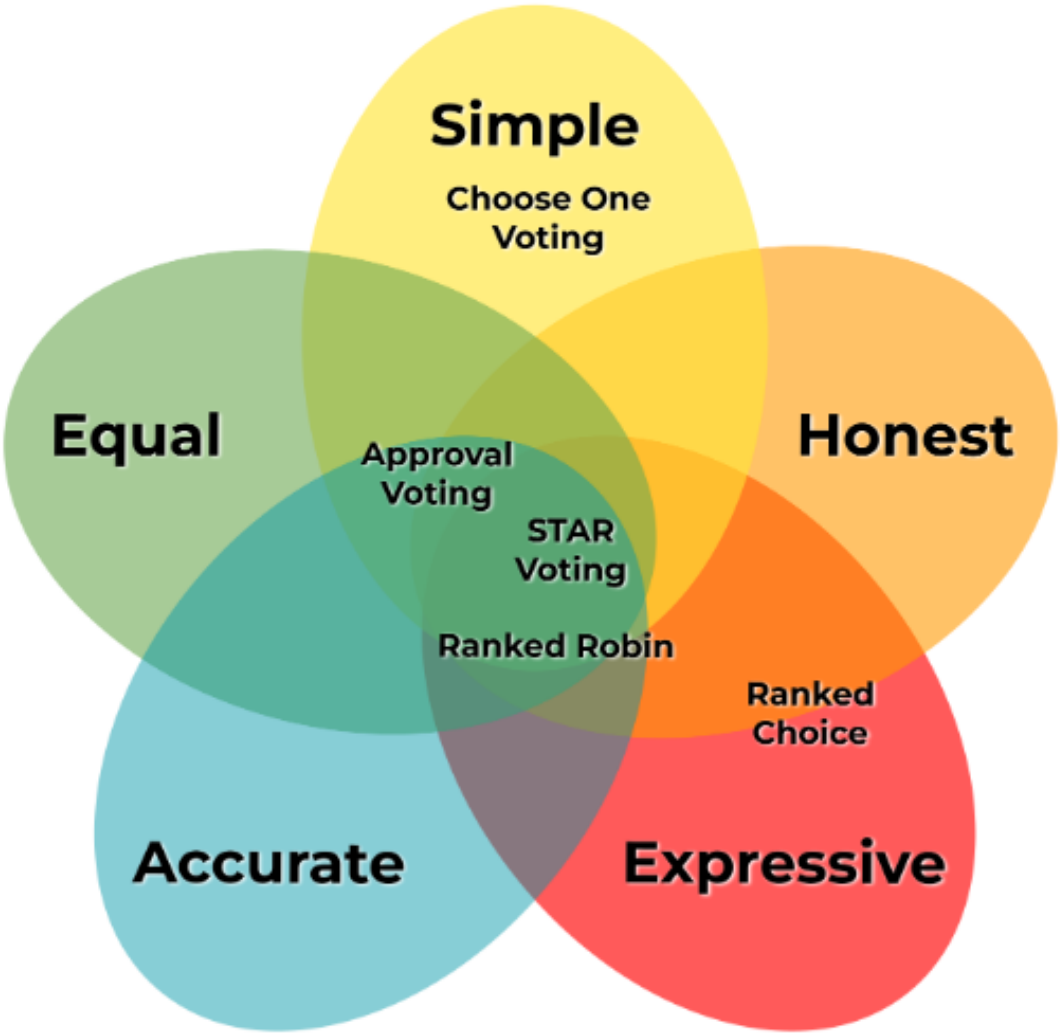
- Condorcet paradox: Collective preference can become cyclic $\implies$ impossible to select a clear winner
- Lack of expressiveness: Methods like plurality rule allow voters to express only their top choice, potentially losing valuable information about their preferences for other alternatives

Properties of Voting Systems

These properties are essential for ensuring that the voting system reflects the true preferences of the electorate in a fair and balanced manner

- Anonymity: All voters equal (their votes are given equal weight)
- Neutrality: All candidates equal
- Monotonicity: If a voter ranks a candidate higher, it should not harm the candidate's chances of winning
- Independence of Irrelevant Alternatives (IIA): The system's result should not change if a non-winning candidate is added or removed from the ballot
- **Condorcet loser criterion:** The candidate who would lose every pairwise contest does not win
- **Condorcet winner criterion:** The candidate who would win every pairwise contest always win
- Non-dictatorship: No single voter can determine the outcome
- Pareto efficiency: If all voters prefer one candidate over another, that candidate will win

Properties of Voting Systems



- **Simple:** easy to understand, easy to tabulate, easy to implement, easy to audit.
- **Honest:** safe to vote your conscience.
- **Expressive:** voters able to express their full opinion.
- **Accurate:** winners accurately reflect the will of the people.
- **Equal:** levels the playing field and gives every voter an equally powerful vote.

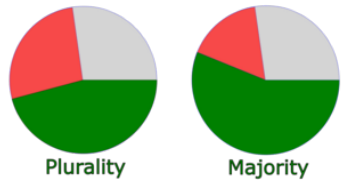
source: https://www.equal.vote/

Plurality Voting System

Mechanism: ▷ aka first-past-the-post (FPTP) used in Pak, US, UK

- Each voter selects one candidate from a set of alternatives
- The candidate who receives the most votes wins
- Need a tie-breaking rule ▷ e.g., runoff election or random draw

- Simple to understand: Each voter has a single vote to cast
- Winner-takes-all: The candidate with the most votes wins
- Majority not required: Can be won with less than 50% of the total vote if multiple candidates are running
- Susceptible to many issues



Candidates $\mathcal{A} = \{A, B, C\}$ and voters $N = \{1, 2, 3, 4, 5\}$:

- Voter 1: A, Voter 2: B, Voter 3: A, Voter 4: C, Voter 5: B

Vote counts: $[A : 2]$, $[B : 2]$, $[C : 1]$

Time complexity: $O(n)$, as we iterate through the voters once

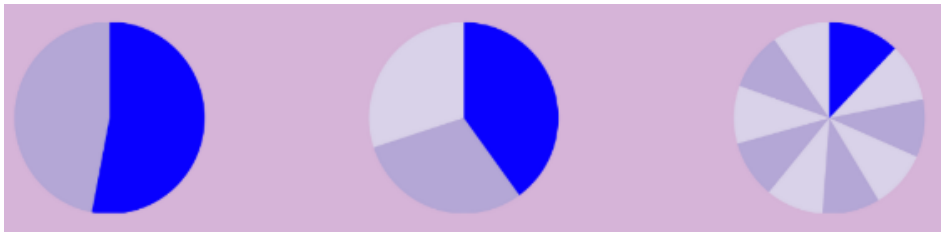
Strategic Voting

Voters may not vote for their true favorite candidate if they believe that candidate has little chance of winning

- Voters may choose to vote for a "lesser evil" candidate who has a better chance of winning over their least preferred candidate

  ▷ Leads to distorted outcome: the elected candidate may not be the most preferred choice of the majority

Candidate A is favorite of a small group, B and C are more popular. A's voters may strategically vote for B to prevent C from winning



source: Institute for Mathematics and Democracy

The Spoiler Effect

A candidate enters the race to split the vote and change the outcome in a way that might not reflect the true preferences of the electorate

The spoiler is often a candidate with no real chance of winning but who siphons votes away from a more viable candidate

- Candidate A (40% of votes)
- Candidate B (35% of votes)
- Candidate C (25% of votes)

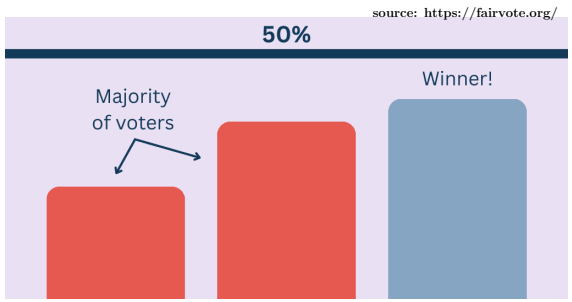
A wins with 40% of the vote. If C had not run, C's voters might have gone to B, allowing B to win a majority

For example, say a group is voting on their favorite fruit:

Candidate	% of votes
Banana	49%
Tangerine	48%
Clementine	3%

Banana wins, even though Clementine voters likely prefer Tangerine to Banana. Clementine is the spoiler.

source: Institute for Mathematics and Democracy



Vote Splitting in Plurality Voting

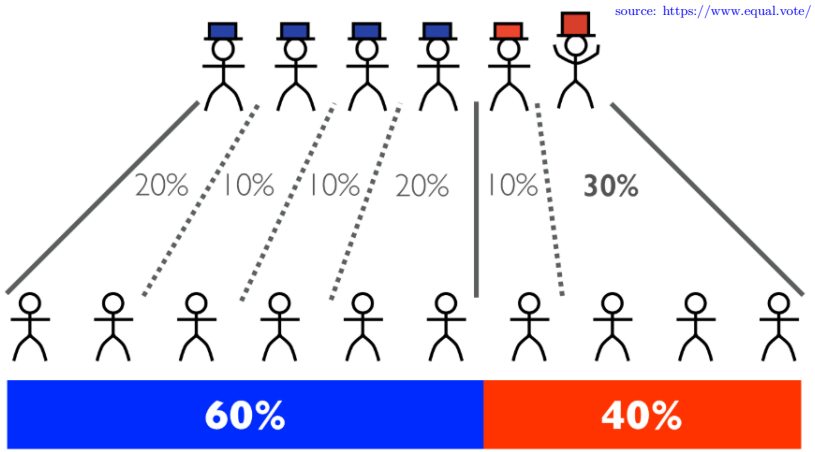
Two or more candidates appealing to the same voter group may split votes, potentially allowing a less popular candidate to win

Candidates $\{A, B, C\}$, where A and B share similar policy positions

A receives 35 votes B receives 30 votes C receives 35 votes

Here, Candidate C wins, even though candidates A and B together represent a majority of the voters (65 out of 100)

- Weakening of majority preference:
  A majority-preferred candidate can lose if their support base is divided among similar candidates
- Underrepresentation of ideologically similar candidates:
  They collectively hold more votes but lose due to vote splitting



Impact on Representation

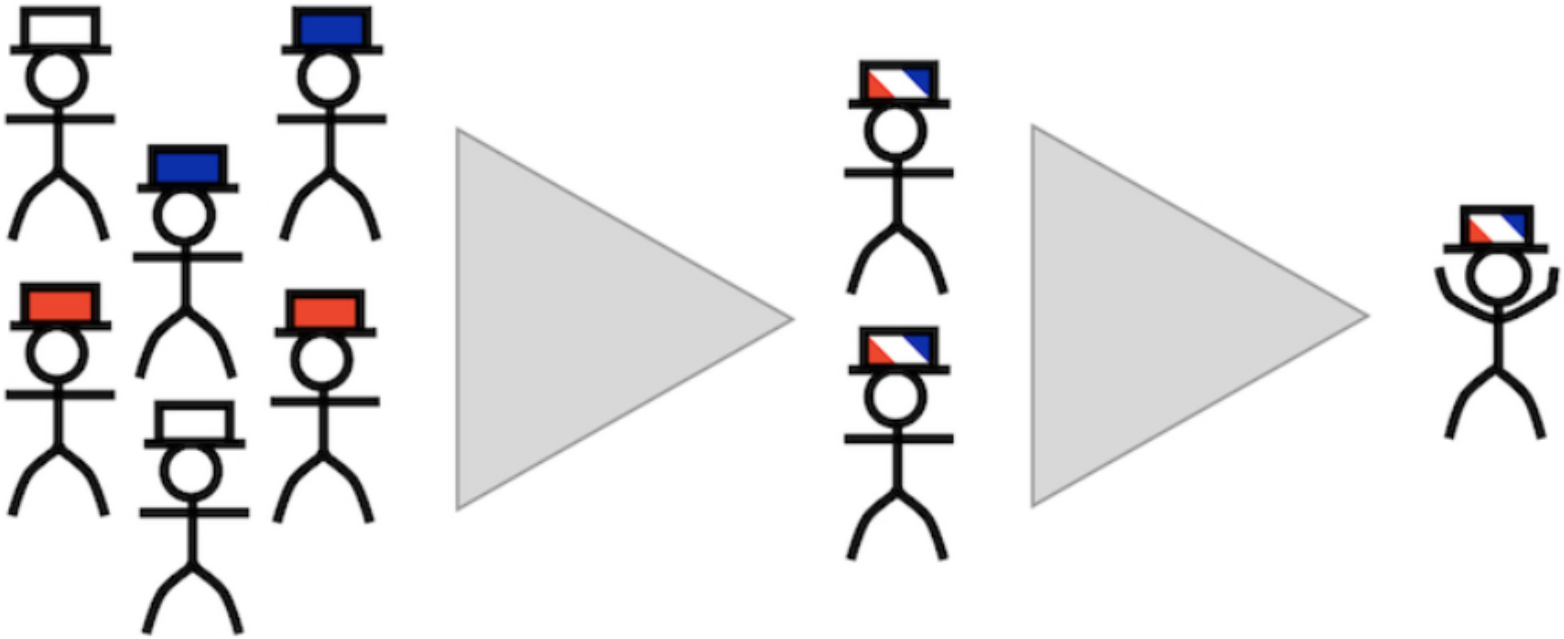
Plurality voting can distort representation, especially in elections with more than two candidates

- A candidate wins without the support of the majority of voters
- Minority candidates with strong support in specific regions or groups are underrepresented
- Popular candidates lose because of vote splitting, leading to unrepresentative outcomes

Runoff Voting

Runoff voting (a Sequential-Loser method) sometime also called Two-Round system is based on plurality voting

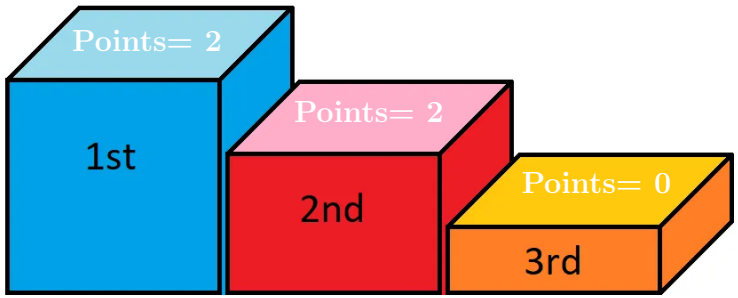
- The top two candidates from the first plurality voting round continue to the second round



Borda Count

Borda Count is a Positional Voting System It works as follows

- Voters rank the candidates from most to least preferred
- Each rank/position is assigned a specific point value, typically with the top-ranked candidate receiving the most points
  - A candidate ranked first by a voter receives the maximum number of points (equal to the total number of candidates minus 1)
  - A candidate ranked second receives one point fewer, and so on, until the last-ranked candidate receives zero points
  - Points assigned to each position can vary
- Points for candidates are summed over all voters to get final score



Borda Count: Aggregation

Algorithm 6 Borda Count

1: Input: Set of voters N and set of alternatives A
2: Initialize array $score[a] \leftarrow 0$ for each $a \in A$
3: **for** each voter $i \in N$ **do**
4: **for** each alternative a ranked in position k by voter i **do**
5: Add $m - k$ points to $score[a]$ ▷ Where m is the number of alternatives
6: Output: Alternative $a^* = \arg\max_{a \in A} score[a]$

Candidates $\mathcal{A} = \{A, B, C\}$ and voters $N = \{1, 2, 3\}$:

Voter 1: $B \succ_1 A \succ_1 C$ Voter 2: $C \succ_2 A \succ_2 B$ Voter 3: $A \succ_3 C \succ_3 B$

Ranking	Voter 1	Voter 2	Voter 3
	B	C	A
	A	A	C
	C	B	B

- A : Voter 1 (1 points), Voter 2 (1 points), Voter 3 (2 point)
- B : Voter 1 (2 point), Voter 2 (0 points), Voter 3 (0 points)
- C : Voter 1 (0 points), Voter 2 (2 point), Voter 3 (1 points)

Borda Scores: A: $1 + 1 + 2 = 4$ B: $2 + 0 + 0 = 2$ C: $0 + 2 + 1 = 3$

Weighting Positions

In positional voting systems, the number of points awarded to a candidate depends on their rank. Different systems use different weightings:

- Borda Count: Points decrease linearly based on rank, the top candidate gets $m-1$ points and last candidate gets 0 points
  - ▷ Balanced and comprehensive, susceptible to strategic voting
- Plurality Voting: Only the top-ranked candidate gets points
  - ▷ Simple, efficient, prone to vote splitting
- Anti-Plurality Voting: All candidates except last-ranked get points
  - ▷ Focuses on avoiding least popular option, ignores important voter preferences

- For a 3-candidate election, Borda count assigns 2 points to first-ranked candidate, 1 point to second, and 0 points to third
- In anti-plurality voting, the top two candidates each get 1 point, and last-ranked candidate gets 0 points

Fairness Borda Count

Borda count accounts for voters' entire preferences, not just top choices

m candidates and n voters. For candidate c, their Borda score is the sum of the points they receive from all voters

$$S_c = \sum_{i=1}^{n} (m - \text{rank}(c_i))$$

- Pairwise comparison: If a candidate consistently ranks higher than another across voters, they will have a higher score
- Borda count satisfies non-dictatorship and Pareto efficiency
- Borda count satisfies Condorcet loser criterion: A candidate, ranked last by majority will accumulate the lowest score and thus cannot win
- It may not satisfy Condorcet winner criterion

Strengths and Limitations of the Borda Count

- Comprehensive representation: Takes into account voters' entire preferences, not just top choices, leading to more balanced outcomes
  - ▷ Highly ranked candidate wins even if not top choice of all
  - ▷ Avoid extreme outcomes (candidate who is polarizing/disliked)
- Avoid vote splitting: The Borda Count reduces the risk of vote splitting by accounting for each voter's full ranking
- Reduced spoiler effect: Since candidates receive points based on their rank, third-party candidate does not significantly distort the result
- Susceptibility to tactical voting/manipulation: Voters can strategically rank less-preferred candidates lower to boost their favorite's score
  - ▷ A voter prefers A and to reduce B's score they may rank B last, even if B is not truly their least preferred candidate
- Failure to select the Condorcet winner: Borda Count does not always select the Condorcet winner

Manipulation in Borda Count

The Borda Count is susceptible to strategic voting

3 candidates: A, B, and C, and three voters. Each voter assigns points based on their ranking: 2 points for first, 1 for second, and 0 for third

True Preferences:

- Voter 1: $A \succ B \succ C$ $\quad (A(2), B(1), C(0))$
- Voter 2: $B \succ A \succ C$ $\quad (B(2), A(1), C(0))$
- Voter 3: $C \succ B \succ A$ $\quad (C(2), B(1), A(0))$

Final Tally: $[A : 3], [B : 4], [C : 2]$ Outcome: B wins

Voter 1 strategically votes $A \succ C \succ B$ to lower B's rank

Final Tally: $[A : 3], [B : 3], [C : 3]$

Outcome: A tie between A, B, and C, potentially favoring A in tiebreaker

Ranked-Choice Voting

Ranked-Choice Voting is more expressive than plurality

- Voters rank the candidates from most to least preferred
- If a candidate has majority of first-place vote, they win
- The candidate with fewest first-place vote is eliminated ▷ instant runoff
- The process is repeated until there is a majority

Round 1

	Voter				
	1	2	3	4	5
Ranking	A	A	B	B	C
	B	C	A	C	B
	C	B	C	A	A

C (least favored) is eliminated

Round 2

	Voter				
	1	2	3	4	5
Ranking	A	A	B	B	B
	B	B	A	A	A

Candidate B wins

- Unless a tie, the winner will receive the majority of votes
- Voters do not have to worry about wasting their vote
- Avoids the spoiler effect
- Encourages more diverse candidates to run
- Discourages negative campaigning

Approval Voting

Approval Voting is a single-winner rated voted system

- Voters can approve of as many candidates as they like
- Voters express their support for multiple candidates in no order

  ▷ Cardinal rather than ordinal voting

- The candidate with the most approval wins

Approve ☑ any number of candidates

☐ A
☐ B
☐ C
☐ D

Approval Ballot

	1	2	3	4	5
A	☑	☑	☐	☑	☑
B	☐	☑	☑	☑	☐
C	☑	☐	☐	☑	☑
D	☑	☐	☐	☑	☐

Candidate A wins (4 approvals)

- Eliminates the spoiler effect
- Reduces the chances of vote splitting
- Susceptible to strategic voting
- Votes cannot express true preferences

Condorcet Voting System: Pairwise Comparisons of Candidates

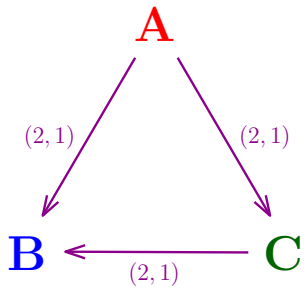
Condorcet Method: All candidate pairs are compared head-to-head

- Voters express their preferences between each pair of candidates
- Candidate preferred by majority in this pairwise comparison wins

Condorcet Aggregation of outcomes of pairwise contests to find winner

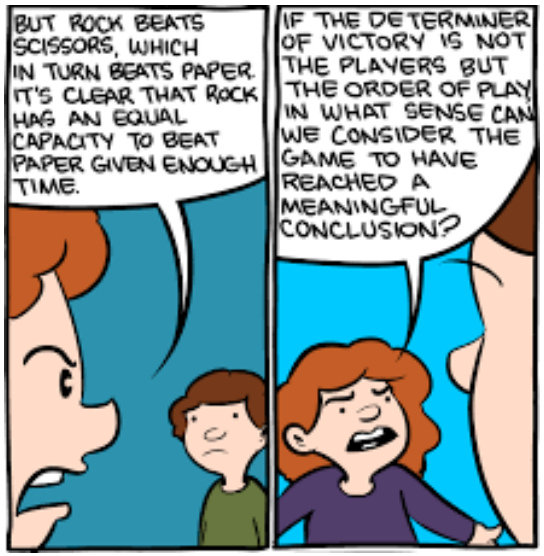
- Make a digraph with candidates as nodes and an edge from A to B means A wins in pairwise contest against B ▷ Majority prefers A over B
- The candidate who wins all pairwise comparisons has no incoming edges and is the Condorcet winner ▷ source node
- The Condorcet winner is not always the same as the plurality or Borda winner, especially in elections with more than two candidates
- If a Condorcet winner exists, they are a strong consensus choice

	Voter 1	2	3
Ranking	B	C	A
	A	A	C
	C	B	B



Cycle in Condorcet Voting

The Condorcet Paradox: When collective preferences, derived from pairwise comparisons, is cyclic, even though individual preferences are not



No clear winner in case of cyclic preferences (intransitive ranking)

▷ A source node may not exist if the graph is not a DAG

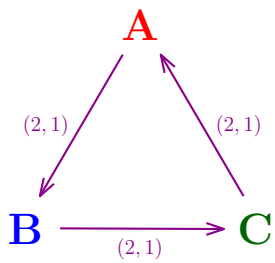
For 3 candidates (A, B, C), and the following 3 positional votes

Voter 1: $A \succ B \succ C$ Voter 2: $B \succ C \succ A$ Voter 3: $C \succ A \succ B$

Pairwise comparisons shows a cycle

- A vs. B: A wins (2 votes for A, 1 vote for B)
- B vs. C: B wins (2 votes for B, 1 vote for C)
- C vs. A: C wins (2 votes for C, 1 vote for A)

Ranking	Voter 1	Voter 2	Voter 3
1	A	B	C
2	B	C	A
3	C	A	B



- A defeats B B defeats C C defeats A

Impact of Cyclicity on Decision-Making

- Ambiguous Results: Cyclic preferences fails to provide a definitive outcome
- Need for Tie-Breakers: Additional rules to resolve the cycle
- Potential for Manipulation: Strategic voting exacerbate cyclic preferences

True Preferences:

- **Voter 1:** $A \succ B \succ C$
- **Voter 2:** $B \succ C \succ A$
- **Voter 3:** $C \succ A \succ B$

Pairwise results:

- A vs. B: A wins (2 to 1)
- B vs. C: B wins (2 to 1)
- C vs. A: C wins (2 to 1)

Voter 3, who prefers C, strategically votes $C \succ B \succ A$ to break the cycle in C's favor-changes outcome in pairwise comparison between A and B

Dealing with Cycles in Condorcet Voting

- Tideman's Ranked Pairs: Locks in the strongest pairwise victories while avoiding the creation of cycles
- Schulze Method: Identifies the "best" path of victories and breaks the cycle
- Random Tie-Breaking: Random tie-breaking to select the winner

Ranking Systems: Applications

Positional voting systems can be applied beyond elections to ranking problems where items or alternatives must be ranked by a group of people

- Ranking sports teams in competitions - usually Borda count
- Aggregating rankings for university or product/content reviews
  ▷ use ranking algorithms to aggregate user preferences and interactions with content
- Ranking candidates for job positions or grants
  ▷ Universities or journals may use positional systems to rank applicants for grants, fellowships, or awards
- Social media and crowdsourcing platforms (Reddit or Stack Overflow) use up/downvotes (forms of positional feedback) to rank posts

Distance Metric Between Rankings: Kendall Tau

Rank correlation metrics measure the agreement between two ranked lists

Kendall Tau: Measures similarity between two rankings by counting pairwise agreements and disagreements ▷ **Range:** [-1, 1]

$$\tau = \frac{|\{\textbf{concordant pairs}\}| - |\{\textbf{discordant pairs}\}|}{n(n-1)/2}$$

$$\tau = \frac{\left|\{(i,j) : i < j, \sigma(i) < \sigma(j), \pi(i) > \pi(j)\}\right|}{n(n-1)/2}$$

Three rankings of five contestants:

- π : 1, 2, 3, 4, 5
- σ : 2, 1, 4, 3, 5
- γ : 5, 4, 3, 2, 1

Total Pairs: $\binom{5}{2} = 10$

	conc. pairs	disc. pairs
(π, σ):	6	4
(π, γ):	0	10
(σ, γ):	4	6

$$\tau(\pi, \sigma) = \frac{6-4}{10} = 0.2$$

$$\tau(\pi, \gamma) = \frac{0-10}{10} = -1.0$$

$$\tau(\sigma, \gamma) = \frac{4-6}{10} = -0.2$$

Distance Metric Between Rankings: Spearman's Rank Correlation

Rank correlation metrics measure the strength and direction of association between two ranked lists

Spearman's Rank Correlation: Measures similarity between two rankings based on their relative positions ▷ **Range:** [-1, 1]

$$\rho = \frac{\text{COV}(R[\sigma], R[\pi])}{\text{STD}(R[\sigma])\text{STD}(R[\pi])} = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where $d_i = \sigma(i) - \pi(i)$ represents the rank differences

Three rankings of five contestants:

- π : 1, 2, 3, 4, 5
- σ : 2, 1, 4, 3, 5
- γ : 5, 4, 3, 2, 1

Total Pairs: $n = 5$

Pair	Rank Diffs
(π, σ)	$-1, 1, -1, 1, 0$
(π, γ)	$-4, -2, 0, 2, 4$
(σ, γ)	$-3, -3, 1, 1, 4$

$$\rho(\pi, \sigma) = 1 - \frac{6 \times 4}{5(25-1)} = 0.8$$

$$\rho(\pi, \gamma) = 1 - \frac{6 \times 40}{5(25-1)} = -1.0$$

$$\rho(\sigma, \gamma) = 1 - \frac{6 \times 36}{5(25-1)} = -0.8$$

Distance Metric Between Rankings: Spearman-Footrule Distance

Rank correlation metrics measure the sum of absolute differences between ranked lists

Spearman-Footrule Distance: Computes the displacements of two orderings

▷ **Range:** $[0,\ n(n-1)/2]$

$$F(\sigma, \pi) = \sum_i |\sigma(i) - \pi(i)|$$

where $|\sigma(i) - \pi(i)|$ represents the absolute rank differences

Three rankings of five contestants:

- π : 1, 2, 3, 4, 5
- σ : 2, 1, 4, 3, 5
- γ : 5, 4, 3, 2, 1

Total Pairs: $n = 5$

Pair	**Abs Diffs**
(π, σ)	$1, 1, 1, 1, 0$
(π, γ)	$4, 2, 0, 2, 4$
(σ, γ)	$3, 3, 1, 1, 4$

$F(\pi, \sigma) = 1+1+1+1+0 = 4$

$F(\pi, \gamma) = 4+2+0+2+4 = 12$

$F(\sigma, \gamma) = 3+3+1+1+4 = 12$

Rank Aggregation with Minimal Disagreement

The Kemeny-Young method combines individual voters' rankings into a "consensus ranking" that minimizes the number of pairwise disagreements with the voters' individual rankings

- A pairwise disagreement occurs when the consensus ranking reverses the order of two candidates compared to a voter's ranking

For two rankings of m candidates σ and π, Let $d(\sigma, \pi)$ be the number of pairwise disagreements between σ and π

Suppose $r_1, r_2, \ldots, r_n$ are complete rankings of m candidates by n voters

The KY method seeks to find a consensus ranking $\arg\min_{\sigma} \sum_{i=1}^{n} d(\sigma, r_i)$

1. For each pair (A, B), find number of votes for $A \succ B$ and $B \succ A$
2. Find one of the $m!$ orderings to maximize sum of scores for all pairs

Finding the optimal ordering is NP-Hard

Kemeny-Young Aggregation

For 3 candidates (A, B, C), and the following 3 positional votes

Voter 1: $A \succ B \succ C$ Voter 2: $B \succ C \succ A$ Voter 3: $C \succ A \succ B$

Pairwise Comparison: Compare candidates A, B, and C based on the voters' rankings

For each pair, count how many voters prefer one candidate over the other

- (A, B): A preferred by Voter 1 and Voter 3, B preferred by Voter 2. Result: A wins
- (A, C): A preferred by Voter 1, C preferred by Voter 2 and Voter 3. Result: C wins
- (B, C): B preferred by Voter 1, C preferred by Voter 2 and Voter 3. Result: C wins

Consensus Ranking: Based on these pairwise comparisons, the final ranking is $C \succ A \succ B$, minimizing the total pairwise disagreements

For each pair (i, j), count how often voter rankings disagree on their relative order

For m voters and n candidates, the complexity of computing the Kendall tau distance for all pairs is $O(m \times n^2)$

Find one of $n!$ rankings that minimizes total Kendall tau distance over all voters

Five Axioms of Arrow's Theorem

Arrow's Impossibility Theorem: No rank-order voting system can convert individual preferences into a collective decision meeting all of the following five axioms (desirable characteristics) under certain conditions

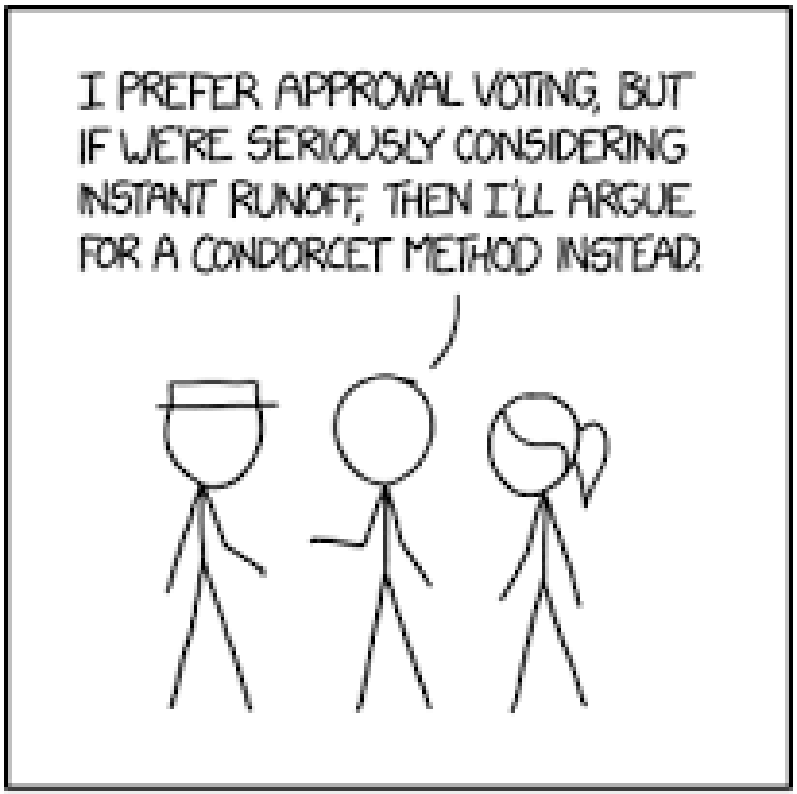
1. Universality: The voting system should work for all voter preferences
2. Pareto Efficiency: If all voters prefer one candidate to another, the system should reflect this

   If $A \succ_i B$ for all i, then the social ranking should have $A \succ B$
3. Independence of Irrelevant Alternatives (IIA): Ranking between two candidates should not be affected by the presence of other candidates
4. Non-Dictatorship: No single voter should dictate the outcome
5. Transitivity: Collective preference should be transitive
   $(A \succ B \wedge B \succ A \implies A \succ C)$

Practical voting systems often relax one or more of Arrow's axioms to achieve acceptable outcomes in real-world settings

Five Axioms of Arrow's Theorem

Arrow's Impossibility Theorem: No rank-order voting system can convert individual preferences into a collective decision meeting all of the following five axioms (desirable characteristics) under certain conditions



Independence of Irrelevant Alternatives (IIA)

The IIA axiom: Relative Ranking between two candidates should not be affected by the presence of other candidates

Consider three candidates $\{A, B, C\}$ and the following voters preferences (not votes)

- $A \succ B \succ C$ 25% voters
- $B \succ C \succ A$ 40% voters
- $C \succ A \succ B$ 35% voters

- Plurality Winner?
- Borda Winner?
- Condorcet Winner?

For any voting method (any way of aggregating these preferences)

- Case 1: A wins $\implies$ IIA violated (75% would vote $C \succ A$ if B was not present)
- Case 2: B wins $\implies$ IIA violated (60% would vote $A \succ B$ if C was not present)
- Case 3: C wins $\implies$ IIA violated (65% would vote $B \succ C$ if A was not present)

IIA violations are common and lead to counterintuitive outcomes

- Spoiler Effect
- Strategic Voting
- Impact on Policy and Perception: Perception of undue influence of irrelevant alternatives undermine trust in the fairness of the election

Sen's Axioms and Impossibility Theorem

Sen's Impossibility theorem: "under certain conditions, it is impossible to design a social decision function satisfying two axioms:

1. Decisive Voter Axiom (Minimal Liberalism): For every candidate pair (A, B), there exists at least one decisive voter , i.e., the group's preference between A and B always reflects the preference of this decisive voter, regardless of others' preferences

   Preferences of a single voter can override the collective will of the group

2. Transitivity in Group Decisions: The collective preferences must be transitive. If the group prefers A over B and B over C, then the group must prefer A over C. Transitivity ensures that the group's preferences is acyclic

Consider three candidates $\{A, B, C\}$ and the following voters preferences

Voter 1: $A \succ B \succ C$ Voter 2: $B \succ C \succ A$ Voter 3: $C \succ A \succ B$

Apply Decisive Voter Axiom: Assume Voter 1 is decisive for the pair $A \succ B$. According to the Decisive Voter Axiom, the group preference must also be $A \succ B$

Apply Transitivity: Now consider the group's preferences between A, B, and C. By transitivity, the group must rank $A \succ C$ and $B \succ C$, but this leads to a cycle where transitivity breaks down

Sen's Theorem - Example

To make this more clear lets look at the following example:

Suppose a two-member search committee for an economics department is charged with hiring one of the final candidates Amy, Bill, and Cindy.

As part of evaluation, each candidate's citation index and quality of published papers are examined; assume this leads to the following listing of relevant information

Candidate	**IQ**	**Macro citations**	**Micro citations**	**Years from PhD**
Amy	120	60	10	4
Bill	110	55	80	3
Cindy	100	65	70	5

(1)

Sen's Theorem - Example (continued)

The rules for assembling the committee ranking are natural:

- Unrestricted Domain: Each committee member can rank the candidates in any desired manner as long as the ranking is complete and transitive
- Pareto. If everyone ranks a pair in the same manner, this common ranking will be the committee's ranking
- Minimal Liberalism (ML)—or Division of Expertise. Committee members were selected because of their expertise. As Garrett's expertise is macroeconomics—an area in which both Amy and Bill claim ability—it is natural to defer to Garrett's knowledge by asserting that how he ranks Amy and Bill will be the committee ranking. Similarly, Sandy is an expert in microeconomics where both Bill and Cindy claim abilities: Sandy's ranking of Bill and Cindy determines their committee ranking
- No Cycles. In order to make a decision, the committee ranking must be cycle free

Sen's Theorem - Example

By identifying this example with Sen's Theorem, it follows that with three or more candidates and two or more committee members, *no ranking rule will always satisfy these conditions*

Without the transitivity assumption, Sen's result loses all surprise: the conclusion becomes obvious and immediate. After all, if the voters can have cyclic preferences, then we must expect cyclic societal outcomes

Sen's assertion mandates that situations exist where the search committee cannot satisfy the specified requirements. Indeed, according to the table:

1. Garrett prefers Amy over Bill because of their relative performances in macroeconomics.
2. His one disappointment is that Cindy, who has the best performance, no longer is interested in this area: Garrett's ranking is $C \succ A \succ B$
3. Sandy, on the other hand, is impressed with Bill's citation record based on his theory papers, so she ranks Bill above Cindy. But with her negative opinion of Amy, Sandy's ranking is $B \succ C \succ A$

Sen's Theorem

Using a dash to represent where information from a person is irrelevant (because the decision is determined by another member), the information used to assemble the committee ranking follows:

Member	**Ranking**	$\{A, B\}$	$\{B, C\}$	$\{A, C\}$
Garrett	$C \succ A \succ B$	$A \succ B$	$-$	$C \succ A$
Sandy	$B \succ C \succ A$	$-$	$B \succ C$	$C \succ A$
	Committee Ranking	$A \succ B$	$B \succ C$	$C \succ A$

Sen's assertion is demonstrated by the cyclic outcome, and, in practice, by the need to hold more committee meetings. Obvious modifications can be made; e.g., each committee member could be replaced with several members where decisions are made by majority vote, the committee could use a wider assortment of information including letters of recommendation, etc.

Intensity of Preference (IIIA)

The Intensity of Preference (IIIA) measures not just the order of voters' preferences but also the strength of those preferences

In contrast to IIA treats all preferences equally

- Voters indicate the degree of their preference between candidate pairs
- System aggregates rankings and intensity of preferences (numeric values)
- IIIA incorporates strengths of preferences $\implies$ more representative outcome

Example: Voter 1: $A \succ B$ [7] and $B \succ C$ [3]

- More Nuanced Collective Decisions
- Better Reflects Voter Sentiment
- Reduces Impact of Irrelevant Alternatives

Possible Voting Systems that Avoid Arrow's Paradox

Alternative voting systems can be constructed based on modified assumptions that avoid the paradoxes identified in Arrow's theorem for use in

Used in corporate governance, academic peer review, crowdsourcing and collaborative platforms

- Single-Peaked Preferences: Restrict preferences to be single-peaked, avoids many Arrow's paradoxes, as there is a clear "middle" candidate who can win
- Weighted Voting Systems: Allowing for weighted votes based on intensity of preference or expertise. can resolve conflicts between axioms
- Ranked Pairs or Schulze Method: Avoid some cyclic inconsistencies in Condorcet methods by focusing on the strongest pairwise victories
- Approval Voting: Voters can approve of multiple candidates, and candidate with most approvals wins. Avoids some issues with rank-order methods
- Majority Judgment: Ranks candidates based on median evaluation of voters, rather than on rank-order or pairwise comparisons, avoids many paradoxes
- Range Voting: Voters assign a score to each candidate, and the candidate with the highest total score wins, avoids the need for strict rankings or IIA