Web Science

19 February 2019

Christina Lioma
c.lioma@di.ku.dk

Lecture plan

- Collective Intelligence II
  - Crowdsourcing evaluation
  - Indirect collective intelligence

Previous lecture:

- Collective intelligence: definition and characteristics

Previous lecture:

- Collective intelligence: definition and characteristics
- Crowdsourcing
  - Aggregate correct answers from high number of error-prone and disagreeing workers
  - Choose tasks and workers according to performance

Previous lecture:

- Collective intelligence: definition and characteristics
- Crowdsourcing
  - Aggregate correct answers from high number of error-prone and disagreeing workers
  - Choose tasks and workers according to performance

Today's lecture:

- Evaluation of crowdsourced tasks

- Indirect collective intelligence

Previous lecture:

- Collective intelligence: definition and characteristics
- Crowdsourcing
  - Aggregate correct answers from high number of error-prone and disagreeing workers
  - Choose tasks and workers according to performance

Today's lecture:

- Evaluation of crowdsourced tasks
  1. Worker consistency
  2. Reproducibility
  3. Error rate
- Indirect collective intelligence

(1) Measure how consistently workers perform throughout the task

- not in terms of their answers, but in terms of their modus operandi

(1) Measure how consistently workers perform throughout the task

- not in terms of their answers, but in terms of their modus operandi

Goal: capture diversity without very extreme outliers

(1) Measure how consistently workers perform throughout the task

- not in terms of their answers, but in terms of their modus operandi

Goal: capture diversity without very extreme outliers

Common practice:

- <u>Mean</u> task completion time (measured in milliseconds)
- Per-worker <u>standard deviation</u> in task completion times

Problem:

- mean & standard deviation are affected by *very* extreme outliers
- very extreme outliers are rare in general, but not so rare in crowdsourced data

Problem:

- mean & standard deviation are affected by *very* extreme outliers
- very extreme outliers are rare in general, but not so rare in crowdsourced data

Common example:

- worker spends 1.5 seconds on trivial task A
- same worker spends >2 minutes on trivial task B (distracted by external event in the middle of the experiment)

Toy example of mean & standard deviation being affected by *very* extreme outliers

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4 (no extreme outlier)

Toy example of mean & standard deviation being affected by *very* extreme outliers

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4 (no extreme outlier)

Mean: ?

Standard deviation: ?

Toy example of mean & standard deviation being affected by *very* extreme outliers

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4 (no extreme outlier)

Mean: 2.45

Standard deviation: 1.04

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 400 (with extreme outlier)

Mean: ?

Standard deviation: ?

Toy example of mean & standard deviation being affected by *very* extreme outliers

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4 (no extreme outlier)

Mean: 2.45

Standard deviation: 1.04

Input: 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 400 (with extreme outlier)

Mean: 35.58

Standard deviation: 114.77

Solution:

- First, detect and remove *very* extreme outliers
- Then, compute mean and standard deviation in task completion time

Solution:

- First, detect and remove *very* extreme outliers
- Then, compute mean and standard deviation in task completion time

Detect very extreme outliers based on (either):

- log-transformed per-worker mean time
- log-transformed per-worker maximum time

Two common ways

Method (1):

- Compute mean & standard deviation

Method (1):

- Compute mean & standard deviation
- Data that is more than two standard deviations from the mean → *very* extreme outliers

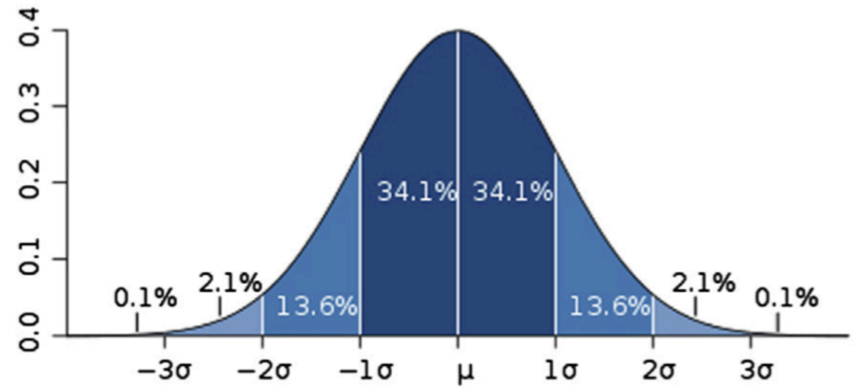


Figure 3 The normal curve.

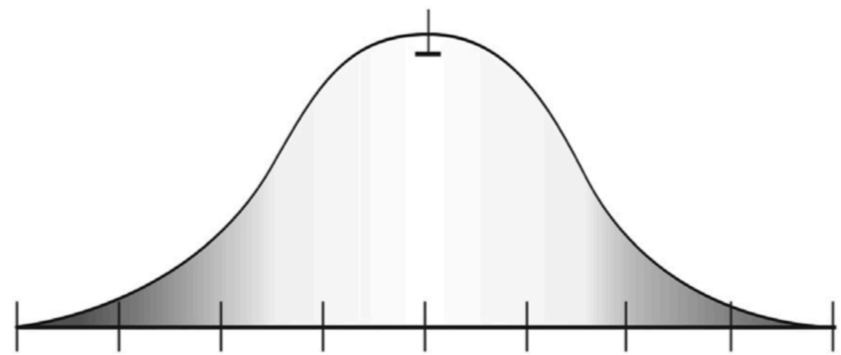


Figure 4 The normal curve (with darker areas at either extreme from the median).

Method (1):

- Compute mean & standard deviation
- Data that is more than two standard deviations from the mean → *very* extreme outliers
- Remove extreme outliers and re-compute mean & standard deviation

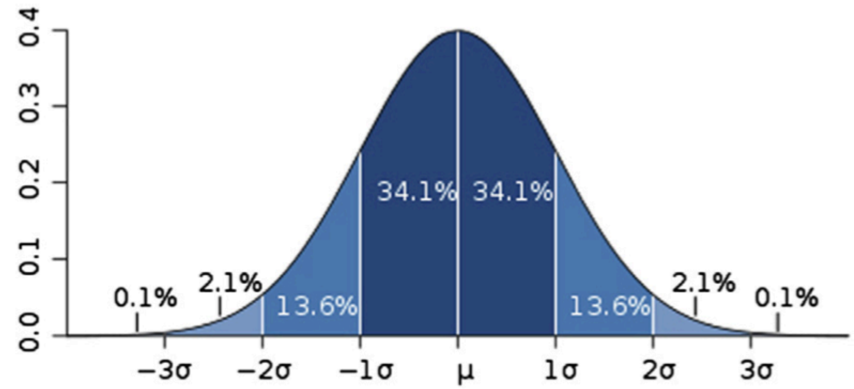


Figure 3 The normal curve.

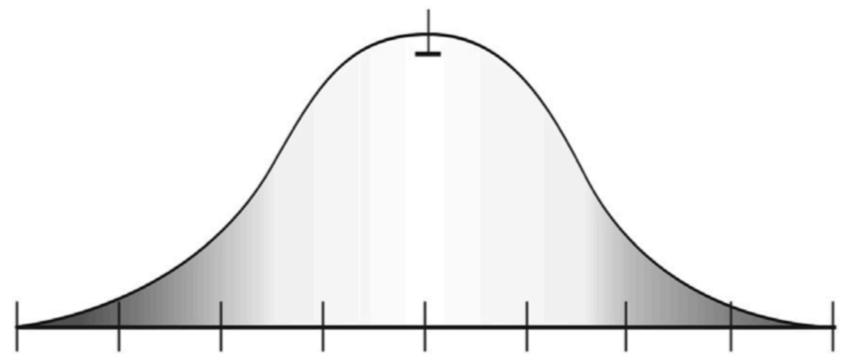


Figure 4 The normal curve (with darker areas at either extreme from the median).

Method (1):

- Compute mean & standard deviation
- Data that is more than two standard deviations from the mean → *very* extreme outliers
- Remove extreme outliers and re-compute mean & standard deviation

Assumption: data is normally distributed

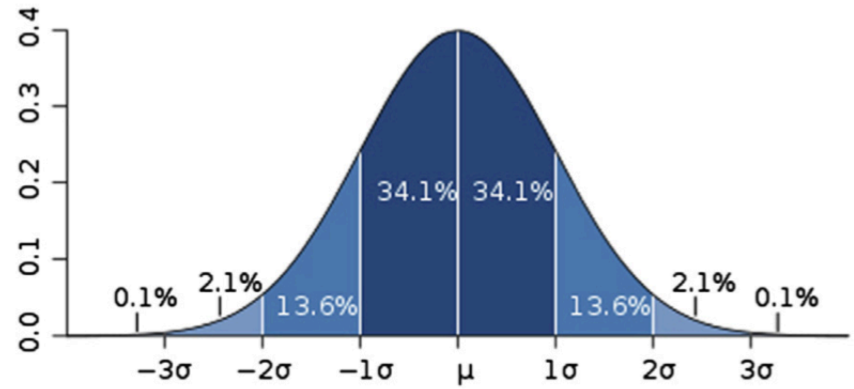


Figure 3 The normal curve.

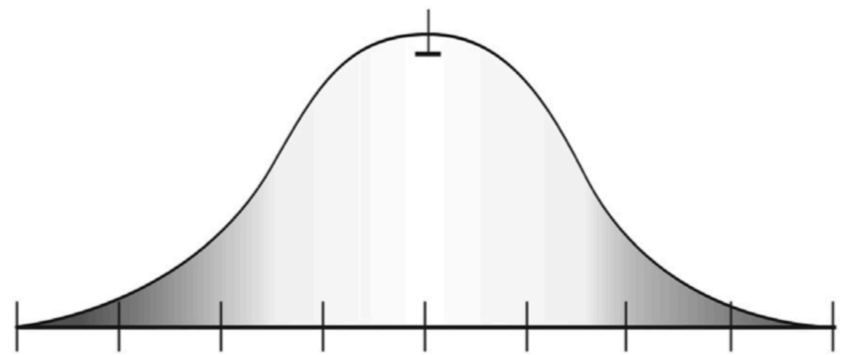


Figure 4 The normal curve (with darker areas at either extreme from the median).

Method (1):

- Compute mean & standard deviation
- Data that is more than two standard deviations from the mean → *very* extreme outliers
- Remove extreme outliers and re-compute mean & standard deviation

Assumption: data is normally distributed

Problem:

- *Very* extreme outliers bias mean & standard deviation
- Not robust

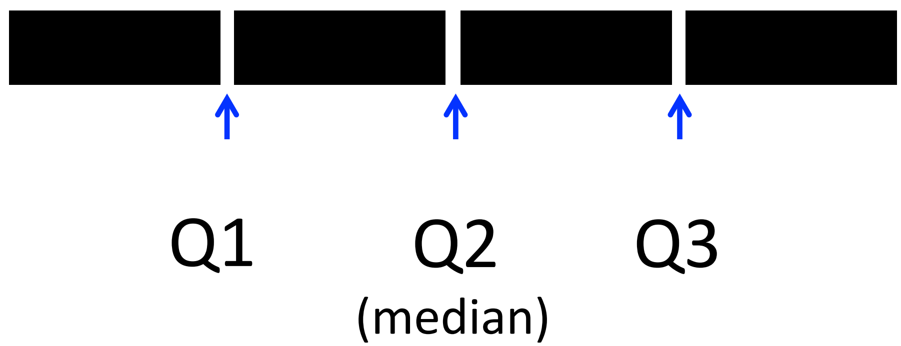
Method (2): detect outliers with InterQuartile Range (IQR)

Method (2): detect outliers with InterQuartile Range (IQR)

- Sort input data
- Divide sorted data into 4 equal parts

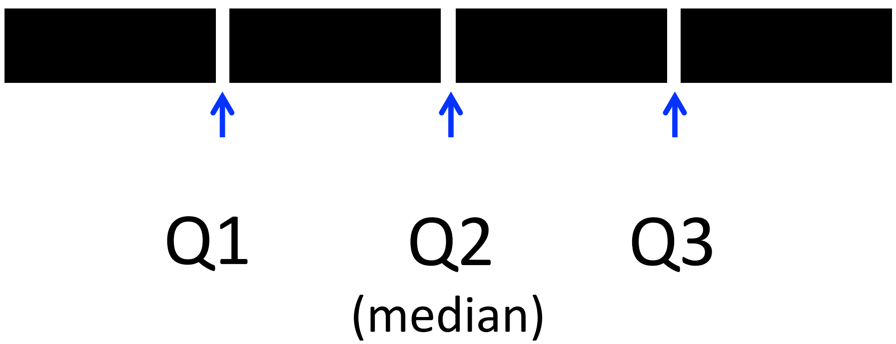
Method (2): detect outliers with InterQuartile Range (IQR)

- Sort input data
- Divide sorted data into 4 equal parts



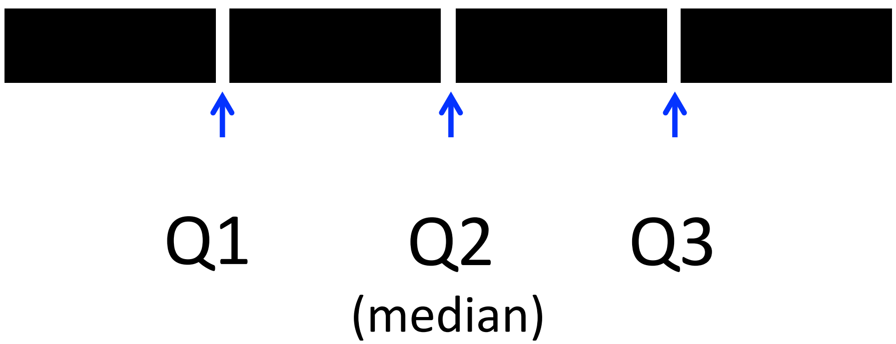
Method (2): detect outliers with InterQuartile Range (IQR)

- Sort input data
- Divide sorted data into 4 equal parts
- IQR = Q3 − Q1



Method (2): detect outliers with InterQuartile Range (IQR)

- Sort input data
- Divide sorted data into 4 equal parts
- IQR = Q3 − Q1

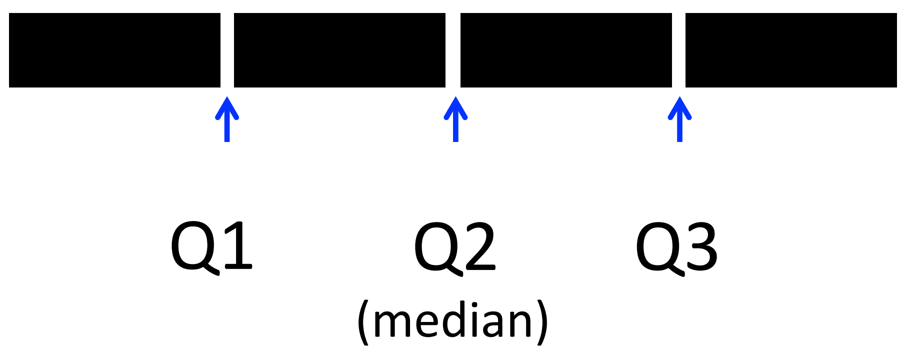


Extreme outlier:

- 3 x IQR more than Q3
- 3 x IQR less than Q1

Method (2): detect outliers with InterQuartile Range (IQR)

- Sort input data
- Divide sorted data into 4 equal parts
- IQR = Q3 − Q1



Will this remove more or less outliers than method 1?

Extreme outlier:

- 3 x IQR more than Q3
- 3 x IQR less than Q1

Example input range: 1, 3, 4, 5, 5, 6, 7, 11

- Method 1: 2 standard deviations from mean

- Method 2: 3 x IQR from Q1 and Q3

Example input range: 1, 3, 4, 5, 5, 6, 7, 11

- Method 1: 2 standard deviations from mean

$\mu = 5.25$, $\sigma = 2.96$

Extreme outliers: less than -0.67 and more than 11.17

- Method 2: 3 x IQR from Q1 and Q3

Example input range: 1, 3, 4, 5, 5, 6, 7, 11

- Method 1: 2 standard deviations from mean

$\mu = 5.25$, $\sigma = 2.96$

Extreme outliers: less than -0.67 and more than 11.17

- Method 2: 3 x IQR from Q1 and Q3

Q1 = 3.5, Q3 = 6.5, IQR = 6.5 − 3.5 = 3

Extreme outliers: less than -5.5 and more than 15.5

Example input range: 1, 3, 4, 5, 5, 6, 7, 11

- Method 1: 2 standard deviations from mean

$\mu = 5.25$, $\sigma = 2.96$

Extreme outliers: less than -0.67 and more than 11.17

Removes 4.6% of the data

- Method 2: 3 x IQR from Q1 and Q3

Q1 = 3.5, Q3 = 6.5, IQR = 6.5 − 3.5 = 3

Extreme outliers: less than -5.5 and more than 15.5

Removes 0.0002% of the data (legitimate diversity kept)

Evaluation of crowdsourced tasks

1. ~~Worker consistency~~
2. **Reproducibility**
3. Error rate

Reproducibility: can the same conclusions be drawn if the study is repeated?

Reproducibility: can the same conclusions be drawn if the study is repeated?

Large variation of worker population over different times of the day → significant impact on performance

Reproducibility: can the same conclusions be drawn if the study is repeated?

Large variation of worker population over different times of the day → significant impact on performance

	Weekday 11am	Weekday 7pm
Female	74%	57%
Median age	43	32
Use mouse	63%	86%
Use touchpad	27%	14%

Reproducibility: can the same conclusions be drawn if the study is repeated?

Large variation of worker population over different times of the day → significant impact on performance

	Weekday 11am	Weekday 7pm
Female	74%	57%
Median age	43	32
Use mouse	63%	86%
Use touchpad	27%	14%

Lab versus crowdsourcing: no significant difference except for age group 45 – 65 (underrepresented in lab)

Evaluation of crowdsourced tasks

1. ~~Worker consistency~~
2. ~~Reproducibility~~
3. **Error rate**

“Error rate” in crowdsourcing

"Error rate" in crowdsourcing

E.g. label pictures showing cats (labels: cat, not-cat)

"Error rate" in crowdsourcing

E.g. label pictures showing cats (labels: cat, not-cat)

- How many cats were found out of all cats?
- How many non-cat pictures were mislabelled as cats?
- How many pictures were correctly labelled out of all pictures?

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP)
	false	false negative (FN)	true negative (TN)

"Error rate" in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) **Type I error**
	false	false negative (FN)	true negative (TN)

How many non-cat pictures were mislabelled as cats?

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) **Type II error**	true negative (TN)

How many cat pictures were mislabelled as not-cats?

"Error rate" in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

"Error rate" in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

(Sensitivity)

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN) (Sensitivity)

Specificity = TN / (TN+FP)

“Error rate” in crowdsourcing

		Ground truth	
		true	false
Worker answer	true	true positive (TP)	false positive (FP) Type I error
	false	false negative (FN) Type II error	true negative (TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

(Sensitivity)

Specificity =

TN / (TN+FP)

Accuracy: (TP+TN) / (TP+TN+FP+FN)

		Ground truth	
		true (5)	false (95)
Worker answer	true	true positive	false positive
	false	false negative	true negative

Accuracy: (TP+TN) / (TP+TN+FP+FN)

Imbalanced dataset (95 false and 5 true labels) → classifying all as false

		Ground truth	
		true (5)	false (95)
Worker answer	true (0)	true positive (0)	false positive (0)
	false (100)	false negative (5)	true negative (95)

Accuracy: (TP+TN) / (TP+TN+FP+FN)

Imbalanced dataset (95 false and 5 true labels) → classifying all as false **gives 0.95 accuracy**

		Ground truth	
		true (5)	false (95)
Worker answer	true (0)	true positive (0)	false positive (0)
	false (100)	false negative (5)	true negative (95)

Accuracy: (TP+TN) / (TP+TN+FP+FN)

Imbalanced dataset (95 false and 5 true labels) → classifying all as false gives 0.95 accuracy

		Ground truth	
		true (5)	false (95)
Worker answer	true (0)	true positive (0)	false positive (0)
	false (100)	false negative (5)	true negative (95)

Balanced accuracy = (recall + specificity)/2

= (0/5 + 95/95)/2 = 47.5

Quick quiz

- Go to: https://b.socrative.com/login/student/
- Enter as room name: LIOMA
- Enter a fake name for yourself
- Take the quiz

Lecture plan

- Collective Intelligence II
  - ~~Crowdsourcing evaluation~~
  - **Indirect collective intelligence**

Collective intelligence

- Direct: explicit information mining from users

Collective intelligence

- Direct: explicit information mining from users
- Indirect: implicit information mining from users

Collective intelligence

- Direct: explicit information mining from users
- Indirect: implicit information mining from users
  Famous case "find best webpages on the web"
  - "Best" approximated as "most popular"
  - "Popular" approximated as "hyperlinked"
  - Mine hyperlinks given to webpages by large crowds → detect best webpages

1921: George Polya, Prof. Mathematics

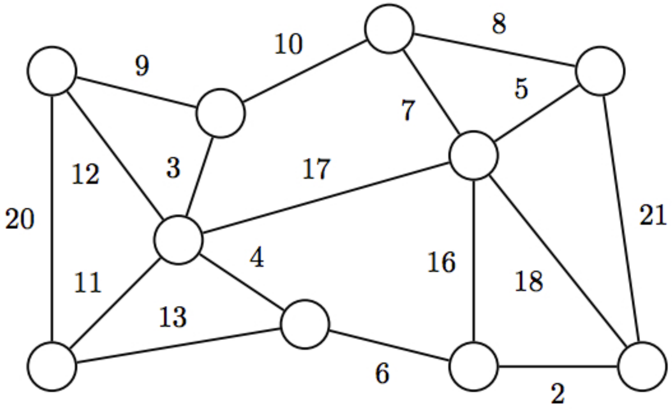
1921: George Polya, Prof. Mathematics

1921: George Polya, Prof. Mathematics

"I did not foresee that he and his fiancee would also set out for a stroll in the woods, and then suddenly I met them there. And then I met them the same morning repeatedly. I don't remember how many times, but certainly much too often and I felt embarrassed: It looked as if I was snooping around which was, I assure you, not the case"

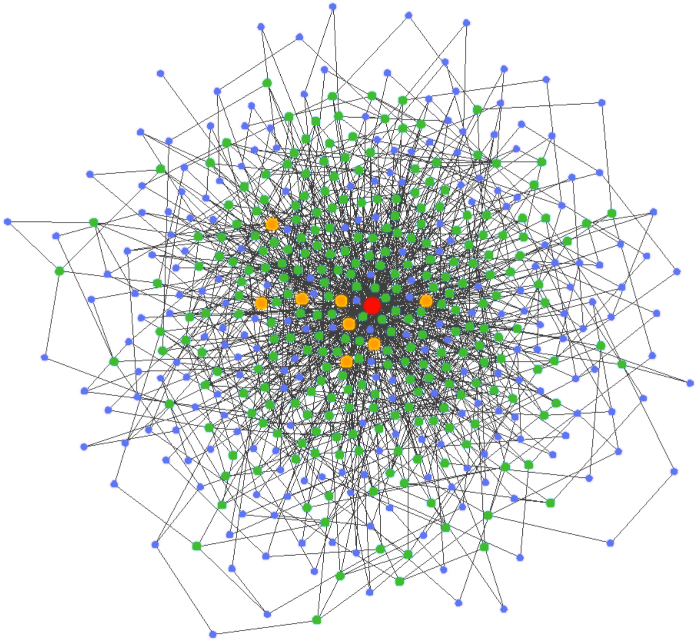
1921: George Polya, Prof. Mathematics

"I did not foresee that he and his fiancee would also set out for a stroll in the woods, and then suddenly I met them there. And then I met them the same morning repeatedly. I don't remember how many times, but certainly much too often and I felt embarrassed: It looked as if I was snooping around which was, I assure you, not the case"



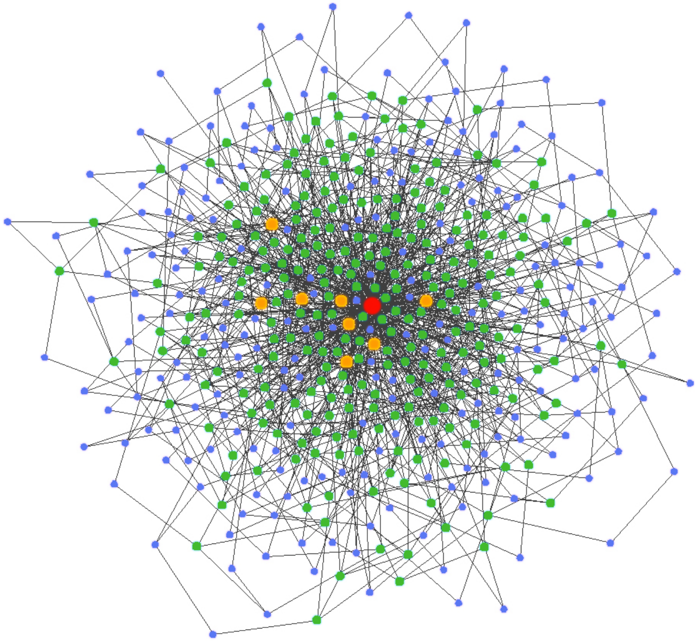
1996: Two CS students, University of Stanford

- Network: 518 million nodes (=webpages)
- Computer simulates Polya's walk (100s of times)



1996: Two CS students, University of Stanford

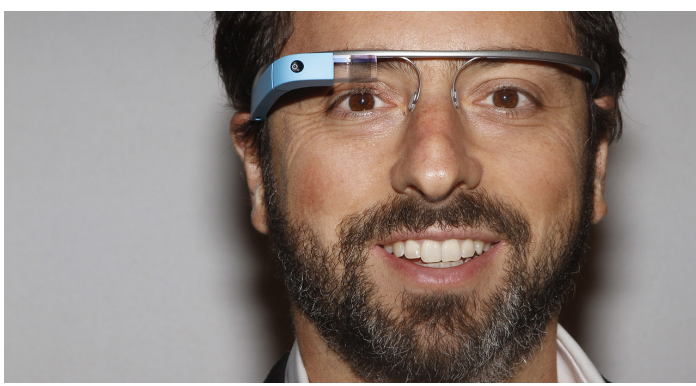
- Network: 518 million nodes (=webpages)
- Computer simulates Polya's walk (100s of times)
- Finding: most popular webpages
- PageRank algorithm



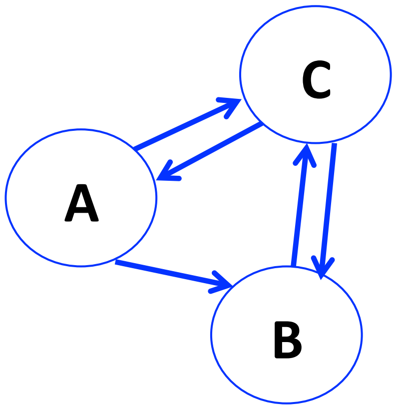
2019: Sergey Brin, Larry Page, Google Founders

Google:
150 trillion webpages in 2017
3.5 billion searches per day (world population: 7 billion)
Indirect collective intelligence of hyperlinks

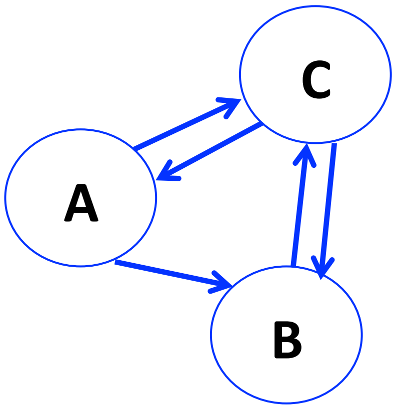
https://searchengineland.com/googles-search-indexes-hits-130-trillion-pages-documents-263378



PageRank



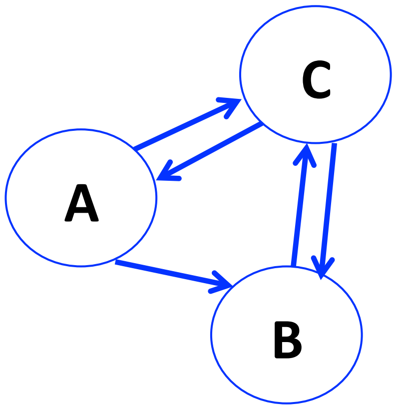
PageRank



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

PageRank

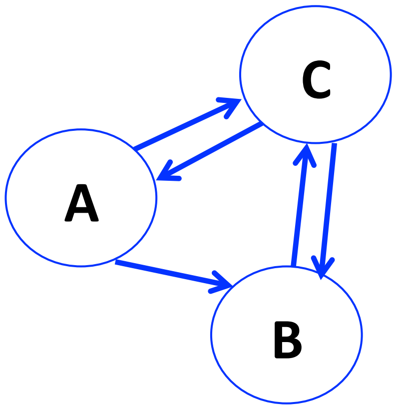
- Assume all pages are equally good. StartScore = 0.5. Aim: EndScore = ?



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

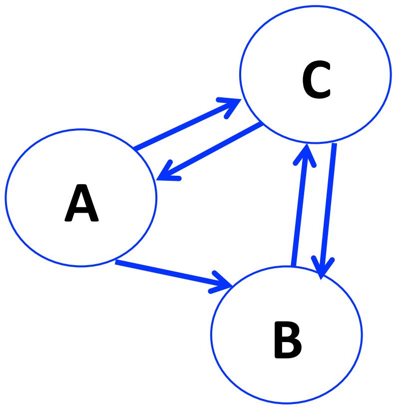
PageRank

- Assume all pages are equally good.
  StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

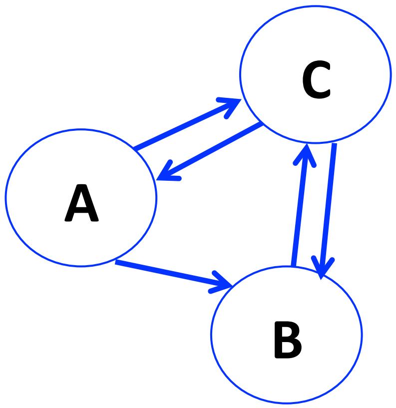
PageRank



- Assume all pages are equally good. StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C.

PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

PageRank

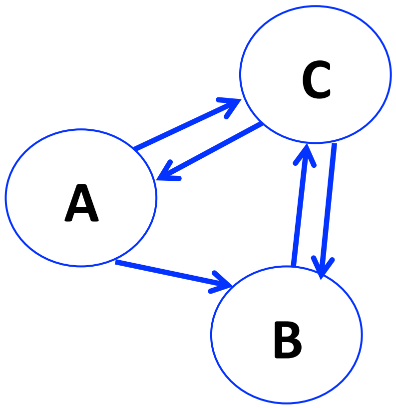


- Assume all pages are equally good. StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C. For each IN:
  (ii) How many outlinks (OUT) does C have?
  (iii) What is C's StartScore?

PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

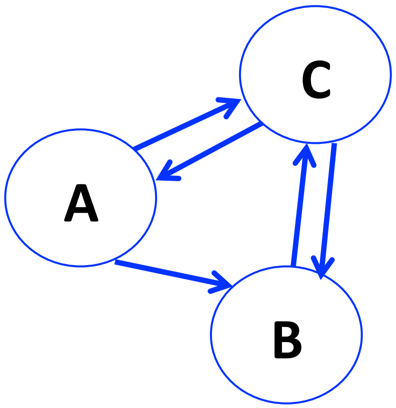
PageRank

- Assume all pages are equally good.
  StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C. For each IN:
  (ii) How many outlinks (OUT) does C have? 2
  (iii) What is C’s StartScore? 0.5



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

PageRank

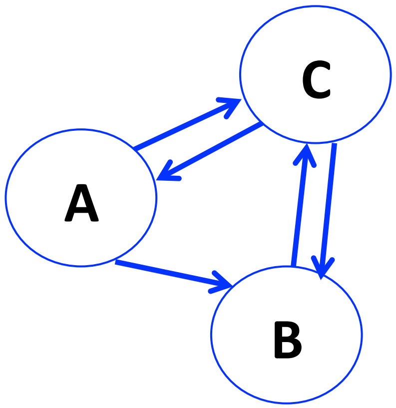


PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

- Assume all pages are equally good. StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C. For each IN:
  (ii) How many outlinks (OUT) does C have? 2
  (iii) What is C's StartScore? 0.5

EndScore (A) = Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank



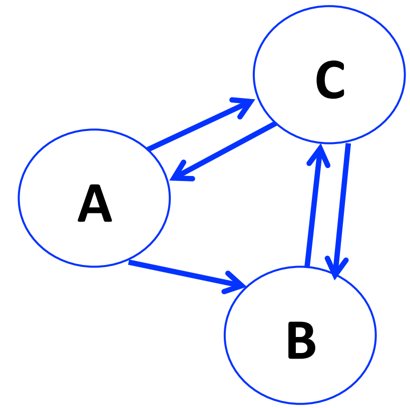
- Assume all pages are equally good. StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C. For each IN:
  (ii) How many outlinks (OUT) does C have? 2
  (iii) What is C's StartScore? 0.5

PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

EndScore (A) = Σ _(A's IN) (StartScore of IN / # OUT of IN)

EndScore(A) = 0.5 / 2 = 0.25

PageRank



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

- Assume all pages are equally good.
  StartScore = 0.5. Aim: EndScore = ?
- Take a random page, e.g. A, and ask:
  (i) How many inlinks (IN) does A have?
  1, from C. For each IN:
  (ii) How many outlinks (OUT) does C have? 2
  (iii) What is C's StartScore? 0.5

EndScore (A) = Σ _(A's IN) (StartScore of IN / # OUT of IN)
EndScore(A) = 0.5 / 2 = 0.25

- PageRank(A) = **(1 – d) + d x EndScore(A)**

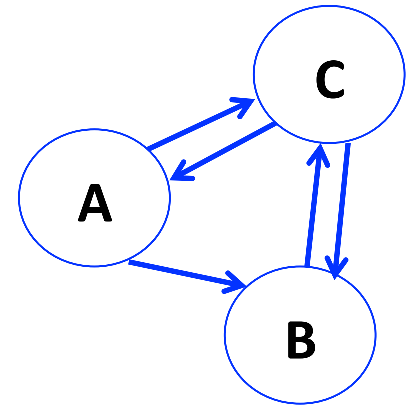
d: probability that user jumps between pages (d = 0-1)

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)



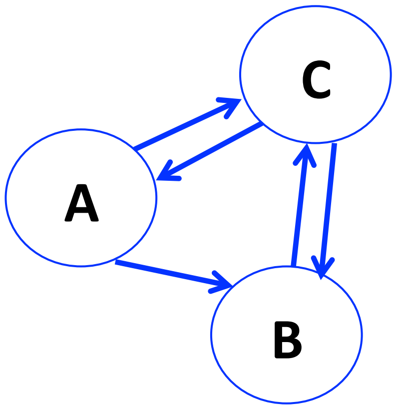
PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)



PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

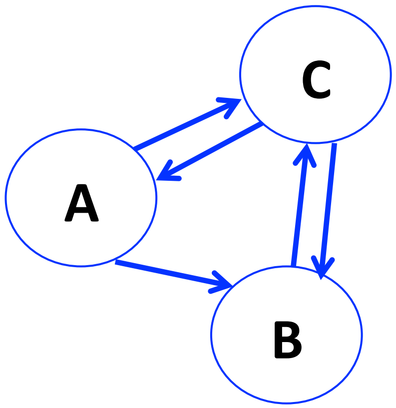
PAGE	**IN**	**OUT**
A	1	2
B	2	1
C	2	2

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

PageRank(B) = (1 – 0.85) + 0.85 x 0.5 = 0.575

PageRank(C) = (1 – 0.85) + 0.85 x 0.75 = 0.7875

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

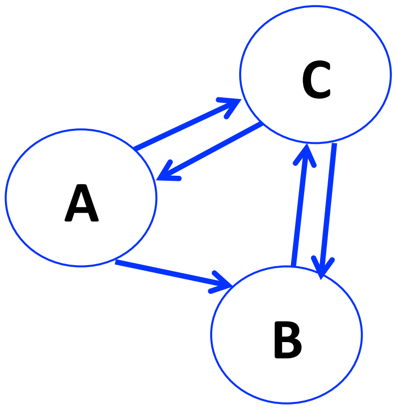
(let d = 0.85)

1st iteration:

PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

PageRank(B) = (1 – 0.85) + 0.85 x 0.5 = 0.575

PageRank(C) = (1 – 0.85) + 0.85 x 0.75 = 0.7875



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)

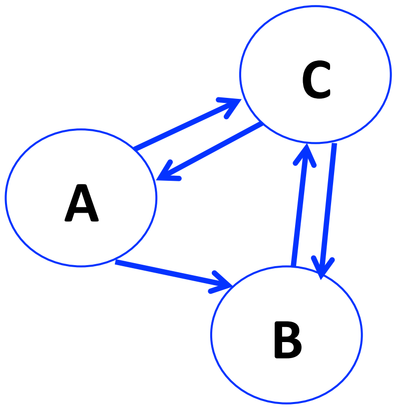
1st iteration:

PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

PageRank(B) = (1 – 0.85) + 0.85 x 0.5 = 0.575

PageRank(C) = (1 – 0.85) + 0.85 x 0.75 = 0.7875

2nd iteration: repeat using the PageRank scores of the 1st iteration as StartScores



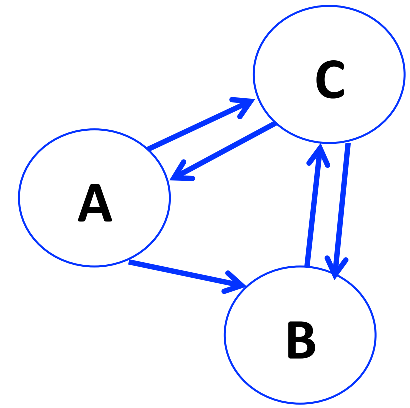
PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

1^{st} iteration:

PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

PageRank(B) = (1 – 0.85) + 0.85 x 0.5 = 0.575

PageRank(C) = (1 – 0.85) + 0.85 x 0.75 = 0.7875

2^{nd} iteration: repeat using the PageRank scores of the 1^{st} iteration as StartScores

PageRank(A) = (1 – 0.85) + 0.85 x (0.7875/2) ≈ 0.5

PageRank(B) = (1 – 0.85) + 0.85 x (0.3625/2 + 0.7875/2) ≈ 0.6

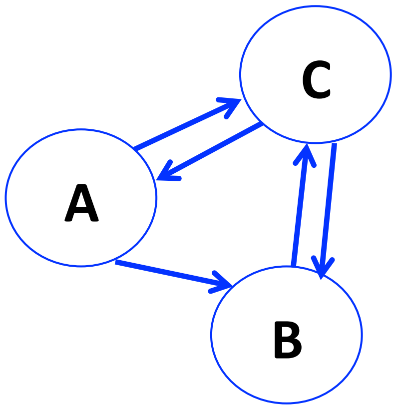
PageRank(C) = (1 – 0.85) + 0.85 x (0.3625/2 + 0.575/1) ≈ 0.8

EndScore (A) =

Σ _(A's IN) (StartScore of IN / # OUT of IN)

PageRank(A) = **(1 – d) + d x EndScore(A)**

(let d = 0.85)



PAGE	IN	OUT
A	1	2
B	2	1
C	2	2

1st iteration:

PageRank(A) = (1 – 0.85) + 0.85 x 0.25 = 0.3625

PageRank(B) = (1 – 0.85) + 0.85 x 0.5 = 0.575

PageRank(C) = (1 – 0.85) + 0.85 x 0.75 = 0.7875

2nd iteration: repeat using the PageRank scores of the 1st iteration as StartScores

PageRank(A) = (1 – 0.85) + 0.85 x (0.7875/2) ≈ 0.5

PageRank(B) = (1 – 0.85) + 0.85 x (0.3625/2 + 0.7875/2) ≈ 0.6

PageRank(C) = (1 – 0.85) + 0.85 x (0.3625/2 + 0.575/1) ≈ 0.8

3rd iteration: repeat using the PageRank scores of the 2nd iteration as StartScores

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

$PR(p_x)$: pagerank score of webpage p_x

d: damping factor

N: total number of webpages

$M(p_i)$: set of p_i's inlinks

$L(p_j)$: cardinality of the set of p_j's outlinks

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

$PR(p_x)$: pagerank score of webpage p_x

d: damping factor

N: total number of webpages

$M(p_i)$: set of p_i's inlinks

$L(p_j)$: cardinality of the set of p_j's outlinks

After 100s iterations, score of webpage quality:

- How many webpages point to it
- How good those webpages are themselves

Recap today's lecture

(1) Evaluation of crowdsourced tasks:

- Worker consistency w.r.t. task completion time
- Reproducibility
- "Error rate": precision, recall, accuracy, sensitivity, specificity

(2) Indirect Collective Intelligence: PageRank

Komarov, et al. 2013. "Crowdsourcing Performance Evaluations of User Interfaces." In proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, April 27-May 2, 2013.

Page et al. 1999. "The PageRank Citation Ranking: Bringing Order to the Web." Technical report, Stanford Infolab