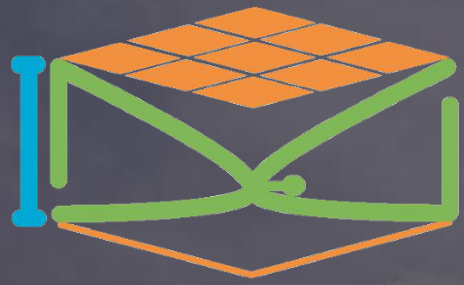


InDeX Lab
Innovative
Data
eXploration
Laboratory

UIC

Masking through Cherry-picked Data Presentation

Abolfazl Asudeh
University of Illinois Chicago
04/07/2025

“A lie that is half-truth is the darkest of all lies.”

Alfred, Lord Tennyson

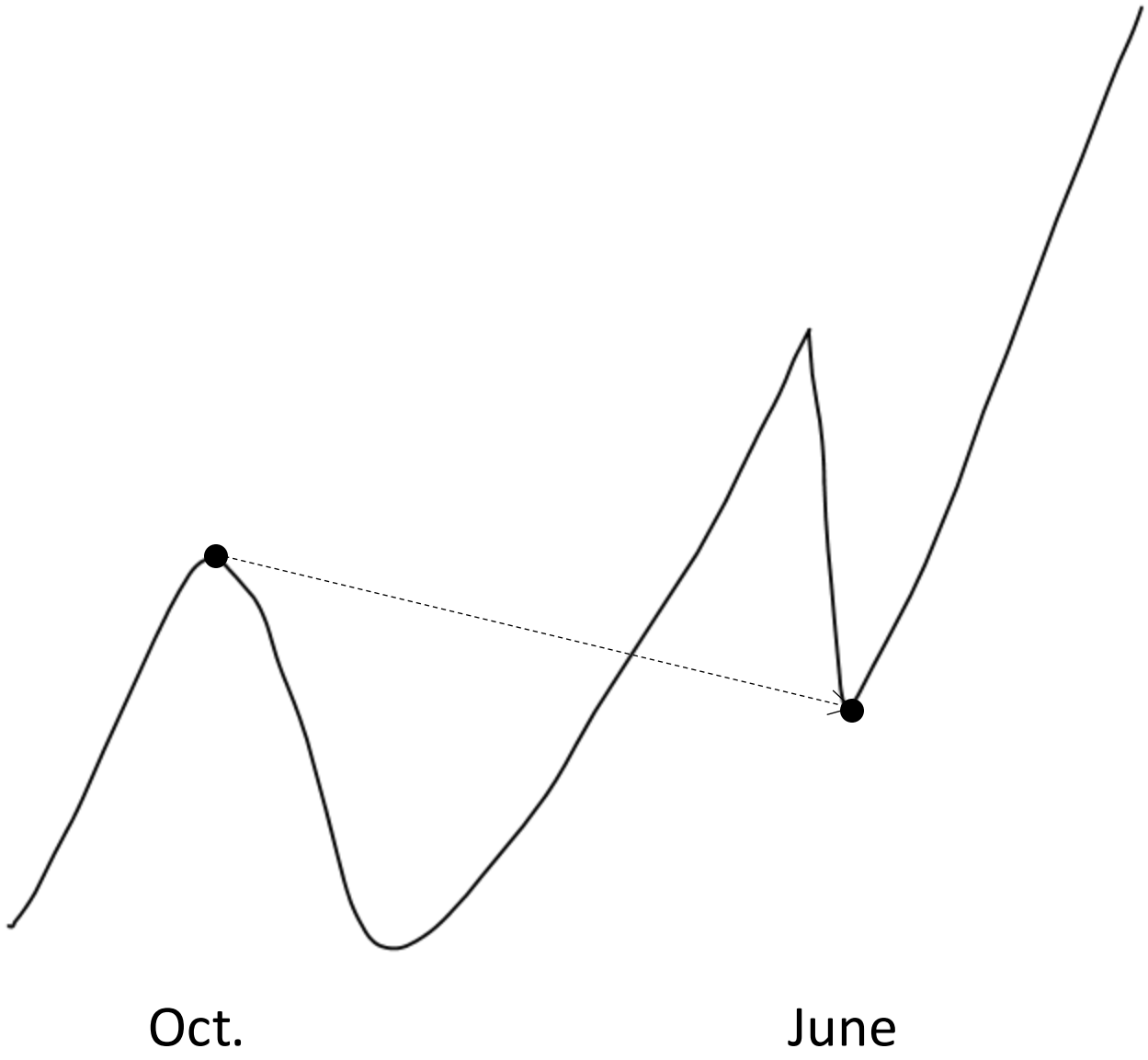
Motivation

- When a statement is made or justified based on one possible selection (e.g., one fact) among a collection of valid alternatives,

- One can cherry-pick the selection to provide a misleading statement.

A Toy Example

- Unemployment rate has reduced by 5% from Oct. to June, which shows our employment policies have been effective



- Cherry-picking has a long history and hence many different forms.

- In a nice article at PolitiFact, L. Jacobson goes over some of the examples of cherry-picking in US politics.
  - PolitiFact reported cherry-picking ``***hundreds of times***'' in their fact-checks.

- What should we do?
  1. Detect Cherry-picking
  2. Provide Unbiased Alternatives
- Lecture Outline: Cherry-picking in
  - News ordering
  - Trendlines (political statements)
  - Ranking functions

Part I: Cherry-picking in New Ordering

Neutrality in News Ordering, in KDD'24

Rishi Advani[1], Paolo Papotti[2], Abolfazl Asudeh[1]
[1]University of Illinois Chicago, [2]EURECOM

Media’s impact on public opinion is undeniable

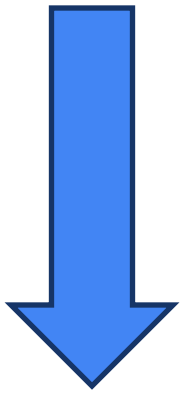

NEWS
World
Business
Tech
Sport


FAKE NEWS

There are more subtle ways than Fake News to deceive the news media's audience.

Misinformation can be very effective also with "Real News”

- Immigration rates on the rise again



- Crime rates in major cities reach historic highs

- Immigration rates on the rise again
- Scientists discover new plant species
- Folding bikes: latest fad among millennials
- Turkeys win championship for third year in a row
- Crime rates in major cities reach historic highs

Just by changing the *ordering* of a set of news headlines…

…we can influence a reader's perception of them!

If we want to combat misinformation, we need to be able to efficiently

 detect cherry-picked orderings and

 compute "neutral" orderings.

definitions

opinion priming: when viewing one headline influences a user's opinion of (the story corresponding to) a second headline, by affecting their

- belief in the truthfulness of the story,
- stance (for or against) on the events in the story, or
- perception of causality between the two stories.

Pairwise Opinion Priming (POP) Function

C	t_1	t_2	t_3
t_2	0.1		
t_3	0.3	0.7	
t_4	0.2	0.8	1

definitions (cont.)

- Pairwise neutrality:
  - Input: a set of news **t**, an ordering **s**, a POP function **C**, and a decay function **D**,

$$D(d) = \begin{cases} 1 & \text{if } d = 1 \\ 0 & \text{otherwise} \end{cases}$$

(a) The decay function

C	t_1	t_2	t_3
t_2	0.1		
t_3	0.3	0.7	
t_4	0.2	0.8	1

(b) The POI function

s_1	s_2	s_3	s_4
1	4	2	3

(c) The ordering

  - Output: the likelihood of opinion priming ***not*** occurring between two headlines

$$N_{i,j} = 1 - D\big(\big|s_i - s_j\big|\big)C(t_i, t_j)$$

N	t_1	t_2	t_3
t_2	1		
t_3	0.7	1	
t_4	1	0.2	0

(d) Pairwise Neutrality values

definitions (cont.)

The **neutrality** of a news ordering is computed by taking the pairwise neutrality of all pairs of adjacent headlines and applying some aggregation function (e.g., AVG, MIN) over the results.

detecting cherry-picked orderings

detection: algorithm

Suppose we have a news ordering **s** with neutrality *X*.

If the average neutrality over all possible orderings is far from *X*, then s was likely cherry-picked.

detection: algorithm (cont.)

It is impractical to compute the average over all possible orderings, so we take the average over a sample of r random orderings.

By the Saw-Yang-Mo inequality, we can bound the probability of a random ordering having neutrality X.

This algorithm takes $O(rn)$ time.

maximizing neutrality

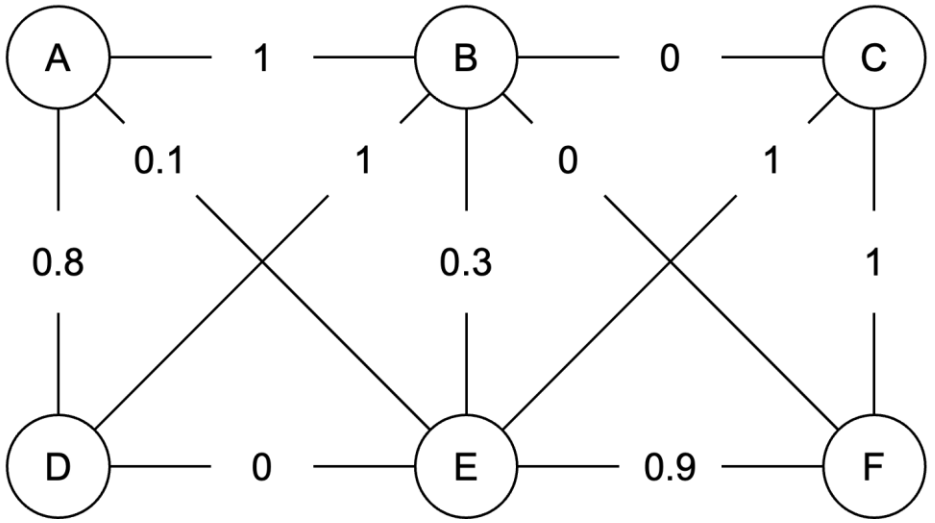
maximizing neutrality: assumptions

We will focus on the average function as our aggregation function. In our paper, we study the minimum function as well.

We will also focus on two of our algorithms.

Graph representation

- each headline is a vertex
- each pair of headlines t_i and t_j is connected by an edge with weight $N_{i,j}$



Maximizing the neutrality of a news ordering is equivalent to finding a Hamiltonian path with maximum weight in a complete graph – which we call PathMaxTSP (NP-hard).

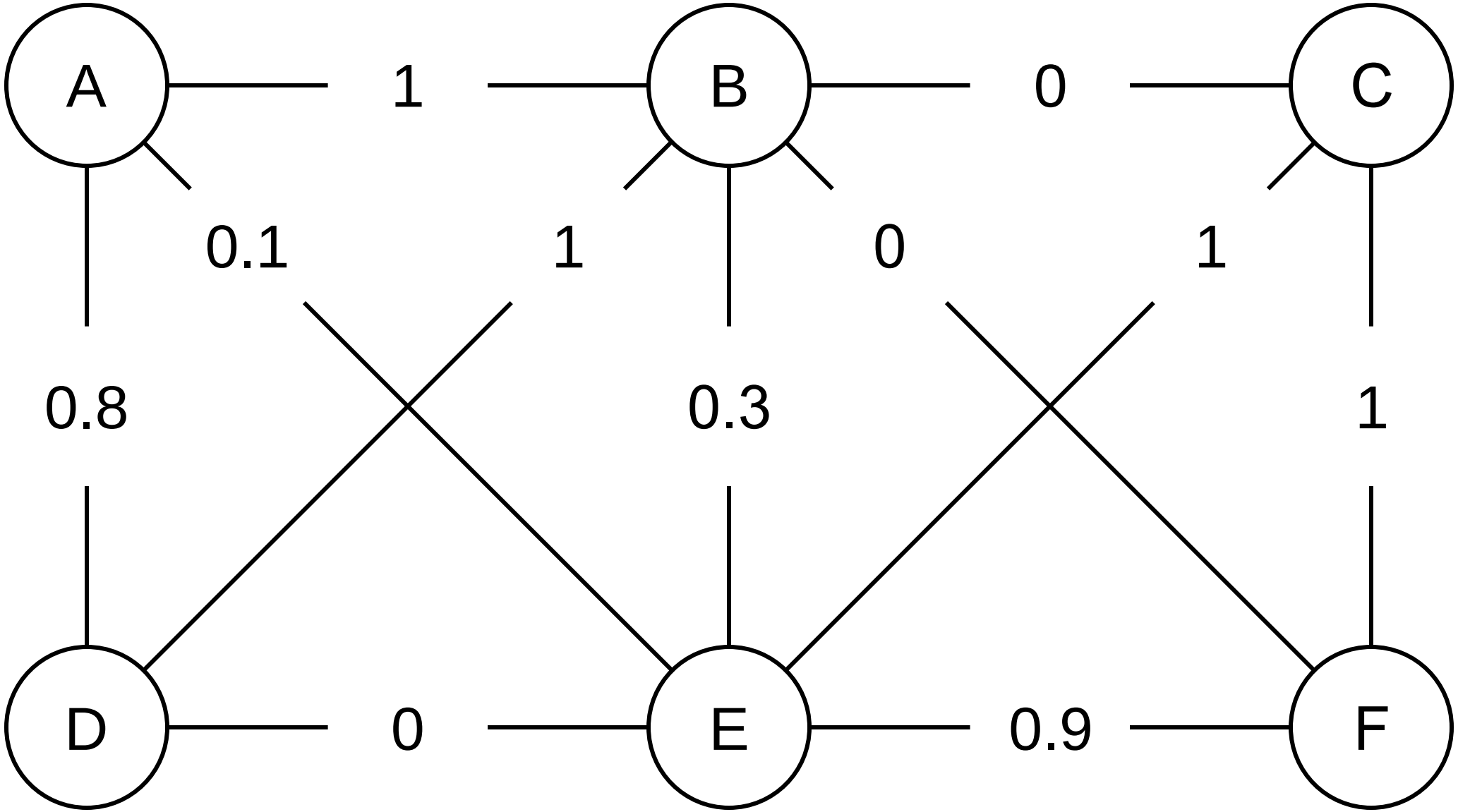
Hamiltonian path: a path that includes each vertex exactly once.

maximizing neutrality: first approximation algorithm

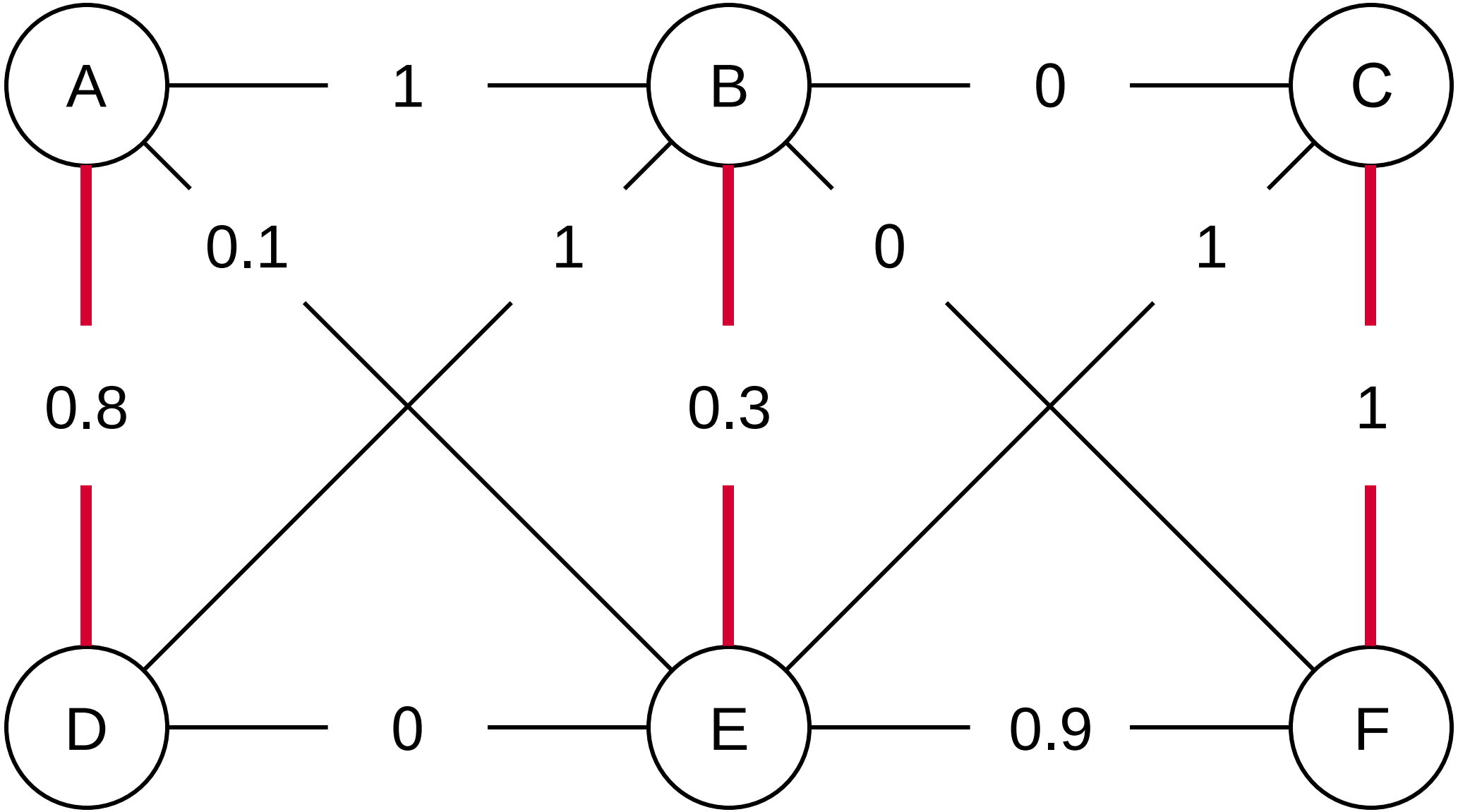
- find a maximum-weight matching (the set of (disjoint) node-pairs with max sum of weights)
- replace each edge with a "super node"
- repeat until there is only one super node left

- approximation factor: ½
- runtime: $O(n^4)$

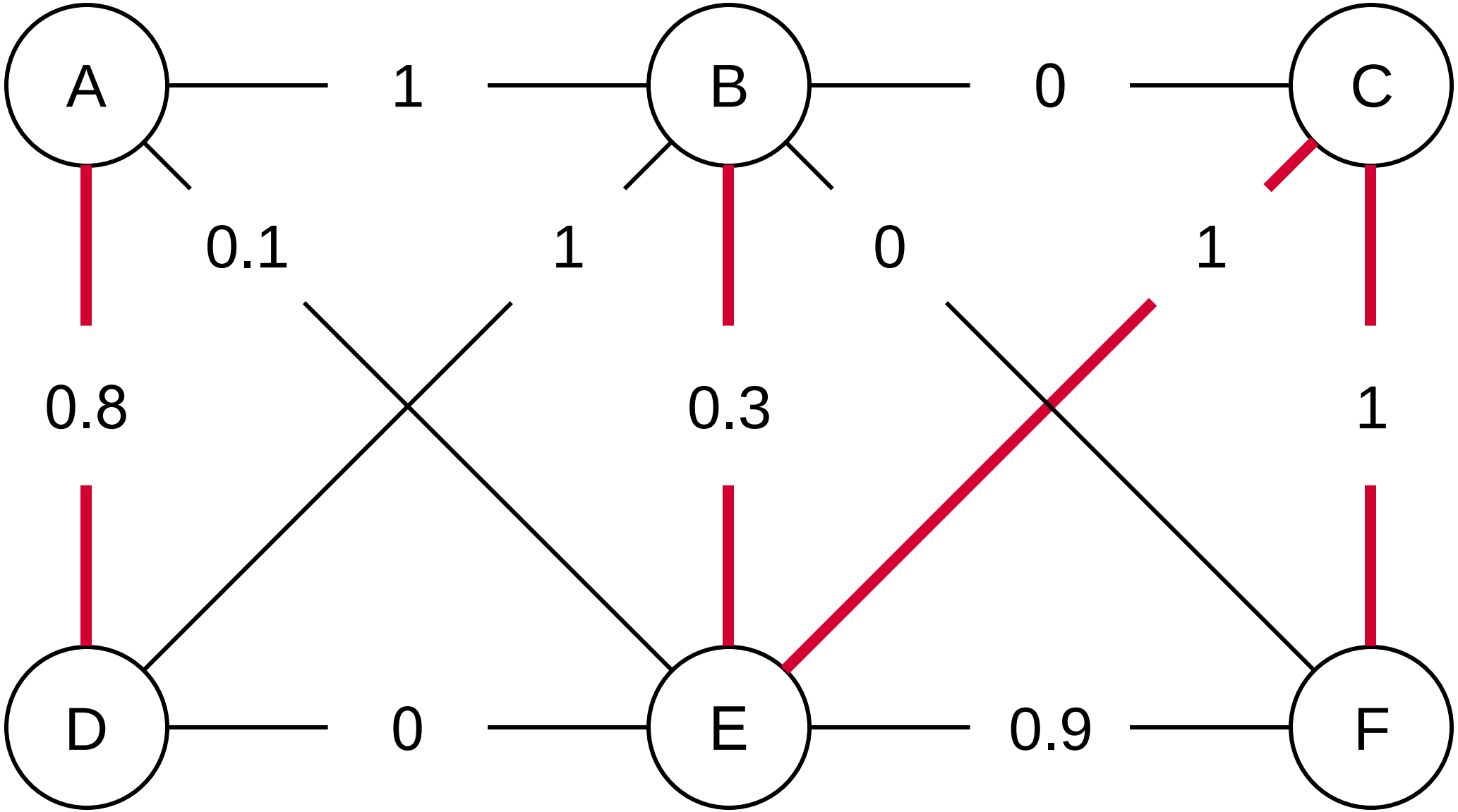
maximizing neutrality: example



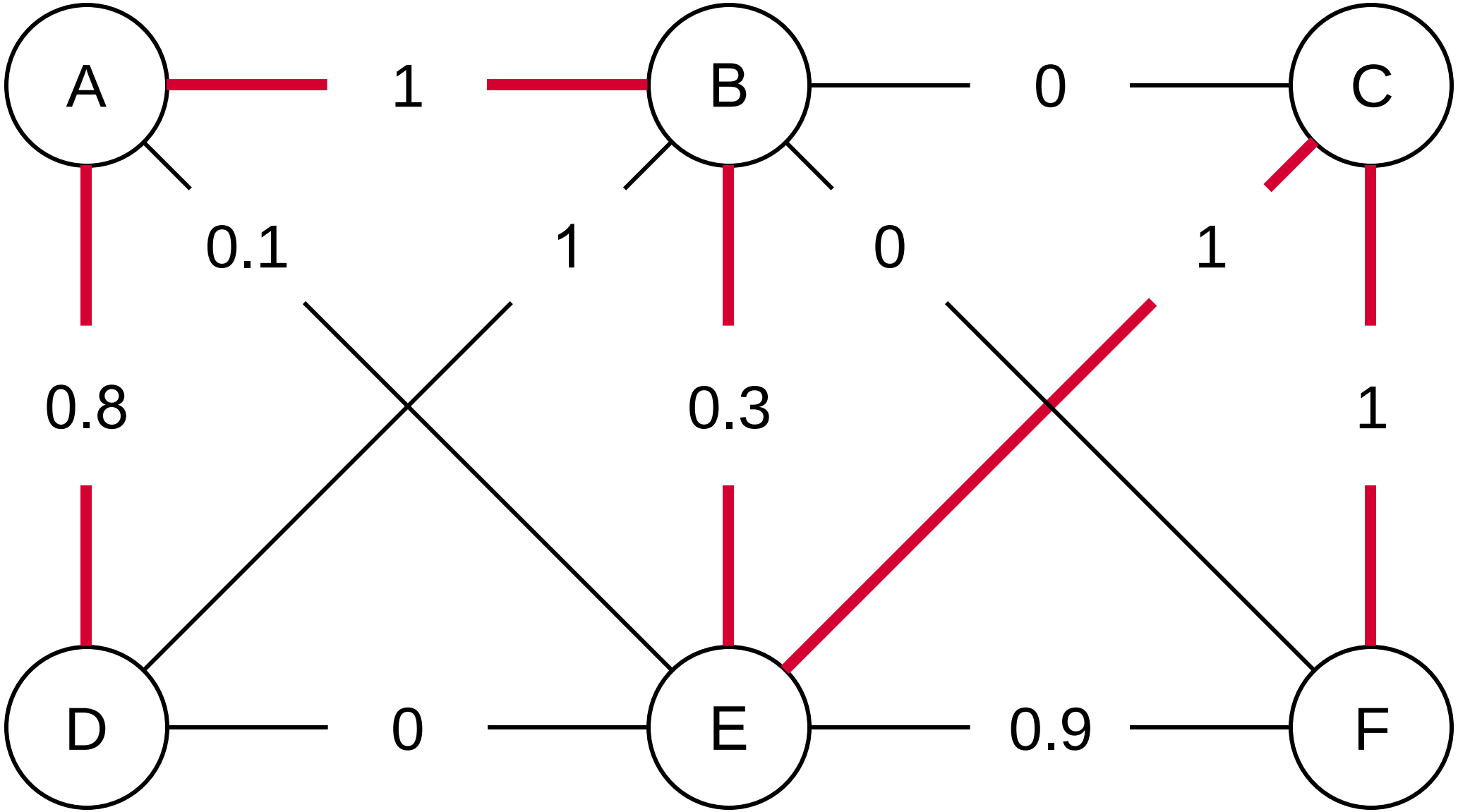
maximizing neutrality: example



maximizing neutrality: example



AVG Neutrality = 4.1/5 = 0.82

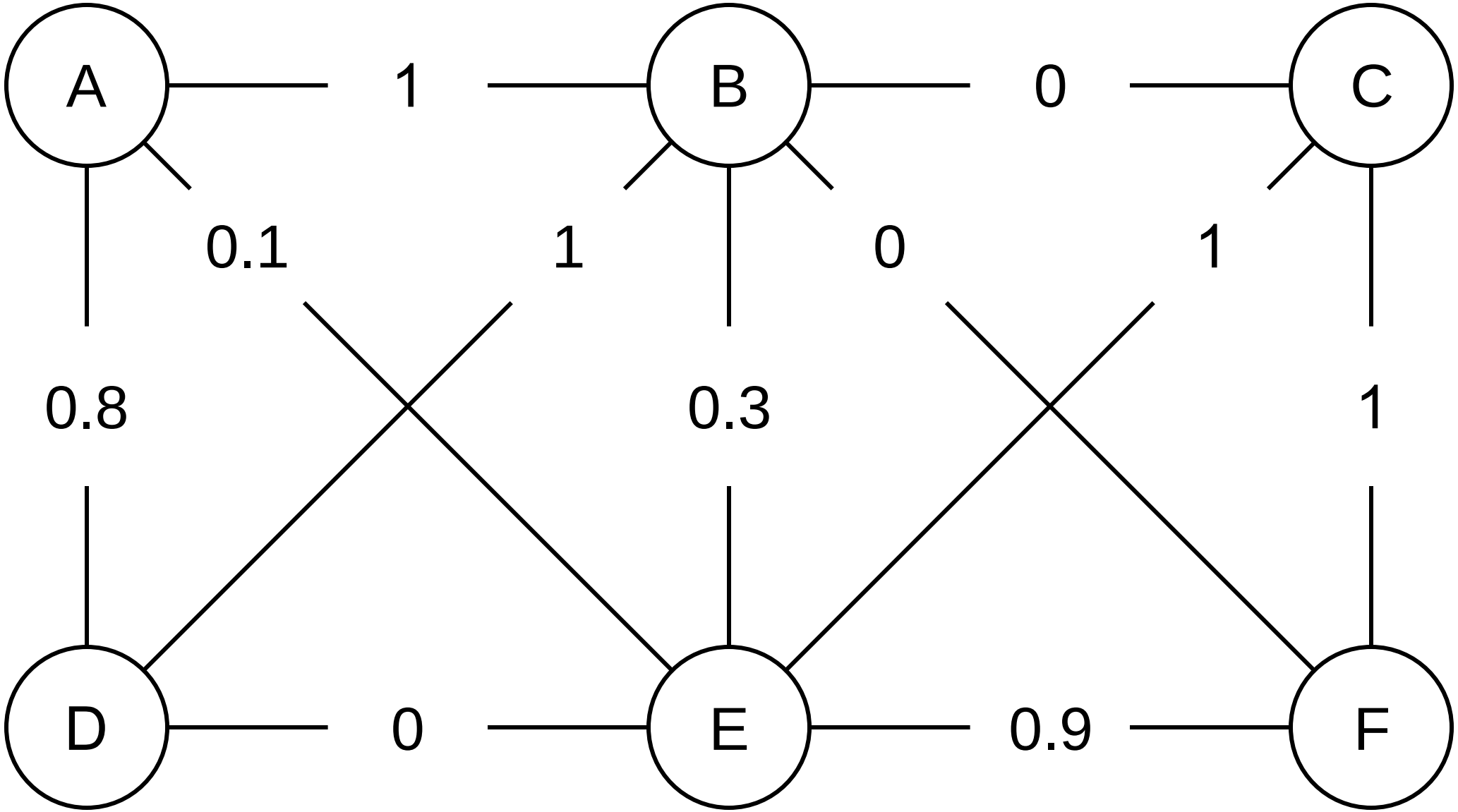


maximizing neutrality: second approximation algorithm

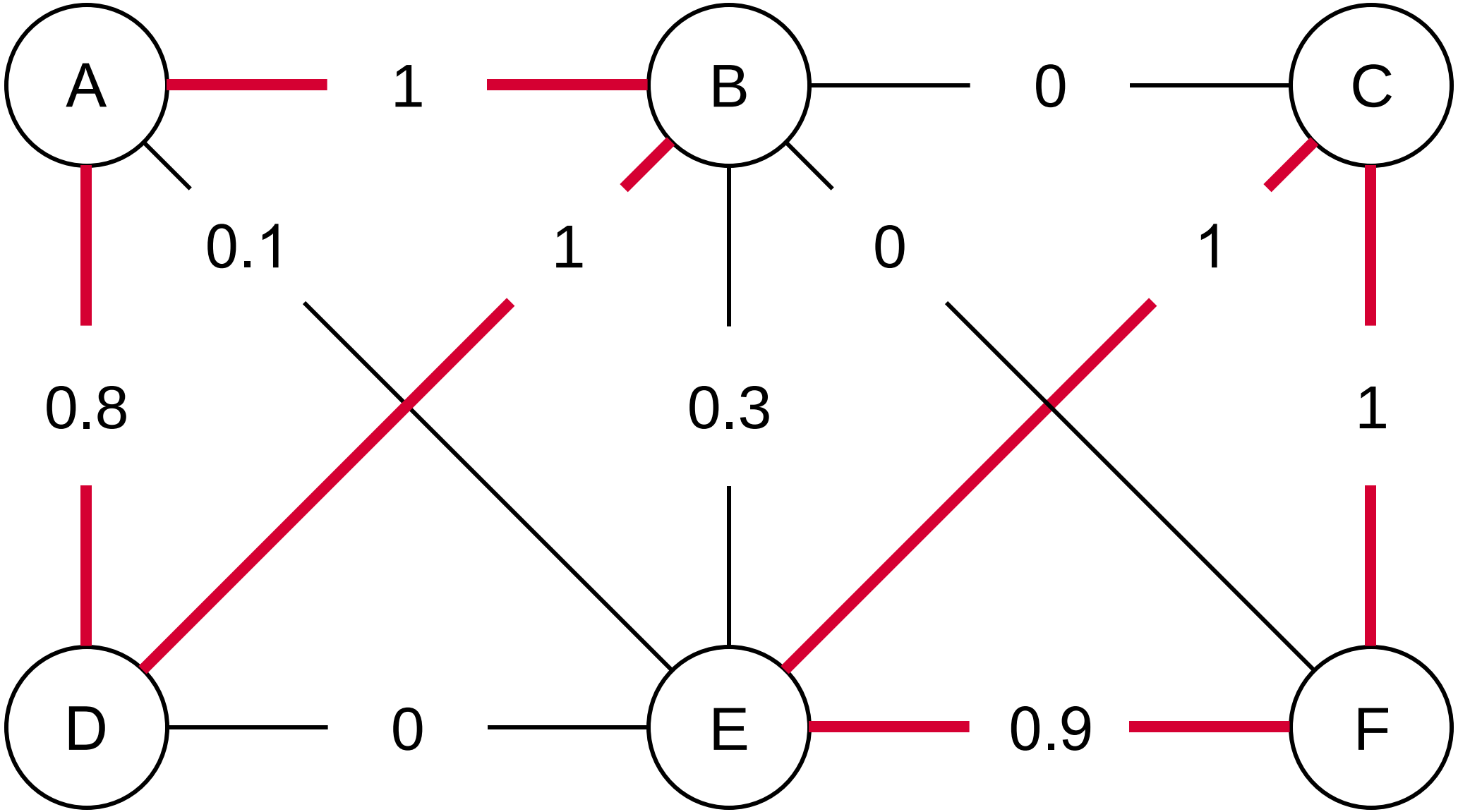
- find a maximum cycle cover (a set of cycles that cover all edges and have the total max sum)
- Convert each cycle to a chain (by removing the min-weight edges)
- replace each chain with a "super node"
- repeat until there is only one super node left

- approximation factor: ½
- runtime: $O(n^3)$

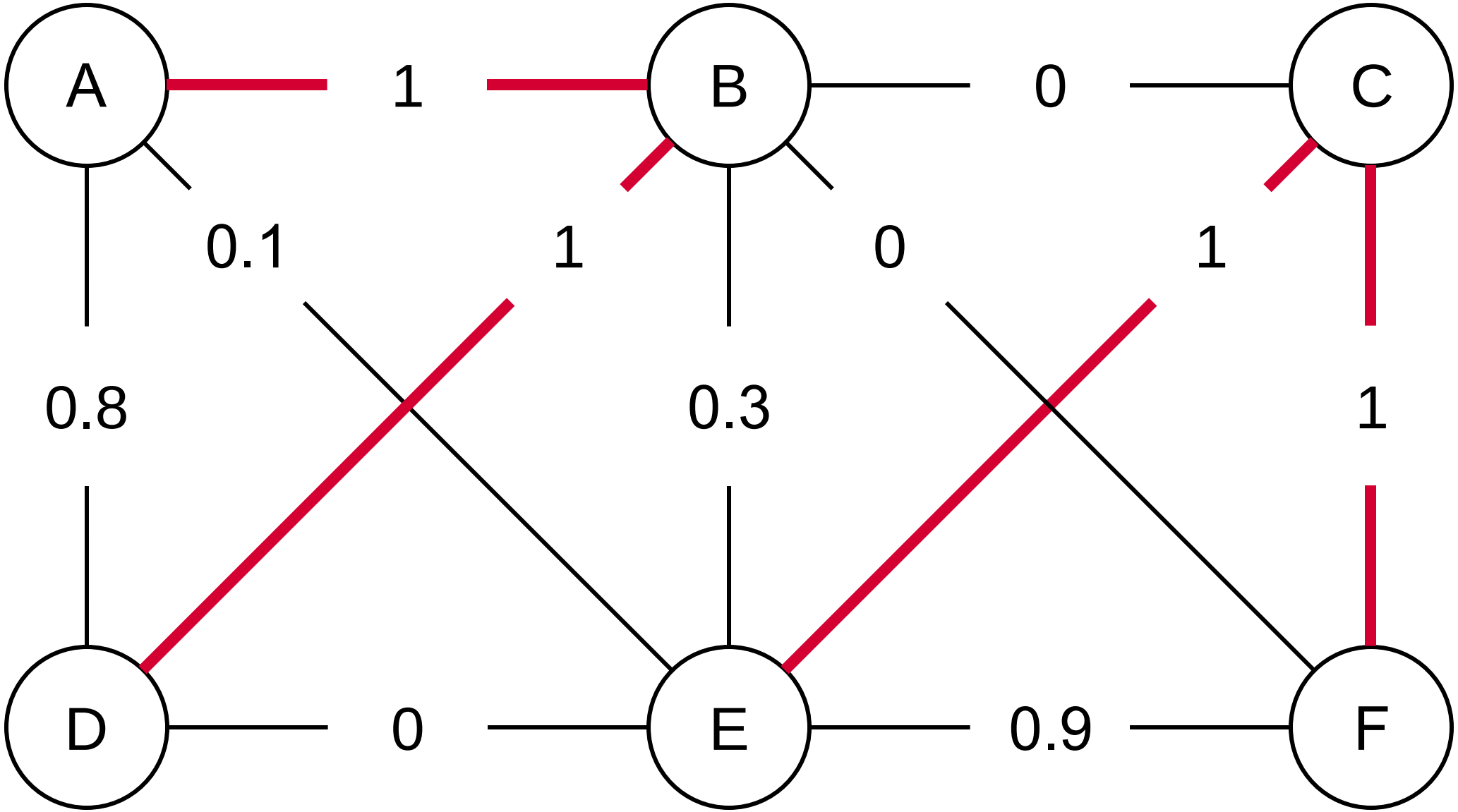
maximizing neutrality: example



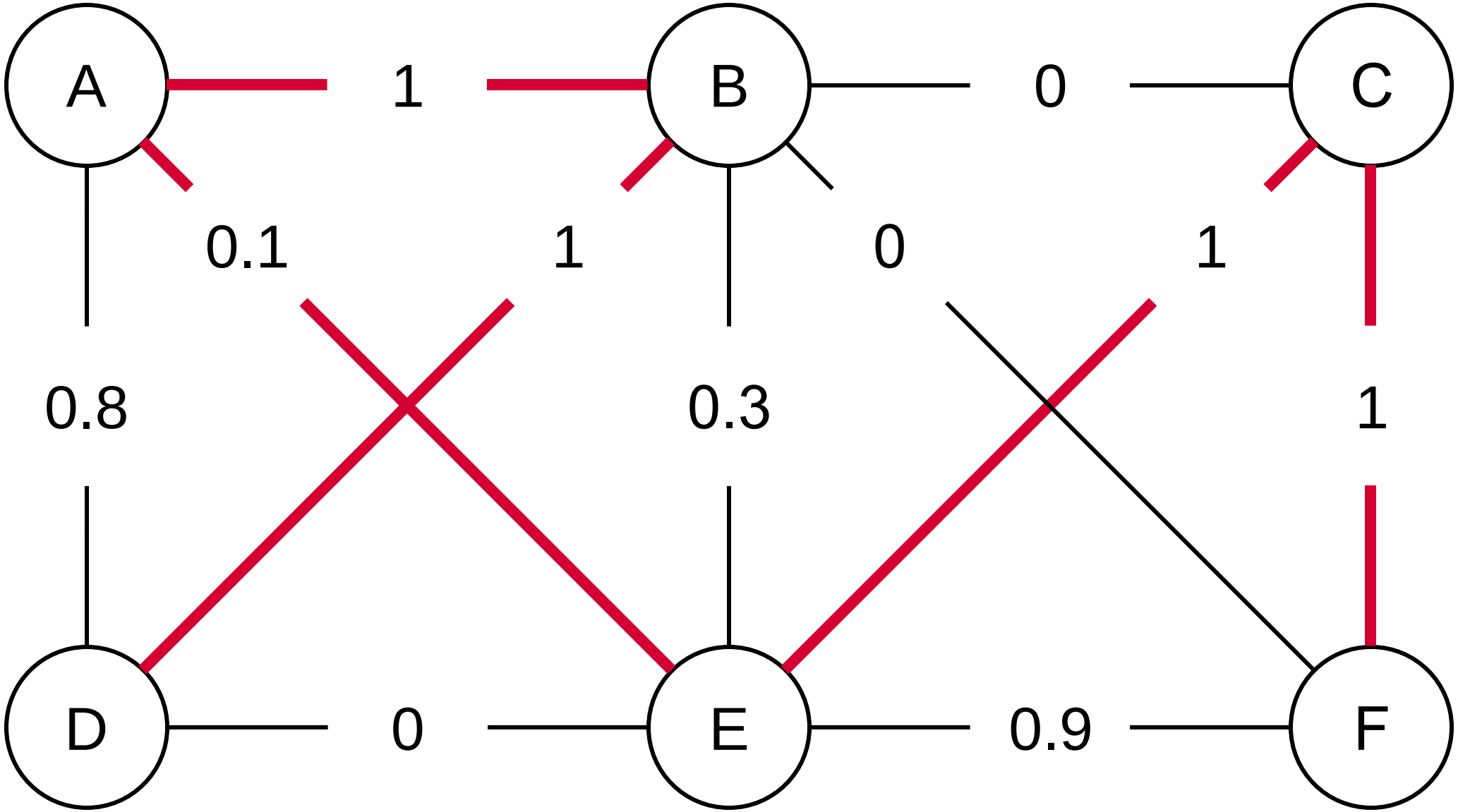
Iter 1: max cycle cover



Iter 1: convert to chain (and super-node)



Iter 2. AVG Neutrality = 4.1/5 = 0.82



user study on existence of opinion priming

- 9 fictional news headlines, including the following two headlines:
  - "City's high school graduation rates at lowest in decades"
  - "High school principal celebrates 10 years"
- 53 participants
  - test group had the two headlines placed together
  - control group had them separated

user study on existence of opinion priming (cont.)

After the participants read the headlines, they were asked their impression of the principal:

- 39% of the test group had a negative impression
- 16% of the control group had a negative impression

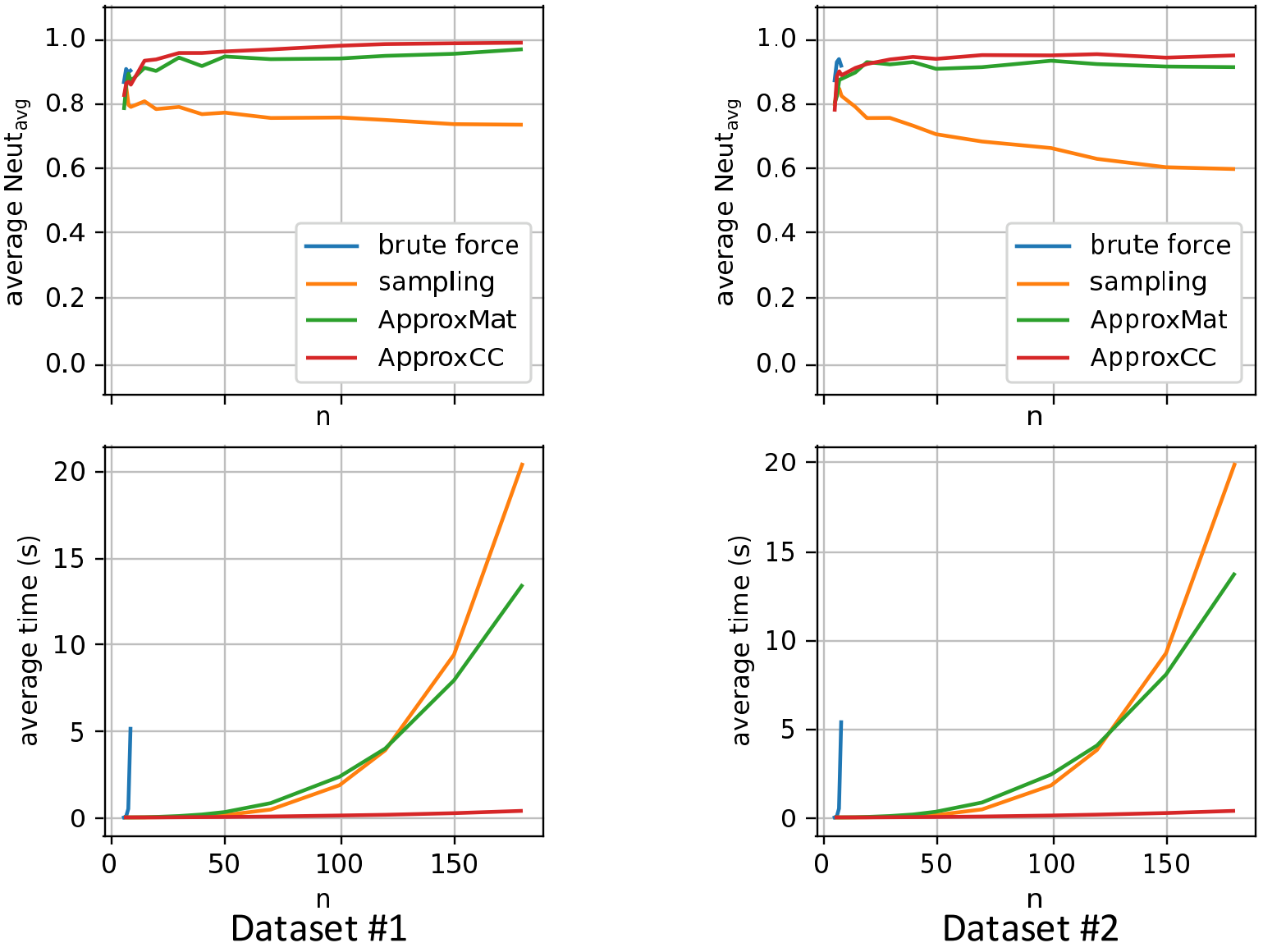
This difference is statistically significant (Boschloo's exact test, p=0.0337).

maximizing neutrality: experiment setup

two larger datasets based on the real data used for cherry-picking detection:

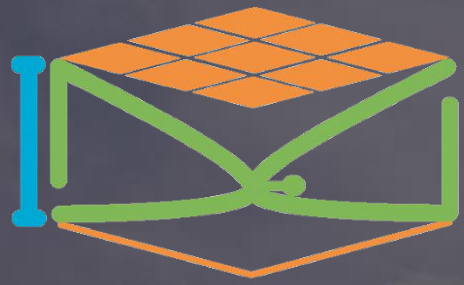
- Dataset #1: edge weights follow same distribution as real data
- Dataset #2: edge weights from Dataset #1 are forced to satisfy a variant of triadic closure
  - if A–B and B–C have low weight, then A–C must have low weight

maximizing neutrality: experiment results



Future Work

- scalable computation of POP function
  - Crowdsourcing ?
  - **LLMs**
- introducing utility
- nonadjacent pairwise neutrality (we already do this for cherry-picking detection)
- Beyond Ordering (selection of news, frequency, …)



InDeX Lab
Innovative
Data
eXploration
Laboratory

UIC

Part II:
Cherry-picked Trendlines

On Detecting Cherry-picked Trendlines, in VLDB'20

A. Asudeh, H. V. Jagadish, You (Will) Wu, Cong Yu
UIC, University of Michigan, Google Research

Motivation

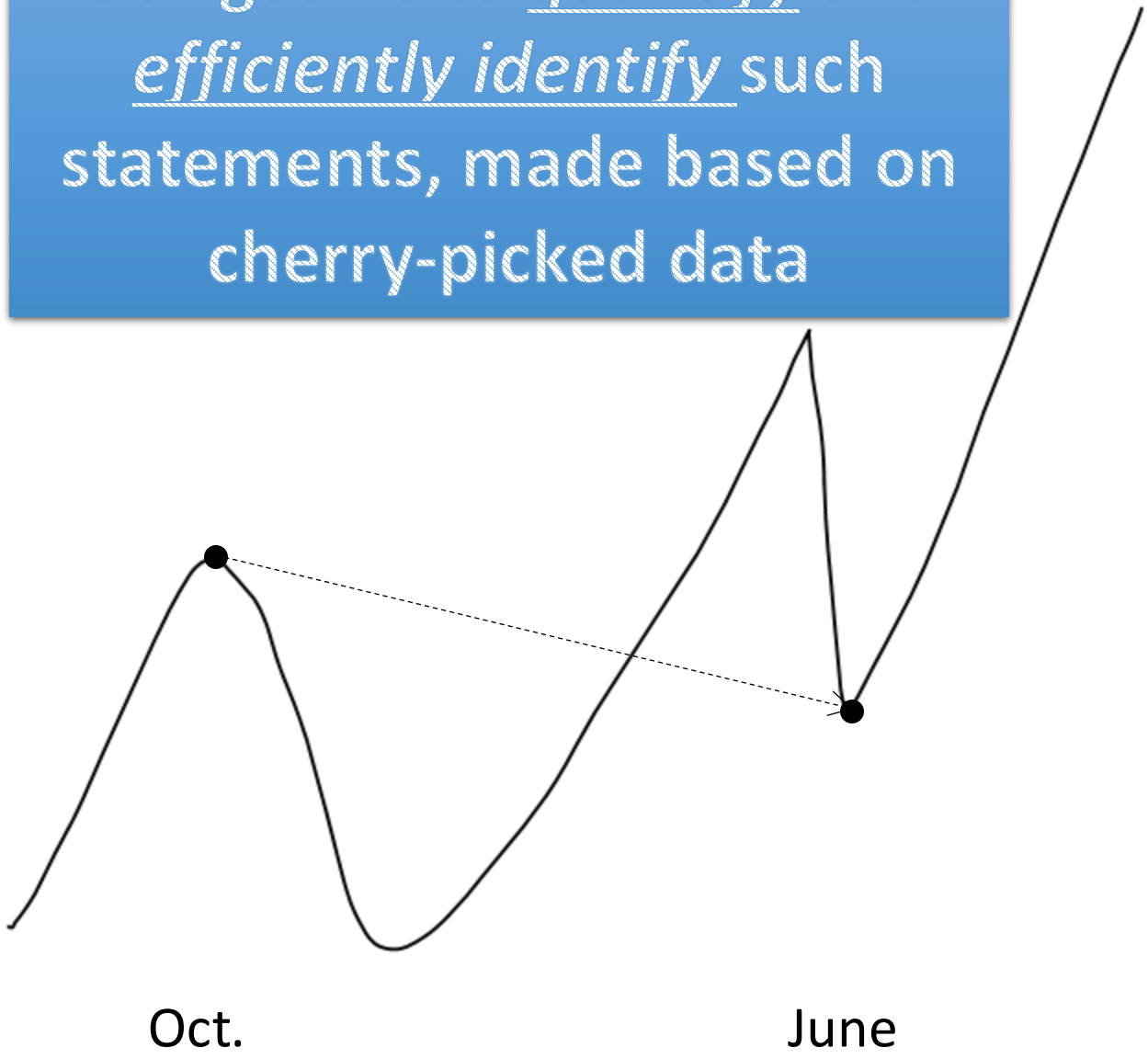
- Politicians would like not to be caught blatantly lying, so they cherry-pick the factual basis for their conclusion.

- The points based on which a *statement* is made are carefully selected to show a misleading *“trendline”* that is not a *reasonable representation* of the situation.

Toy Example

- Unemployment rate has reduced by 5% from Oct. to June, which shows our policies have been successful in reducing the unemployment rate

Our goal is to *quantify* and *efficiently identify* such statements, made based on cherry-picked data



Running Example

- It has been explained how cherry-picking short time-frames can distort the reality of global warming. The monthly climate data can be used to support the following fantasy-like claims:

- “summer was colder than winter in 2012 in the Northern Hemisphere” as, for example, the (average) temperature of Ann Arbor (MI, USA) on Aug. 18 (a summer day) was 58^F, whereas its temperature on Mar. 15 (a winter day) was 66^F.

Trendline

- A trendline θ is a defined as a pair of trend points b (the beginning) and e (the end) and their target values in the form of

$$\theta = \langle\, (b, y(b)), (e, y(e)) \,\rangle$$

  - E.g., the trendline compares the temperature of Ann Arbor on two different days.

- Constrained v.s. unconstrained trendlines

Statement

- Given a trendline θ, the statement S_θ is a range $S_\theta = (\bot, \top)$ such that
$$y(e) - y(b) \in (\bot, \top)$$

- In the running example:
  - b =Aug. 18 2012, Ann Arbor – MI, $y(b) = 58$^F
  - e =March 15 2012, Ann Arbor - MI, $y(b) = 66$^F
  - Statement: summer was colder than winter, is: $S_\theta = (0, \infty)$ which is satisfied by θ since

$$y(e) - y(b) = 66 - 58 > 0$$

Support Model

- Observation: if a statement is not based on a cherrypicked trendline, other data points should also ***support*** it.
  - cherry-picked trendlines are carefully selected and, therefore, significantly change by slightly changing the trend points.
  - In the running example, perturbing the beginning and/or the end points of the chosen dates by even a few days results in trendlines that do not support the statement.
- Consider a neighborhood around the selected trendline, called Support Region
- Support of a statement: The ratio of the "valid" trendlines in the support region for which their target value difference remains within the acceptable range.

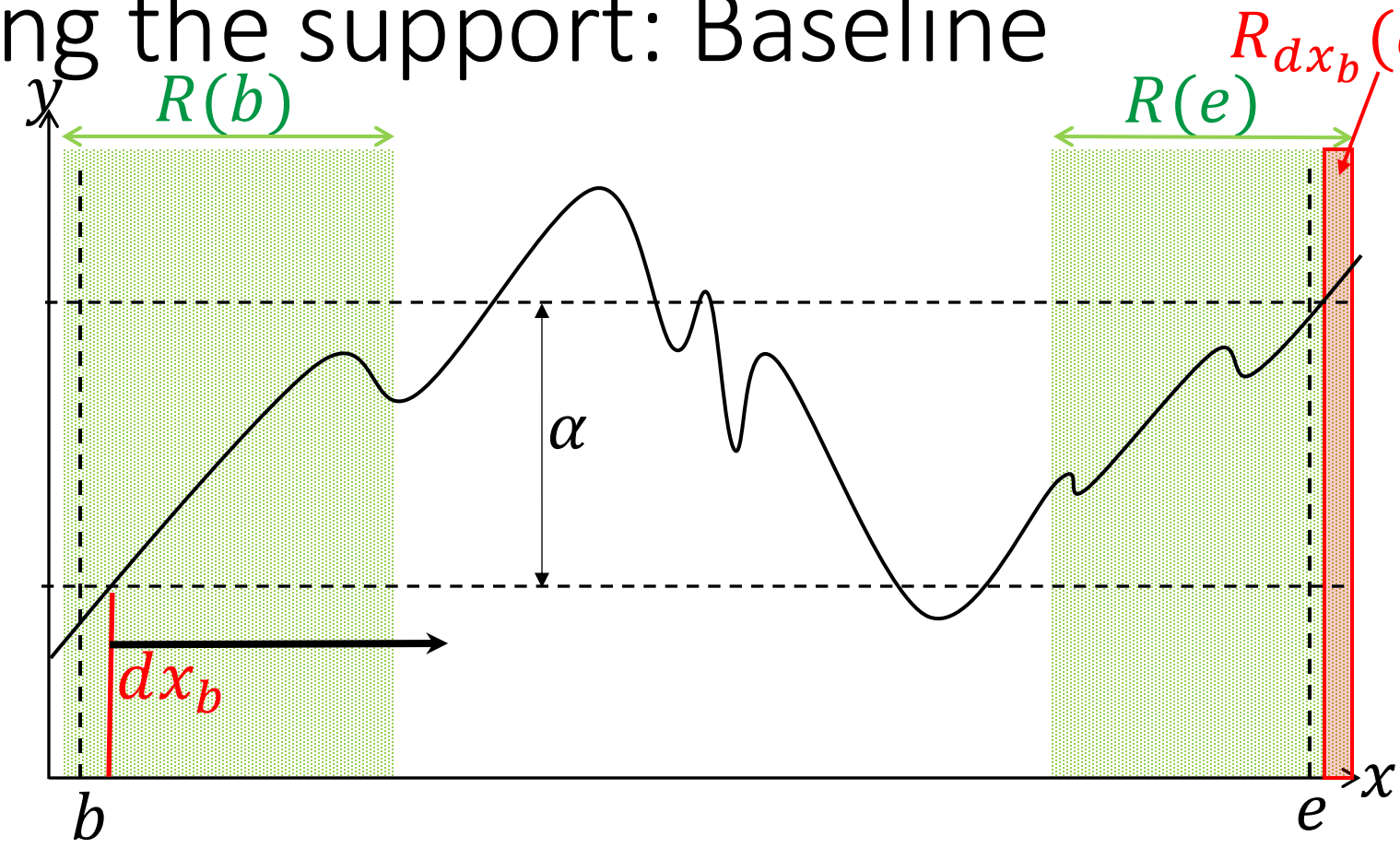
$$\omega(S, R_S, \mathcal{D}) = \frac{vol(\{valid\ \langle p \in R(b), p' \in R(e)\rangle \mid y(p') - y(p) \in (\bot, \top)\})}{vol(\{valid\ \langle p, p'\rangle \mid p \in R(b), p' \in R(e)\})}$$

Problem Formulations

1. Compute the support of an statement
2. Find the most supported statement for a given range
3. Find the tightest statement for a given support value

Computing the support: Baseline

$O(n^2)$



Efficient Algorithm

O(n logn)

- For every point $d_i \in b$, define w_i as the number of points in b for which $y(d_i) - y(d_j) \in (\perp, \top)$. Then support of a statement can be computed as

$$\sum_{\forall d_i \in b} w[i]$$

- Design the cumulative function

$$F(y) = |\{dx \in R(e) \mid y(dx) < y\}|$$

Sort O(n log n)

- Using *F*,

$$w[i] = F\big(y(dx[i]) + \top\big) - F\big(y(dx[i]) + \perp\big)$$

Binary search O(log n)

Randomized Algorithms

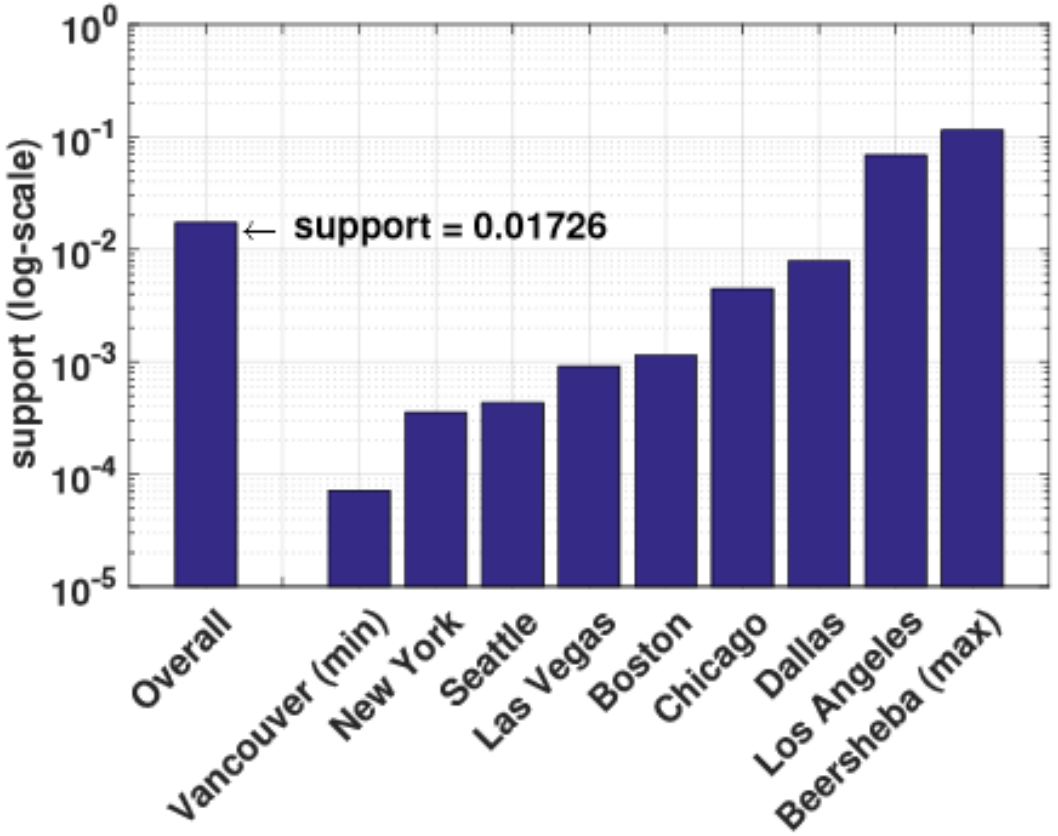
- Based on Monte-Carlo Estimation

1. Pair Sampling

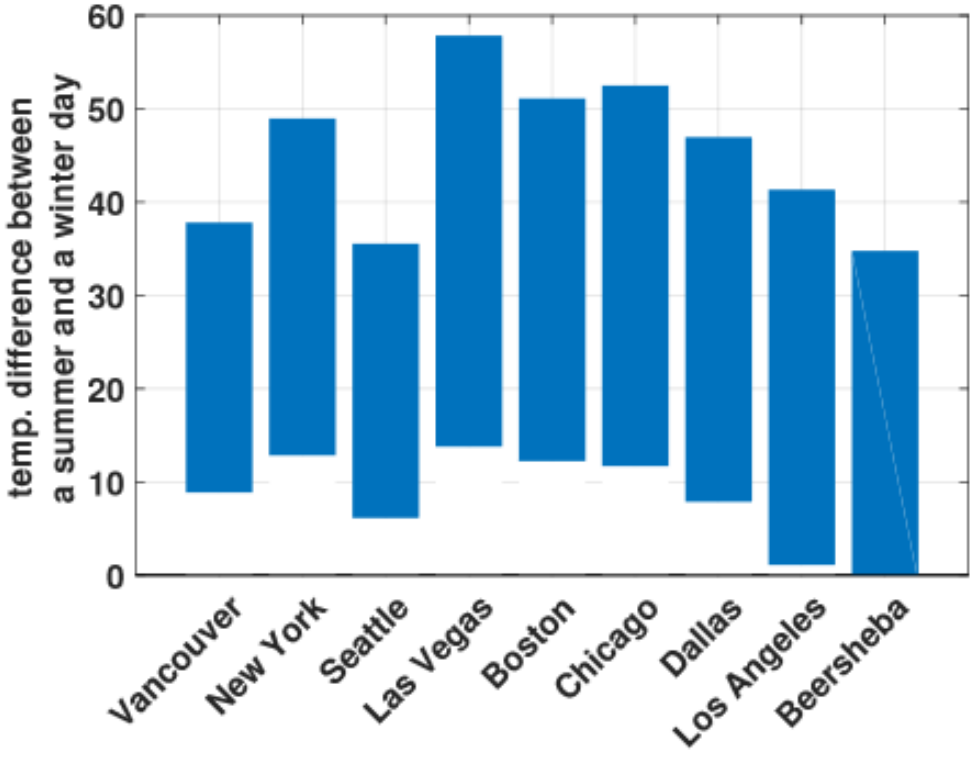
2. Point Sampling ← practical solution

Experiments, Proof of Concept (running example)

Support of (winter colder than summer)

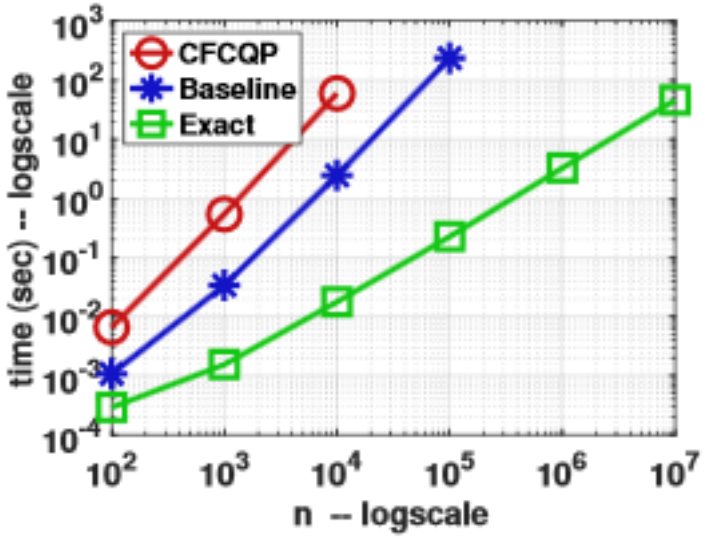


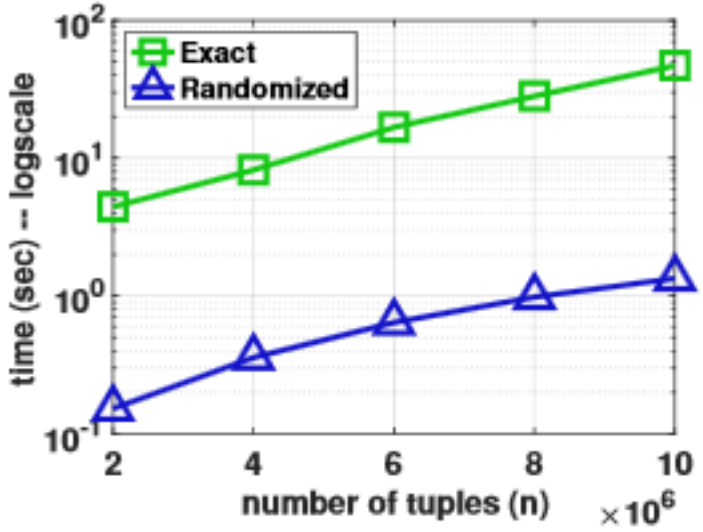
Tightest Statement with support 0.8



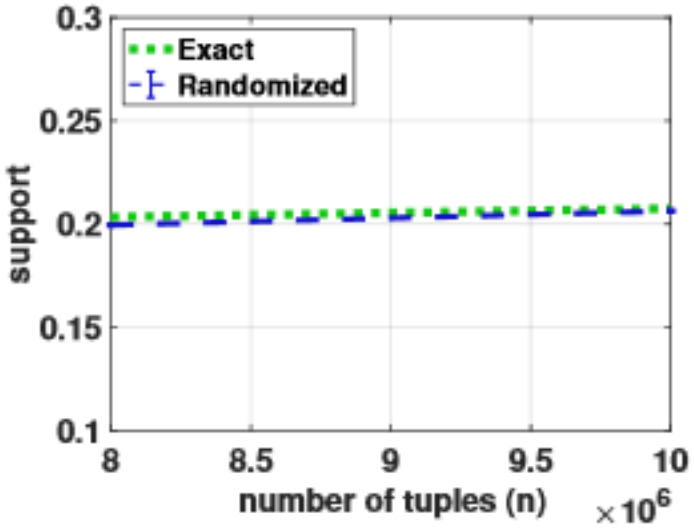
Performance Evaluation

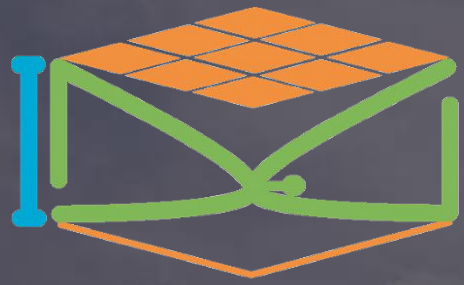
Exact





Randomized





InDeX Lab
Innovative
Data
eXploration
Laboratory

UIC

Part III:
Cherry-picking Ranking Functions:

On Obtaining Stable Rankings, in VLDB'19

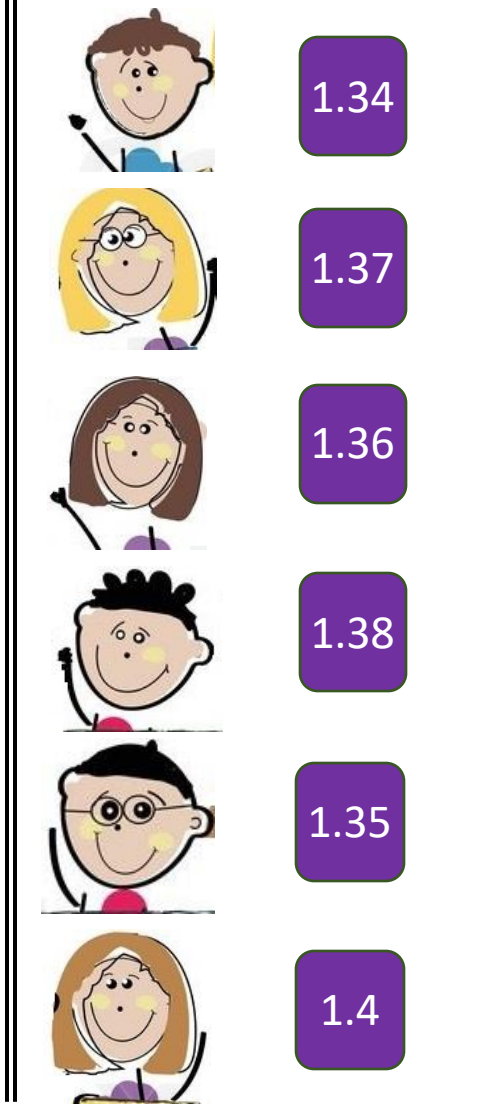
A. Asudeh, H V Jagadish, G. Miklau, J. Stoyanivich
UIC, University of Michigan, Umass, NYU

Toy Example

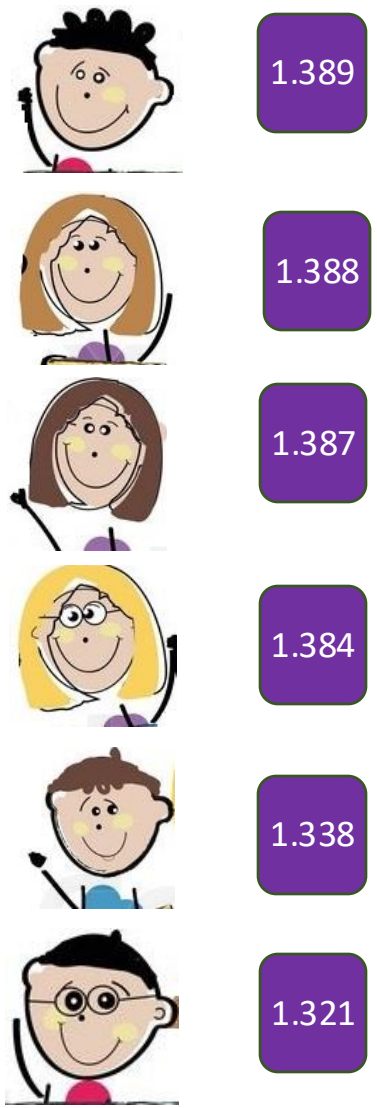
Sale -- Normalized

Customer Satisfaction -- Normalized

1 + 1



1.11 + 0.9



$$\sum w_i x_i$$

$$w_1 + w_2$$

Despite the potential impact of these weights, those are

chosen in an ad-hoc manner!

THE ORDER OF THINGS

What college rankings really tell us.

By Malcolm Gladwell

- "It is easy to see why the U.S. News rankings are so popular. A single score allows us to judge between entities"
- "Rankings depend on what weights we give to what variables"
- "This idea of using the rankings as a benchmark, college presidents setting a goal of 'We're going to rise in the *U.S. News* ranking' …"

Rankings depend on what weight we give to what variables.

Illustration by SEYMOUR CHWAST

Stability: how robust the output is

- Small changes in weights change the output?
  - Cherry-picked?
  - Decisions based on which are questionable

- Stability: The (volume) Ratio of functions that generate an output (ranking, top-k, or partial ranking)

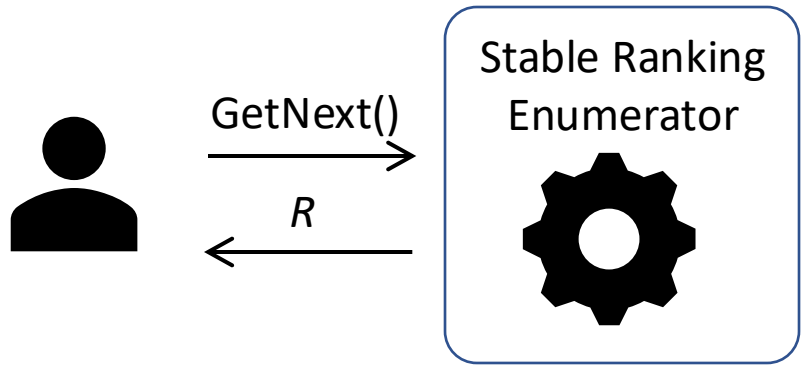
Region of Interest

- The range of functions that are “acceptable” for the ranking designer
  - *e.g. functions with* at least 95% cosine similarity with a ref. vector

High level idea

- *Stability Verification*

- *Stability Enumeration*
  - *GetNext*: An iterative process that generate stable regions one after the other
  - It enables finding (any number of) top-x stable rankings (or top-k)

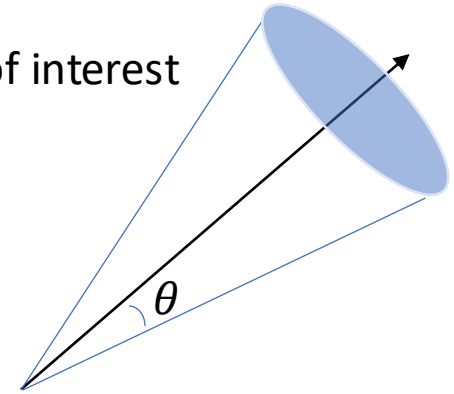


Technical Details

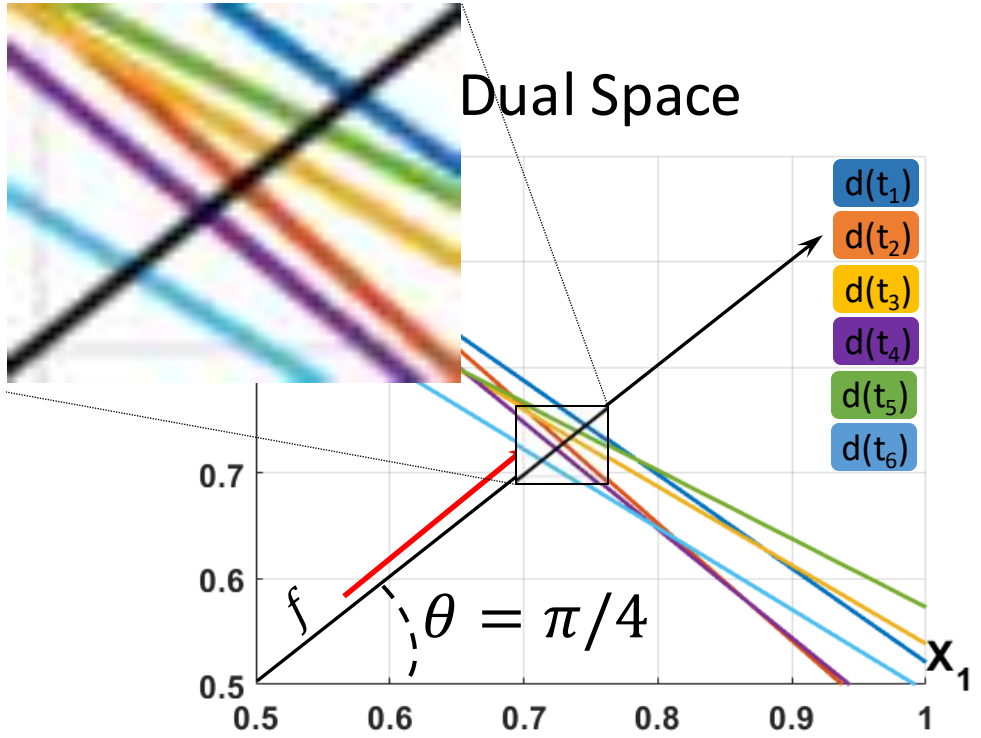
Dual Space

$\mathcal{D}$			f
id	x_1	x_2	$x_1 + x_2$
t_1	0.63	0.71	1.34
t_2	0.72	0.65	1.37
t_3	0.58	0.78	1.36
t_4	0.7	0.68	1.38
t_5	0.53	0.82	1.35
t_6	0.61	0.79	1.4

Region of interest



Dual Space



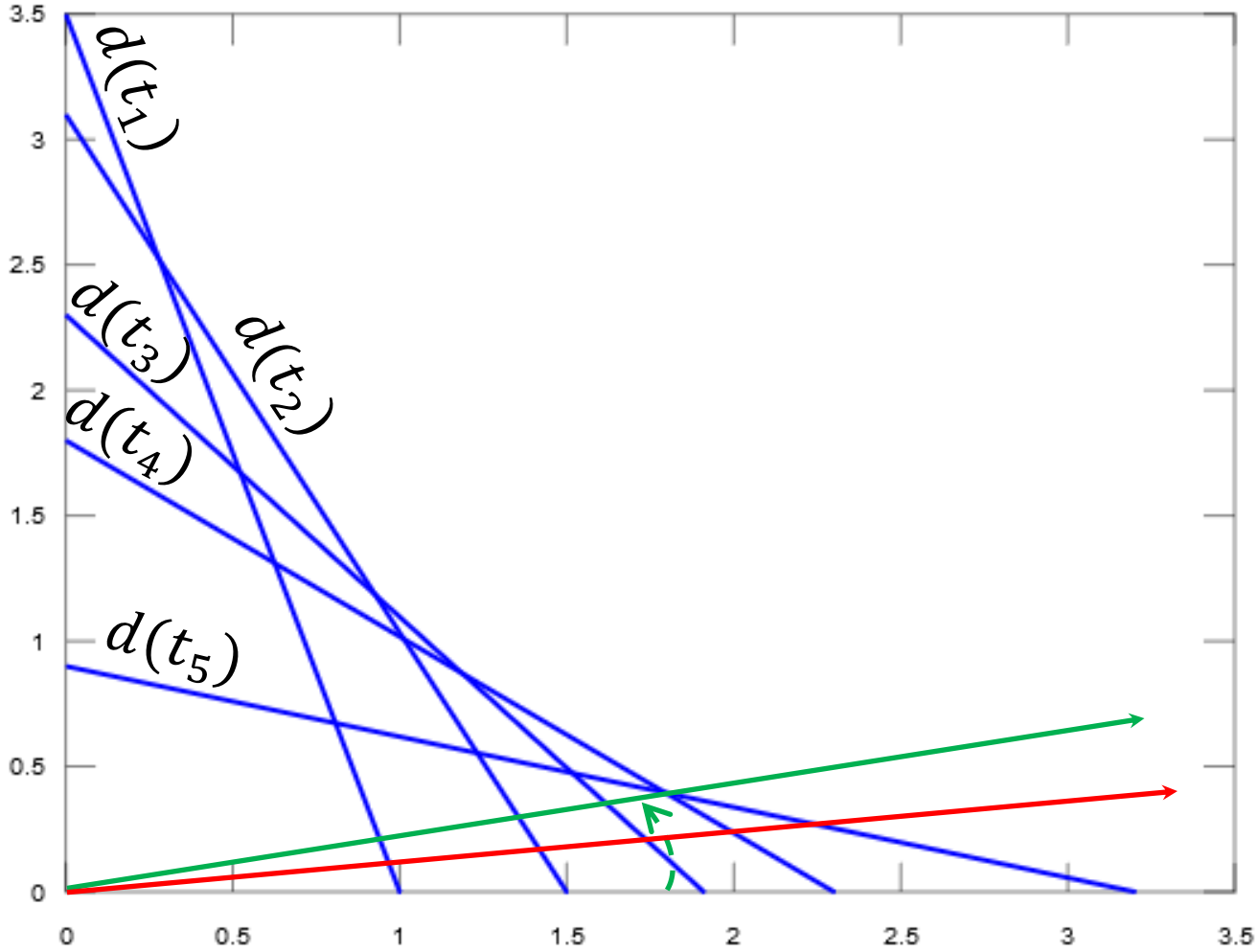
$d(t)$: $\sum t[i] \times x_i = 1$

2D:

$d(t)$: $t[1]x_1 + t[2]x_2 = 1$

Ranking Regions

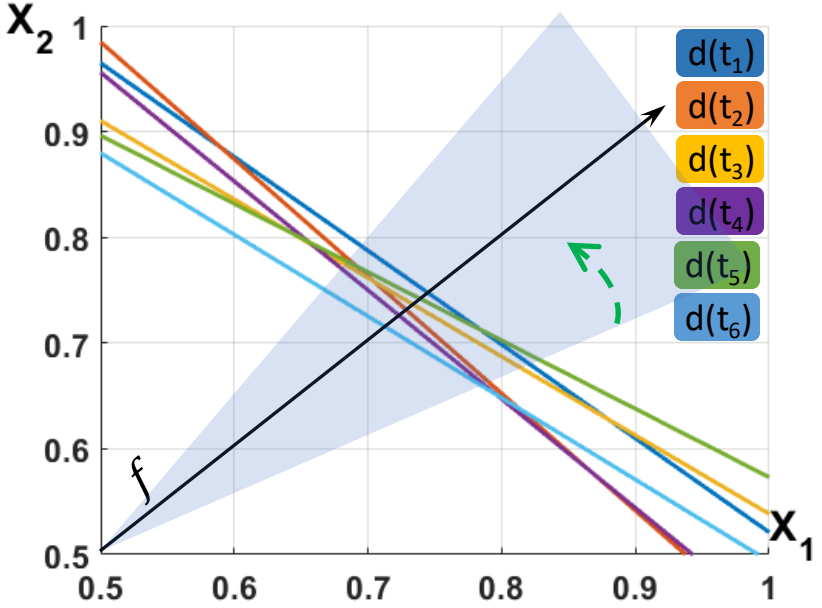
	x_1	x_2
t_1	3.5	1
t_2	3.1	1.5
t_3	2.3	1.91
t_4	1.8	2.3
t_5	0.9	3.2



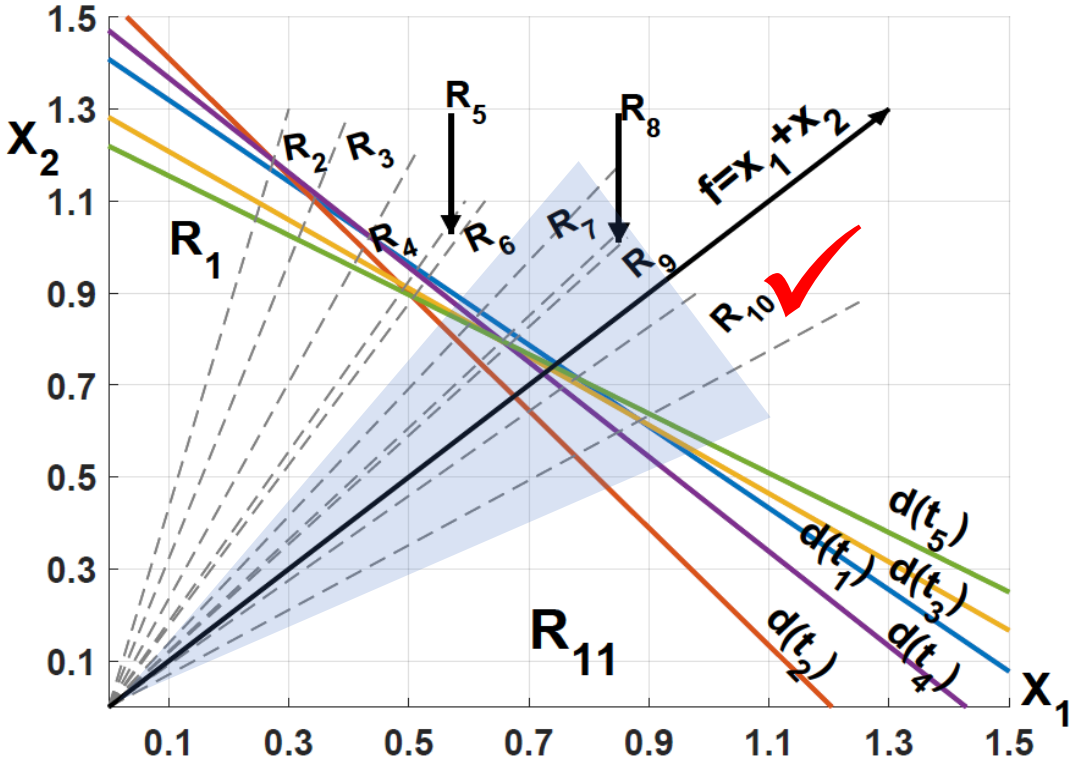
2D Algorithm

- Sweep a ray from along the region of interest
- find the regions (using the intersections)

- Construct a sorted list of regions by their width



2D Algorithm

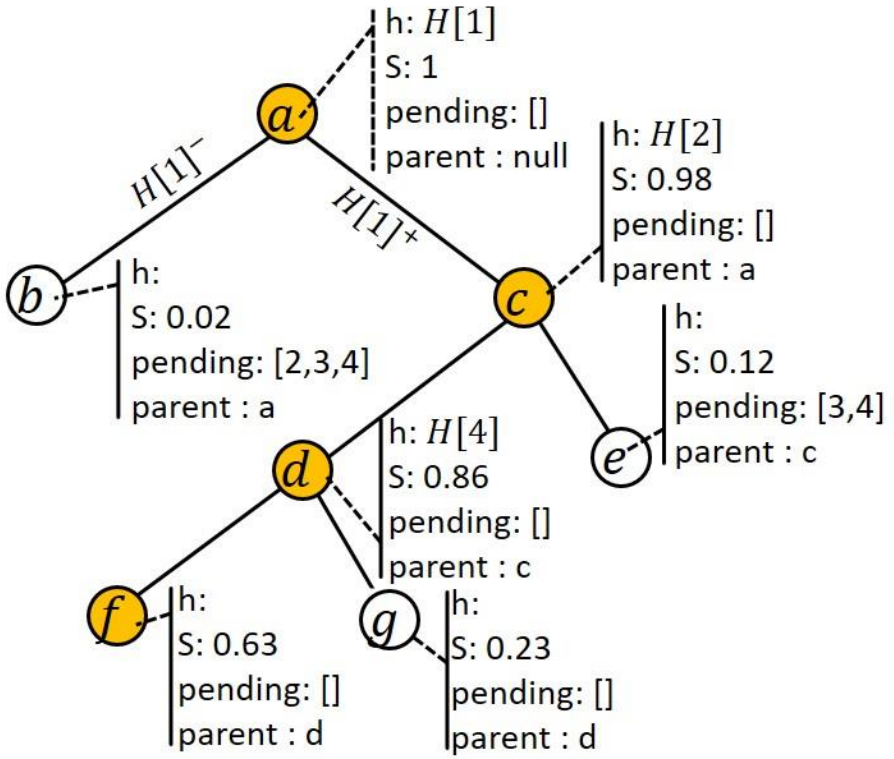


$\mathcal{D}$			f
id	x_1	x_2	$x_1 + x_2$
t_1	0.63	0.71	1.34
t_2	0.83	0.65	1.48
t_3	0.58	0.78	1.36
t_4	0.7	0.68	1.38
t_5	0.53	0.82	1.35

MD -- Threshold-based Algorithm

- The intersections in MD transform to hyperplanes
- The arrangement of hyperplanes partition the space into ranking regions

- In high-level:
  - Constructs the arrangement while only adds a hyperplane to the current largest region, postponing the process for the smaller regions



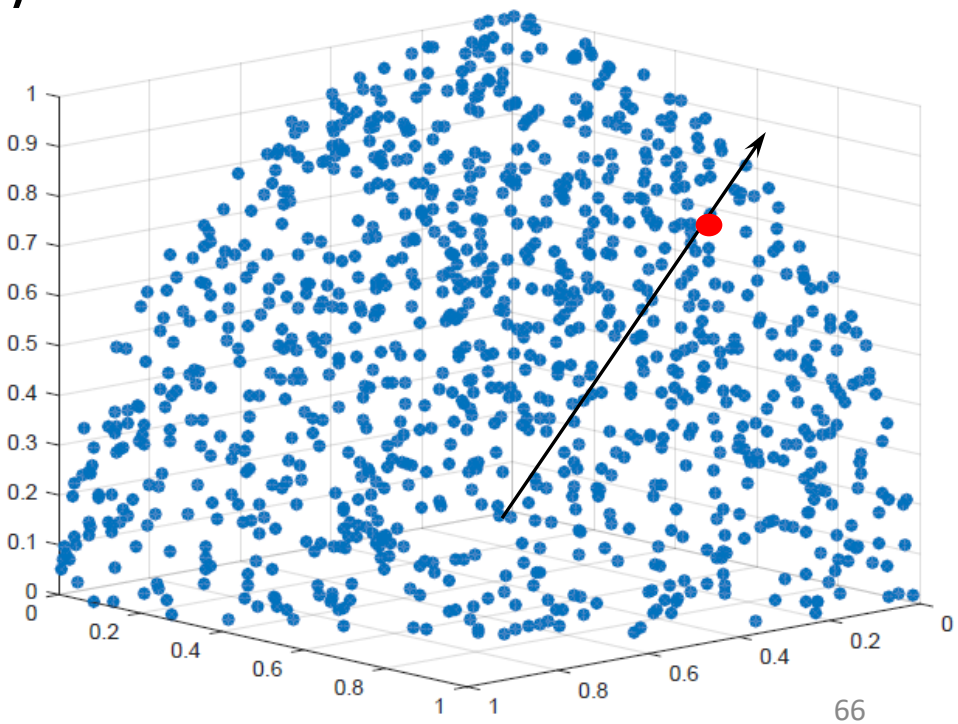
Randomized Get-Next

- A Monte-Carlo method that work based on repeated sampling and the central limit theorem

- Requires unbiased sampling from function space

Unbiased sampling from the function space

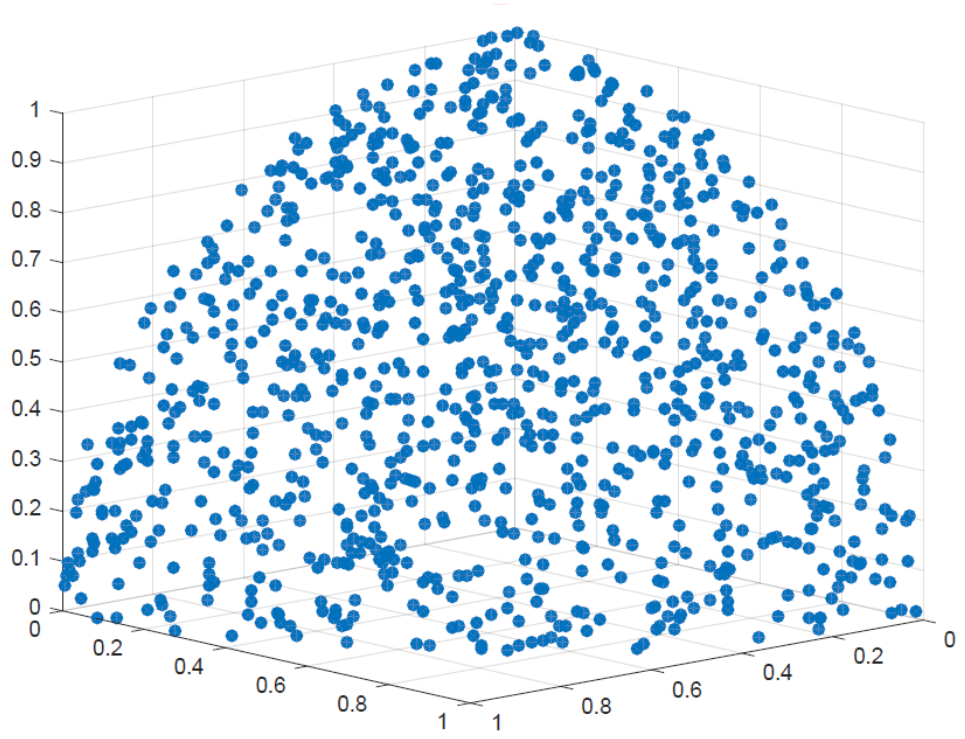
- 1-1 mapping b/w the functions (origin-starting rays) and the points on the *surface* of origin-centered unit *d-sphere* (hypersphere in $\mathbb{R}^d$)



* d: number of attributes

Unbiased sampling from the function space

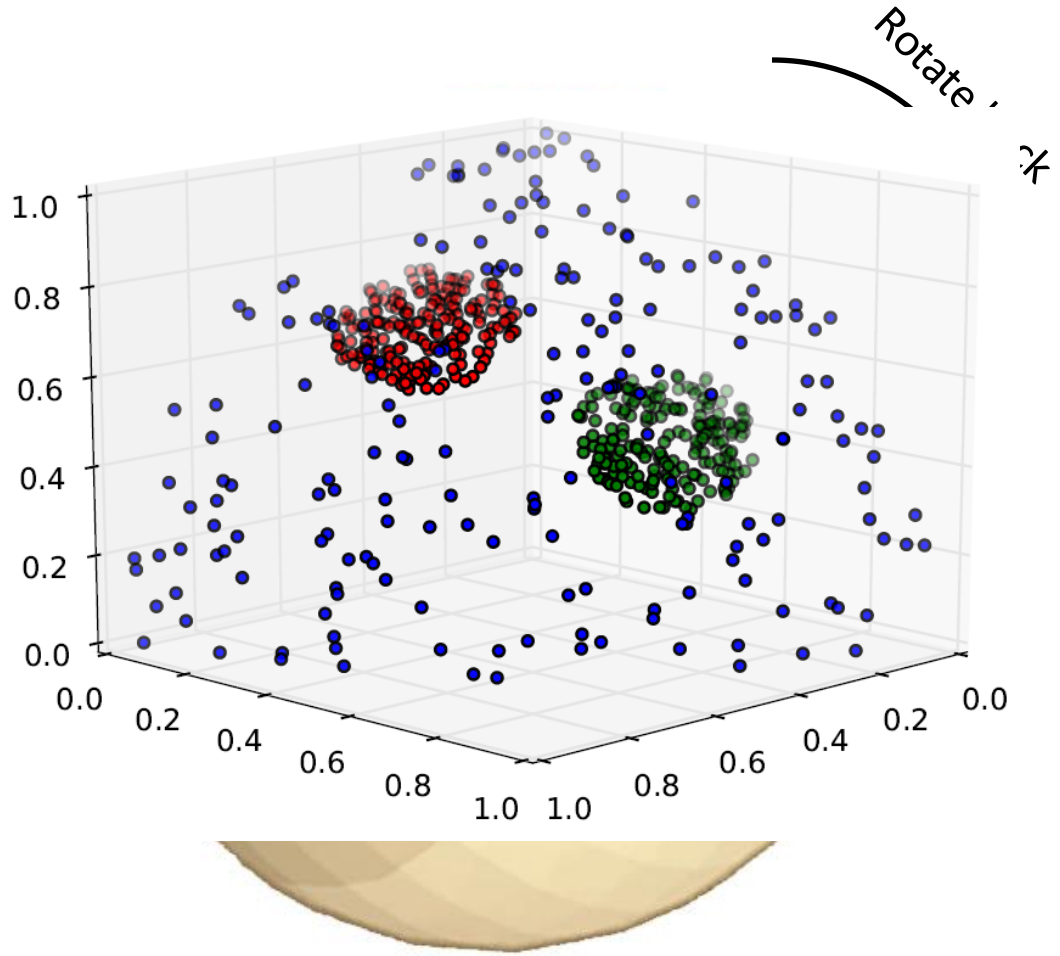
- Sampling the weights Uniformly? ✗
- Sampling the weights using the Normal distribution ✓



Sampling from a region of interest

- Each Riemann Piece is a (d-1)D Sphere (ring in 3D)
- We know how to sample from its surface!: *Normal distribution* ☺

- High-level:
  1. Select each “ring” randomly, proportional to its area
  2. Select a “point” from the surface of ring (using the Normal dist.)
  3. Rotate the space back



Randomized Get-Next

1. Take *N* unbiased sample functions from the region of interest
2. While keeping a hash of outputs, "count" the number of appearance for each output
3. return the output that appeared the most
    - estimate its stability & compute the confidence interval

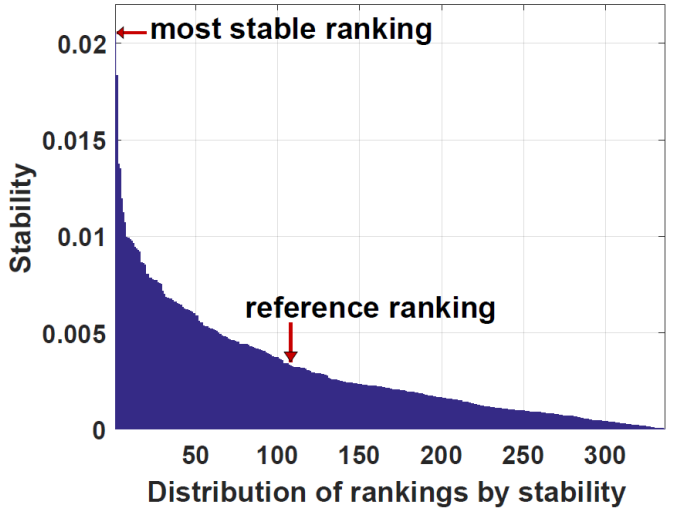
Function Sampling for **Arrangement Construction**:
Efficient! $O(h\,S)$ – independent of d
Interesting connection to the Partition algorithm!

* h: # hyperplanes; S: # Samples; d: # attributes

Some results: CSMetrics

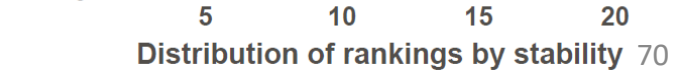
- A ranking of CS research institutions based on publication metrics.
- Ranking function: $M^{\alpha} P^{1-\alpha}$
  - *M*: measured citation
  - *P*: predicted citation
- $x_1 = \log M, x_2 = \log P$
  - $f = \alpha x_1 + (1-\alpha) x_2$
- We study the top-100 institutions

overall distribution of rankings by stability



stability around reference vector

- Cornell is not in the top-10 universities in the reference ranking.
- It replaces the University of Toronto in the top-10 in the most stable ranking

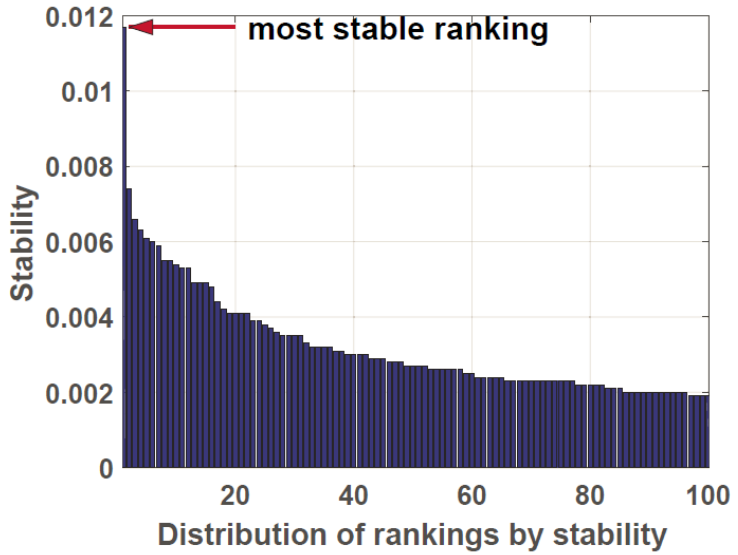


Some results: FIFA Rankings

- Ranking function:
  - $x_1 + .5x_2 + .3x_3 + .2x_4$
  - x_i: performance of a team in past i^{th} year

- Ref. ranking did not belong to the top-100 stable rankings!

stability around reference vector with 0.999 cosine similarity



Thank you!

https://www.cs.uic.edu/~asudeh/

https://www.cs.uic.edu/~asudeh/pub.htm

https://www.cs.uic.edu/~indexlab/

asudeh@uic.edu