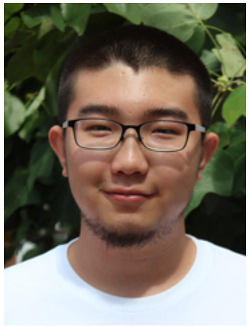
AURORA: Auditing PageRank on Large Graphs

Presented By Jian Kang


Jian Kang


Meijia Wang


Nan Cao

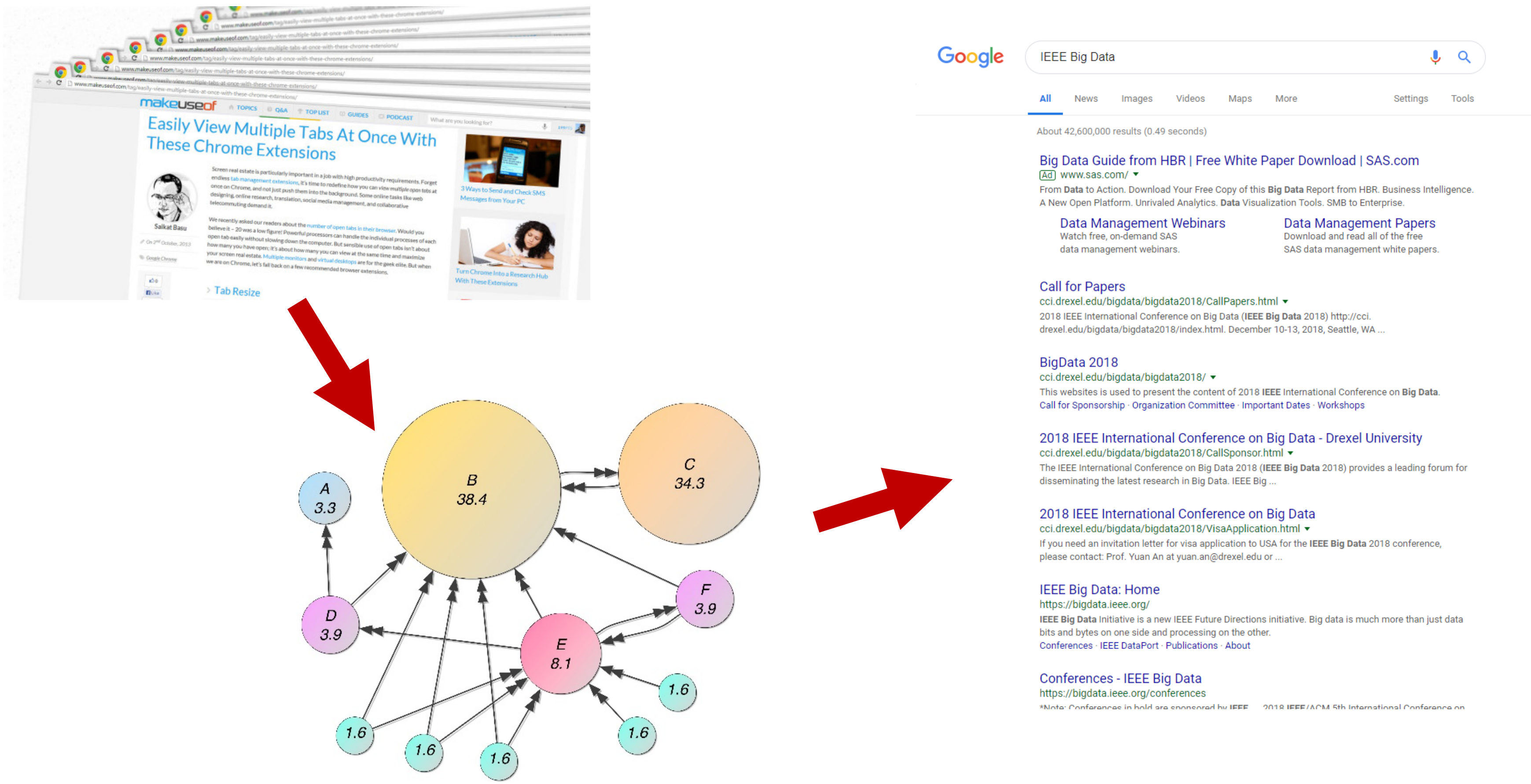

Yinglong Xia


Wei Fan


Hanghang Tong

Ranking on Graphs: PageRank

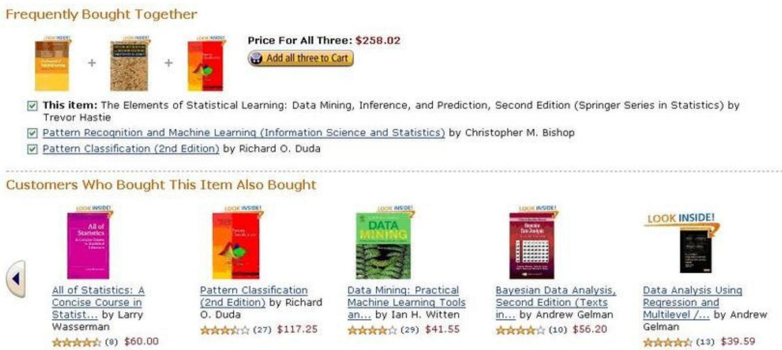
- Webpages are no longer independent
- Rank the webpages by their importance/relevance



DATA Lab

Arizona State University

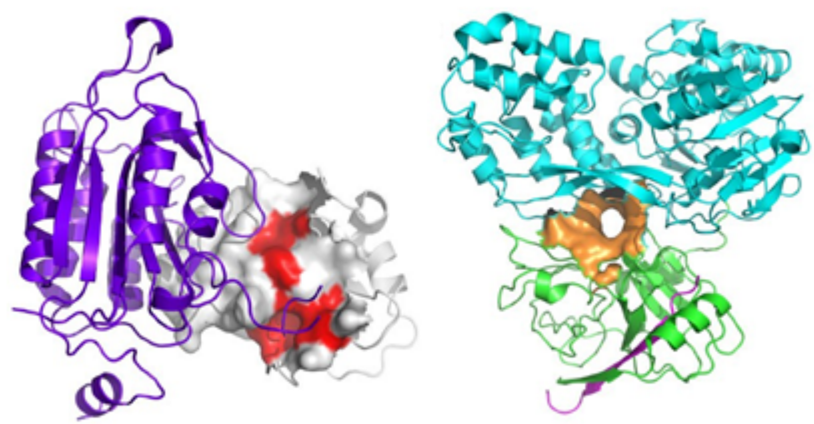
More Applications



Recommender System [Gori'07]

Social Network Analysis [Weng'10]

Sports Team Management [Radicchi'11]



Biology [Singh'07]

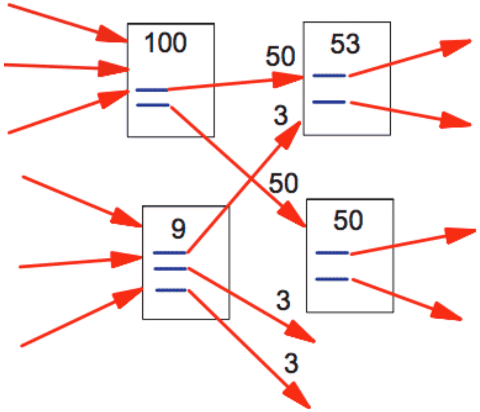
DATA Lab

PageRank: Formulation

- **Assumption:**
  - A webpage is important if it is linked by many other webpages
- **Formulation:**
  - Iteratively solve the following linear system

$$\mathbf{r} = c\mathbf{A}\mathbf{r} + (1 - c)\mathbf{e}$$

  - Mathematically elegant, only topological information is needed
- **Many Variants Exist:**
  - Personalized PageRank
  - Random Walk with Restart
  - And so on



Why Auditing PageRank?

- **Problem:** end-users do not understand how the results were derived

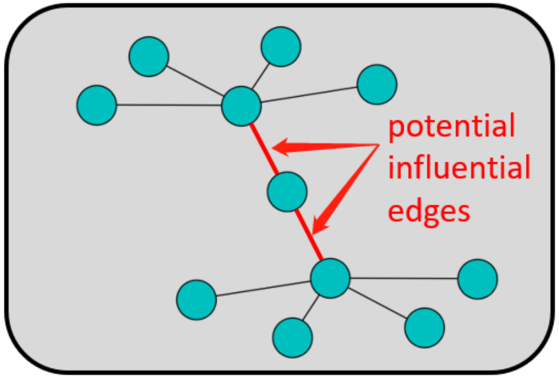
- **Potential Outcomes:**
  - Render crucial explainability of ranking algorithms
  - Optimize network topology
  - Identify vulnerabilities in the network (e.g. preventing adversarial attacks)

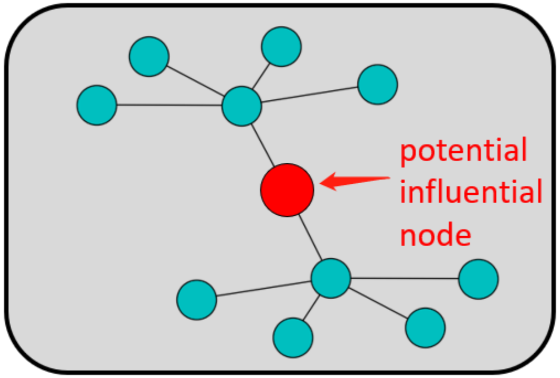
Roadmap

- Motivations ✓
- AURORA Formulation
- AURORA Algorithms
- AURORA Generalizations
- Experimental Results
- Conclusions

Prob. Def.: PageRank Auditing Problem

- **Given:**
  - (1) adjacency matrix $\mathbf{A}$;
  - (2) PageRank $\mathbf{r}$;
  - (3) loss function over PageRank vector $f(\mathbf{r})$;
  - (4) user-specific element type (edges vs. nodes vs. subgraph);
  - (5) integer budget k.
- **Find:** a set of k influential graph elements
- **Intuitive Example:**





AURORA Formulation

- **Intuition:** find a set of influential elements that have largest impact on the loss function over PageRank vector.
- **Optimization Problem:**

$$\max_{S} \quad \Delta f = \left(f(\mathbf{r}) - f(\mathbf{r}_S)\right)^2$$

impact of set S on the loss function

$$s.t. \quad |S| = k$$

- **Choices of Loss Function:**
  - Squar

TABLE II: Choices of $f(\cdot)$ functions and their derivatives

Descriptions	Functions	Derivatives
L_p norm	$f(\mathbf{r}) = \|\|\mathbf{r}\|\|_p$	$\frac{\partial f}{\partial \mathbf{r}} = \frac{\mathbf{r} \circ \|\mathbf{r}\|^{p-2}}{\|\|\mathbf{r}\|\|_p^{p-1}}$
Soft maximum	$f(\mathbf{r}) = log(\sum_{i=1}^{n} exp(\mathbf{r}(i)))$	$\frac{\partial f}{\partial \mathbf{r}} = [\frac{exp(\mathbf{r}(i))}{\sum_{i=1}^{n} exp(\mathbf{r}(i)}]$
Energy norm	$f(\mathbf{r}) = \mathbf{r}'\mathbf{M}\mathbf{r}$	$\frac{\partial f}{\partial \mathbf{r}} = (\mathbf{M} + \mathbf{M}')\mathbf{r}$

($\mathbf{M}$ in Energy Norm is a Hermitian positive definite matrix.)

Challenges

- C1: Measure of Influence
- C2: Optimality
- C3: Scalability

Challenges

- **C1: Measure of Influence**
  - Understanding Black-box Machine Learning Models
    - Quantify influence by perturbing features or training data.
    - **Obs:** Inconsistent with unsupervised graph ranking settings.
  - Influence Maximization
    - Measure the size of ‘infected’ nodes in information propagation process.
    - **Obs:** fundamentally different from finding influential elements in graph ranking settings.
  - **Question:** how to define the influence in the context of graph ranking?

[1] Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., & Venkatasubramanian, S. (2018). Auditing black-box models for indirect influence. *Knowledge and Information Systems*, *54*(1), 95-122.
[2] Koh, P. W., & Liang, P. (2017, July). Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning* (pp. 1885-1894).
[3] Kempe, D., Kleinberg, J., & Tardos, É. (2003, August). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137-146). ACM.

Challenges

- **C2: Optimality**
  - Finding a set of influential graph elements is NP due to its combinatorial nature.
  - **Question:** how to find a set of influential graph elements accurately?

- **C3: Scalability**
  - **Question:** how to scale up the influential elements finding process?

Definition: Graph Element Influence

- **Graph Element Influence**
  - The influence of an edge (i, j) is defined as the derivative of $f(\mathbf{r})$ w.r.t. the edge.

$$\mathbb{I}(i,j) = \frac{\mathrm{d}f(\mathbf{r})}{\mathrm{d}\mathbf{A}(i,j)}$$

  - The influence of a node i is defined as the aggregation of all in and out edges.

$$\mathbb{I}(i) = \sum_{j=1, j\neq i}^{n} \mathbb{I}(i,j) + \mathbb{I}(j,i)$$

  - The influence of a subgraph S is defined as the aggregation of all edges in the subgraph.

$$\mathbb{I}(i) = \sum_{i,j\in S}^{n} \mathbb{I}(i,j)$$

DATA Lab

Calculating Influence

- **Method:**
  - Define $\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$, PageRank: $\mathbf{r} = (1-c)\mathbf{Qe}$
  - Apply chain rule

$$\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}(i,j)} = \mathrm{Tr}[(\frac{\partial f(\mathbf{r})}{\partial \mathbf{r}})' \frac{\partial \mathbf{r}}{\partial \mathbf{A}(i,j)}] = 2c\mathbf{r}(j)\mathrm{Tr}[\mathbf{r}'\mathbf{Q}(:,i)]$$

- **Matrix Form Solution:**

$$\frac{\mathrm{d}f(\mathbf{r})}{\mathrm{d}\mathbf{A}} = \begin{cases} \frac{\partial f(\mathbf{r})}{\partial \mathbf{A}} + \left(\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}}\right)' - \mathrm{diag}\left(\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}}\right) & \text{, if } \mathbf{A} \text{ is undirected graph} \\ \frac{\partial f(\mathbf{r})}{\partial \mathbf{A}} & \text{, if } \mathbf{A} \text{ is directed graph} \end{cases}$$

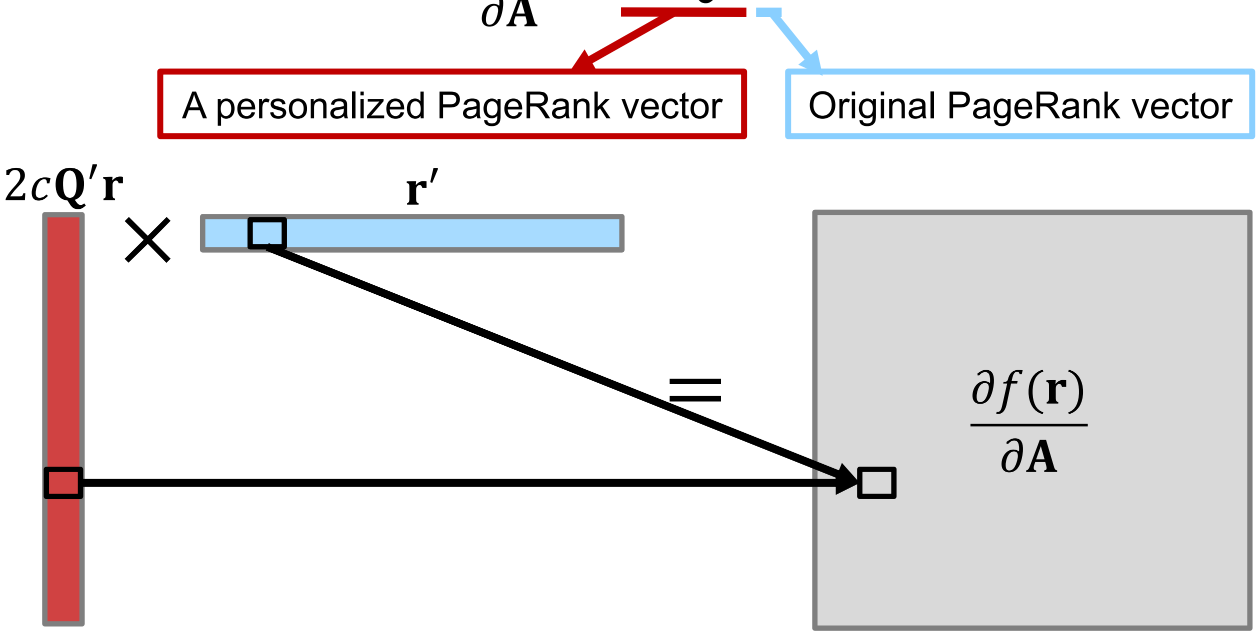
where $\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}} = 2c\mathbf{Q}'\mathbf{r}\mathbf{r}'$, each element in $\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}}$ is $\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}(i,j)}$

  - **Limitation:** $\mathbf{Q}'\mathbf{r}\mathbf{r}'$ is an $n \times n$ full matrix, need $O(n^2)$ space
  - **Question:** how to scale up to large graphs?

Scale Up

- **Solution:** exploring low-rank structure
  - Note that PageRank $\mathbf{r} = (1-c)\mathbf{Qe}$

$$\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}} = 2c\mathbf{Q}'\mathbf{r}\mathbf{r}'$$



  - Reduce $O(n^2)$ space to $O(n)$ space

Roadmap

- Motivations ☑
- AURORA Formulation ☑
- AURORA Algorithms
- AURORA Generalizations
- Experimental Results
- Conclusions

AURORA Algorithms

- **Goal:** select a set of k influential graph elements
- **Observation:**
  - $\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}}$ is a non-negative matrix, so does $\frac{\mathrm{d}f(\mathbf{r})}{\mathrm{d}\mathbf{A}}$.
  - Enjoys diminishing returns property → submodular function
- **Greedy Strategy:**
  - iteratively select the most influential element in each round;
  - remove the selected element and re-rank;
  - repeat above procedure k rounds.
- **Challenges:** computationally expensive to calculate $\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}}$
- How to speed up? → power iterations

Roadmap

- Motivations ☑
- AURORA Formulation ☑
- AURORA Algorithms ☑
- AURORA Generalizations
- Experimental Results
- Conclusions

AURORA Generalizations: Normalized PageRank

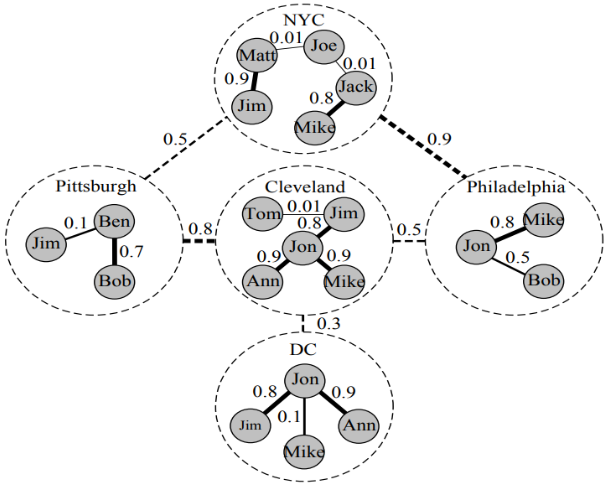
- **Intuition:** normalize PageRank vector to magnitude of 1
- **Key Idea:** divide each PageRank score with the sum of all PageRank scores
- **Formulation**:
  - Let $S(\mathbf{r}) = \sum_{i=1}^{n} \mathbf{r}(i)$, then

$$\frac{\partial f(\mathbf{r})}{\partial \mathbf{A}} = c\mathbf{Q}'(-\frac{2f(\mathbf{r})}{S(\mathbf{r})}\mathbf{1} + \frac{2}{S(\mathbf{r})}\mathbf{r})\mathbf{r}'$$

- **Solution:** apply similar strategy as AURORA
- More details in the paper

AURORA Generalizations: NoN

- **NoN** (Network of Networks) is defined as a triplet $< \mathbf{G}, \boldsymbol{A}, \theta >$.
  - $\mathbf{G}$: main network
  - $\boldsymbol{A}$: domain-specific networks
  - θ: mapping function



- **Ranking on NoN:**

$$\min \ J(\mathbf{r}) = c\mathbf{r}'(\mathbf{I}_n - \mathbf{A})\mathbf{r} + (1-c)\|\mathbf{r}-\mathbf{e}\|_F^2 + 2a\mathbf{r}'\mathbf{Y}\mathbf{r}$$

within-network smoothness query preference cross-network consistency

  - equivalent to PageRank with transition matrix $\mathbf{W} = \frac{c}{c+2a}\mathbf{A} + \frac{2a}{c+2a}\mathbf{Y}$

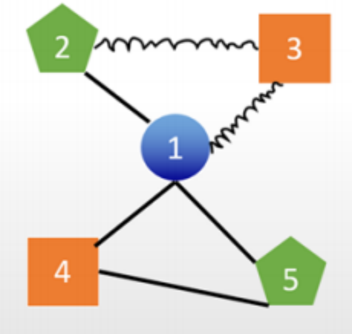
- **Solution:** Apply similar strategy as AURORA

[1] Ni, J., Tong, H., Fan, W., & Zhang, X. (2014, August). Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1356-1365). ACM.

AURORA Generalizations: Attributed Networks

- **Intuition:** find influential attributes in attributed networks.
- **Key Idea:** treat attributes as *attribute nodes* and form an *augmented graph.*
- **Supporting Node Attributes:**
  - (1) $\mathbf{A}$: node-to-node adjacency matrix;

    (2) $\mathbf{W}$: attribute-to-node adjacency matrix.
  - Form an augmented graph $\mathbf{G} = \begin{pmatrix} \mathbf{A} & \mathbf{W}' \\ \mathbf{W} & \mathbf{A}' \end{pmatrix}$



Node attributes: different shapes
Edge attributes: straight vs. curved lines

- **Supporting Edge Attributes:**
  - Let $\mathbf{A}$ be an $n \times n$ adjacency matrix and x be the number of different edge attributes.
  - Embed edge attributes into edge-nodes.
  - Form an $(n+x) \times (n+x)$ augmented graph.
- **Solution:** Apply similar strategy as AURORA

[1] Tong, H., Faloutsos, C., Gallagher, B., & Eliassi-Rad, T. (2007, August). Fast best-effort pattern matching in large attributed graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 737-746). ACM.
[2] Pienta, R., Tamersoy, A., Tong, H., & Chau, D. H. (2014, October). Mage: Matching approximate patterns in richly-attributed graphs. In *Big Data (Big Data), 2014 IEEE International Conference on* (pp. 585-590). IEEE.

Roadmap

- Motivations ☑
- AURORA Formulation ☑
- AURORA Algorithms ☑
- AURORA Generalizations ☑
- Experimental Results
- Conclusions

Datasets

- Over 10+ real-world datasets

Category	Network	Type	Nodes	Edges
SOCIAL	Karate	U	34	78
	Dolphins	U	62	159
	WikiVote	D	7,115	103,689
	Pokec	D	1,632,803	30,622,564
COLLABORATION	GrQc	U	5,242	14,496
	DBLP	U	42,252	420,640
	NBA	U	3,923	127,034
	cit-DBLP	D	12,591	49,743
	cit-HepTh	D	27,770	352,807
	cit-HepPh	D	34,546	421,578
PHYSICAL	Airport	D	1,128	18,736
OTHERS	Lesmis	U	77	254
	Amazon	D	262,111	1,234,877

(In Type, U means undirected graph; D means directed graph.)

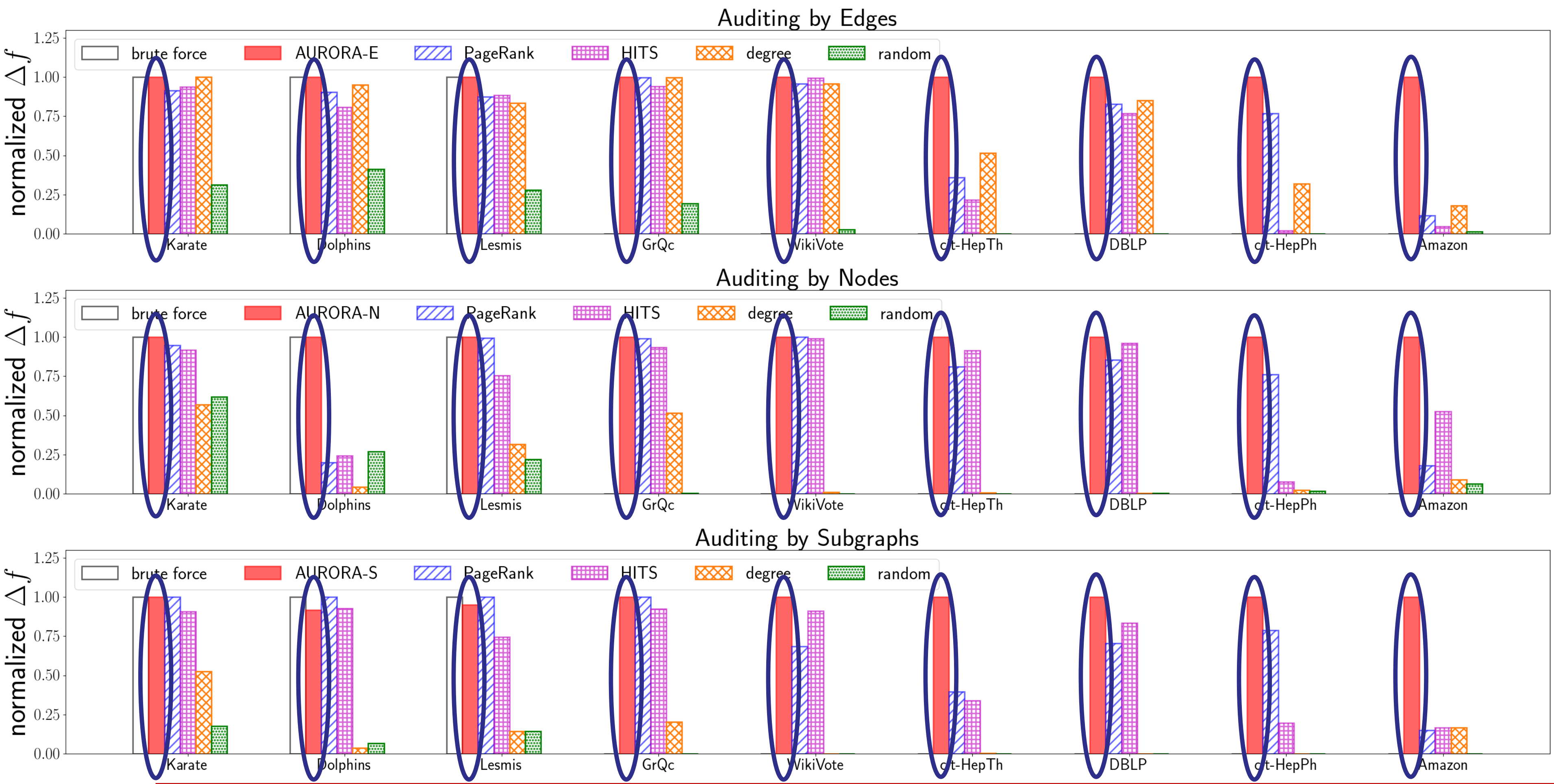
Experimental Settings

- **Evaluation Metric**
  - Effectiveness: difference in $f(r)$
  - Efficiency: running time
- **Baseline Methods**

AURORA (Our Methods)	Baseline Methods
❑ AURORA-E	❑ Brute force
❑ AURORA-N	❑ Random selection
❑ AURORA-S	❑ Top-k degree
	❑ PageRank
	❑ HITS

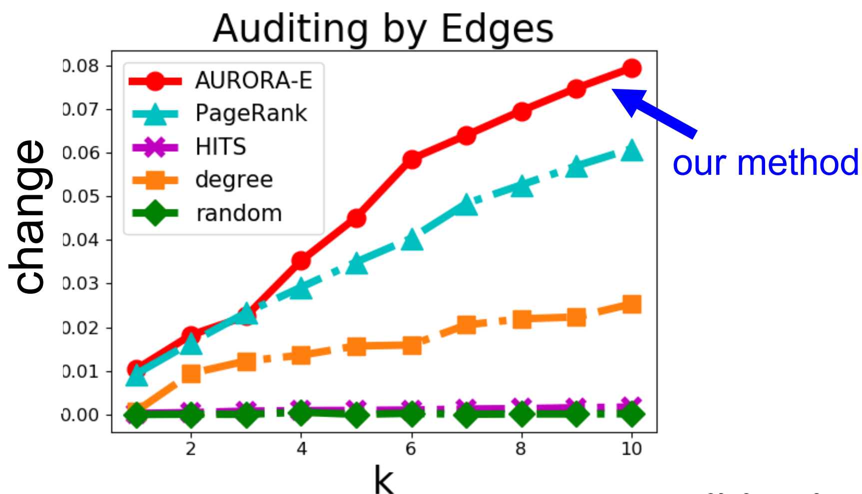
Effectiveness: Fixed Budget (Higher is Better)

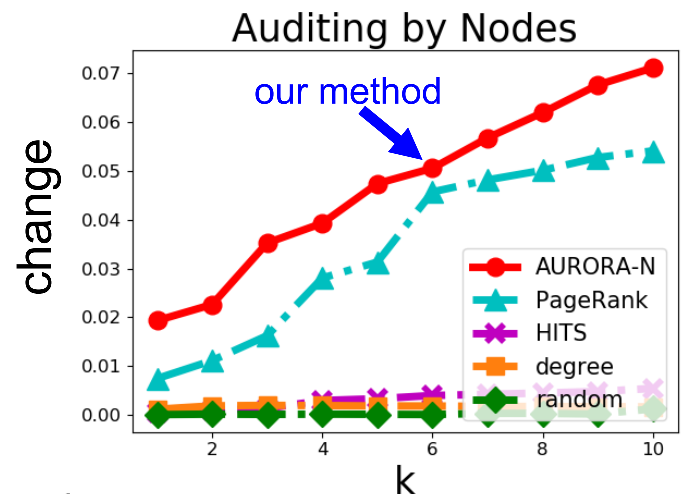
- **Observation:** AURORA outperforms baseline methods

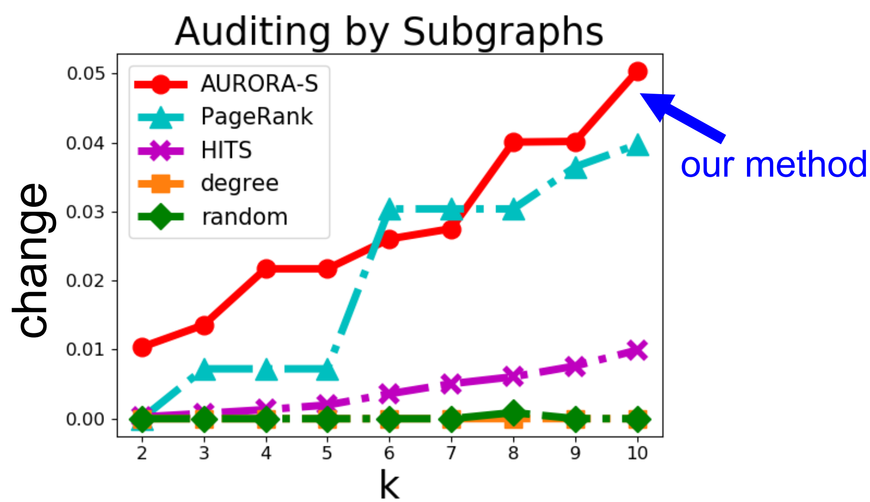


Effectiveness (Higher is better)

- **Observation:** AURORA outperforms baseline methods

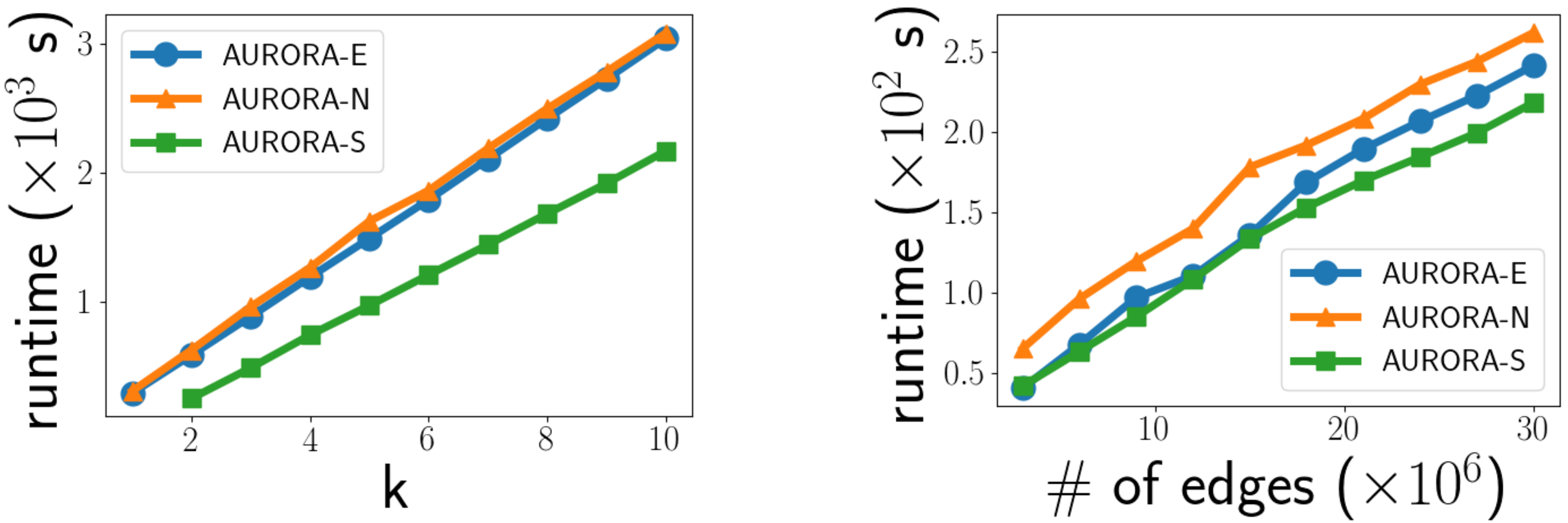






Efficiency

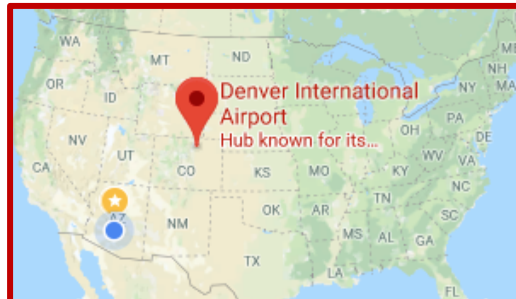
- **Observation:** linear complexity w.r.t. k and m



Case Study on Airport Dataset

- **Goal:** find important airline routes and airports
- **Results:**

Task	PageRank	AURORA
Edge Auditing	ATL-LAS	DEN-ATL
	ATL-DFW	LAX-ORD
Node Auditing	SFO	CLT



DEN serves as a major hub airport to connect west and east coasts

It directly connects Los Angeles (LAX) and Chicago (ORD), two largest cities in United States.

Busiest Airports: CLT(6th) > SFO (7th)
Proximity: existence of LAX and SJC

Case Study on NBA Dataset

- **Goal:** find a team in collaboration network
- **Query:** Allen Iverson
- **Results:**

Task	PageRank	AURORA
Subgraph Auditing (Graph size: 5)	Allen Iverson Larry Hughes Theo Ratliff Joe Smith *Drew Gooden*	Allen Iverson Larry Hughes Theo Ratliff Joe Smith *Tim Thomas*

NEVER played with Allen Iverson.

Roadmap

- Motivations ☑
- AURORA Formulation ☑
- AURORA Algorithms ☑
- AURORA Generalizations ☑
- Experimental Results ☑
- Conclusions

Conclusions

- **Problem:**
  - PageRank Auditing Problem
- **Solution:**
  - Family of AURORA algorithms
  - Near-optimal results
  - Scalability
- **Results:**
  - Outperform other baseline methods
  - Achieves linear time complexity
  - Finds intuitive and meaningful explanations
- More details in the paper