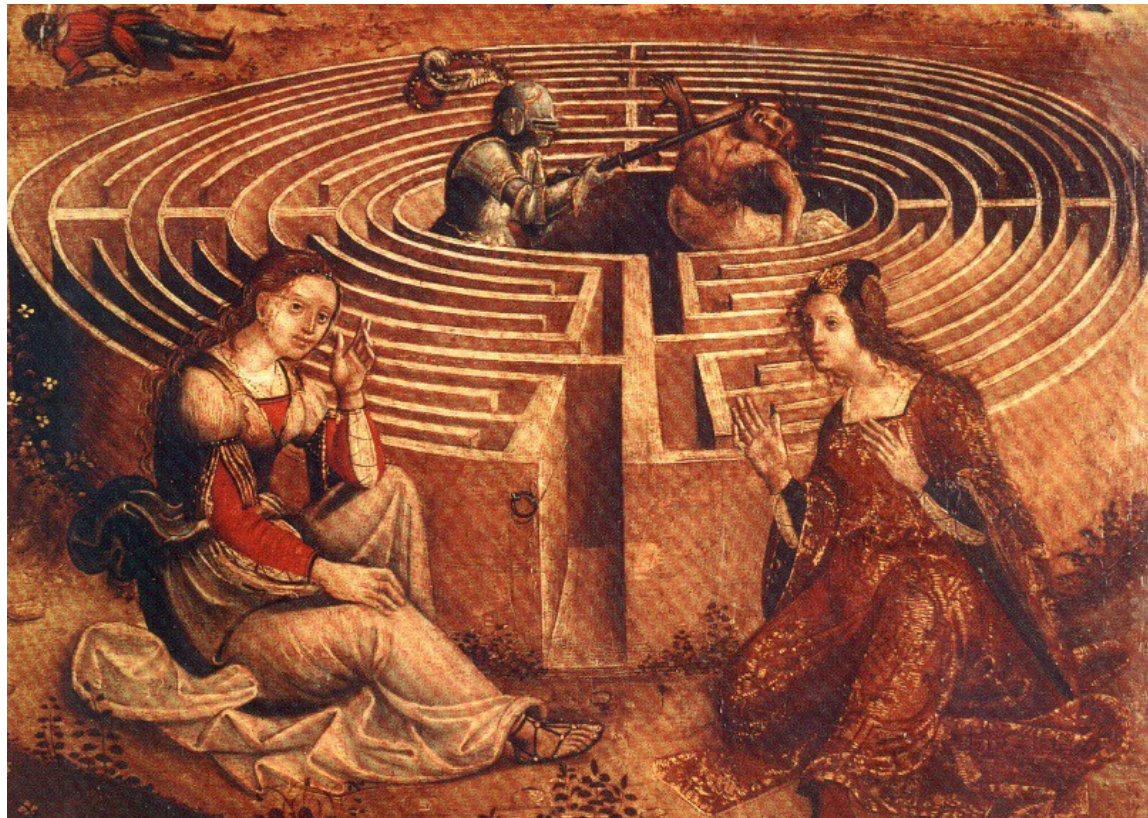


# VC-dimension: Ariadne's Thread in the Big Data Labyrinth

(was: Using VC-dimension for faster computation and tighter analysis)



Matteo Riondato

Thesis Defense

April 18, 2014

# Outline

## Introduction

- Problem
- Thesis statement
- Contributions
- VC-dimension

## Estimating betweenness centrality

## Conclusions

# What am I talking about?

## Sampling-based Randomized Algorithms for Big Data Analytics

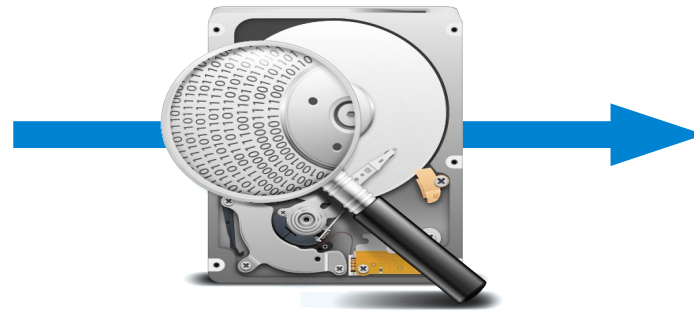
```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

from xkcd.com

# What is data analytics?



Data



Data analytics

=

cleaning, inspecting, transforming, modeling, ...



Information

Needs fast algorithms —————> challenging due to Big Data

# Why is Big Data a challenge?

**Volume:** data size is large and grows

**Variety:** no. of “questions” is large



$$\text{cost}(\text{analytics algorithm}) = \text{cost}(\text{Volume}) + \text{cost}(\text{Variety})$$

E.g.,  $\text{cost}(\text{APriori}) = \text{cost}(\text{size dataset}) + \text{cost}(\text{no. of patterns})$

Smart algorithms may cut  $\text{cost}(\text{Variety})$

- $\text{cost}(\text{Volume})$  always takes over

# Thesis statement

We use VC-dimension to obtain high-quality approximations for many data analytics tasks by processing a small random sample of the data

- Probabilistic guarantees on quality of approximations
- Tasks from data mining, graph analysis, database management
- In 1 line:  
“Hey guys, you forgot about this theorem. Here's how to use it.”



# What are our contributions?

## Database query selectivity

- characterization of VC-dimension of SQL queries
- sampling-based algorithm – **smallest** sample size

## Frequent Itemsets / Association Rules

- sampling-based algorithm – **smallest** sample size
- MapReduce algorithm – **fastest** and **most scalable**
- statistical test – **more statistical power** than available solutions

## Betweenness centrality

- sampling-based algorithm – **fastest** available
- tighter analysis of existing sampling-based algorithm





# Why sampling?

Natural solution to cut cost(Volume)

Implies approximations

- OK: data analytics is exploratory



Trade-off:

- larger sample = better approximation but slower algorithm
  - quantified by deviation bounds (Chernoff, Azuma, VC-dimension, ...)



# Why VC-dimension?

Chernoff+Union **too weak for Big Data analytics**

- Chernoff: guarantee on answer to single question
- Union: guarantee extended to all questions
- Sample size depends on **no. of questions (Variety)**:

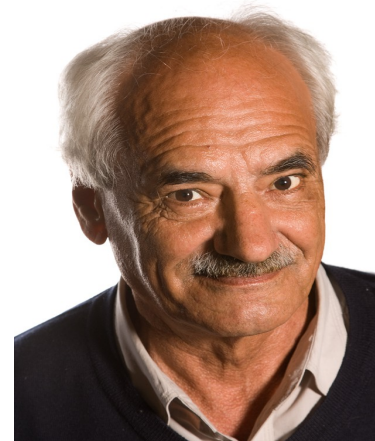
$$|\mathcal{S}| = \frac{1}{\varepsilon^2} \left( \log_2 |Q| + \ln \frac{1}{\delta} \right)$$

**VC-dimension** overcomes this issue:

$$|\mathcal{S}| = \frac{1}{\varepsilon^2} \left( \text{VC}(Q) + \ln \frac{1}{\delta} \right)$$



Vladimir N. Vapnik



Aleksey J. Chervonenkis



# What is VC-dimension?

$D$  : set of points  
 $F$  : collection of subsets of  $D$  (ranges) }  $(D, F)$   
rangeset

VC-dimension of  $(D, F)$  measures “richness” of  $F$

For any  $C \subseteq D$ , let  $P_C = \{C \cap r : r \in F\} (\subseteq 2^C)$

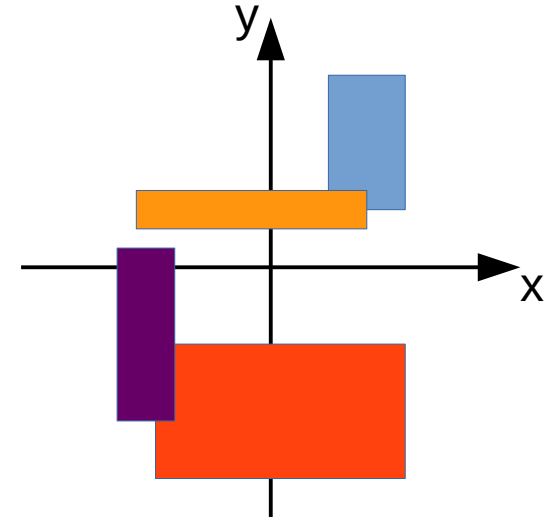
If  $P_C = 2^C$ , then  $C$  is shattered by  $F$

$$\text{VC}(D, F) = \sup \{ |C| : C \subseteq D \wedge P_C = 2^C \}$$

# VC-dimension – Example

$$D = \mathbb{R}^2$$

$F$  = all axis-aligned rectangles

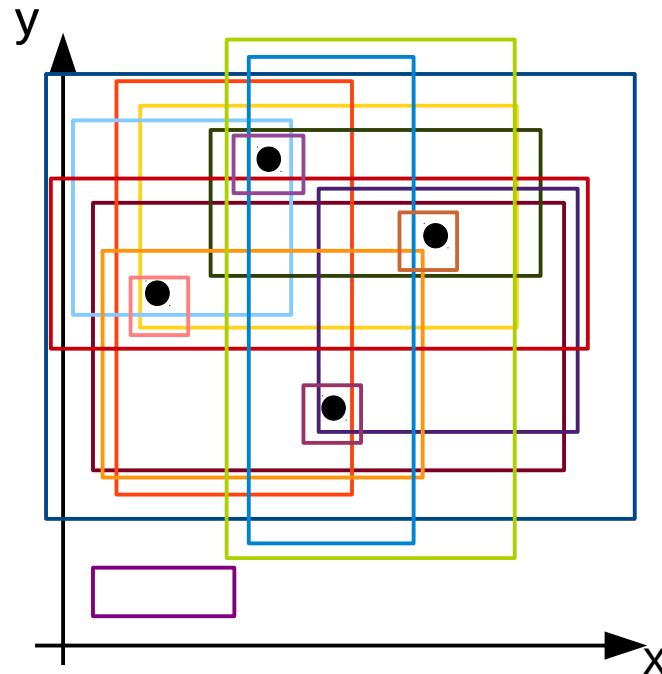
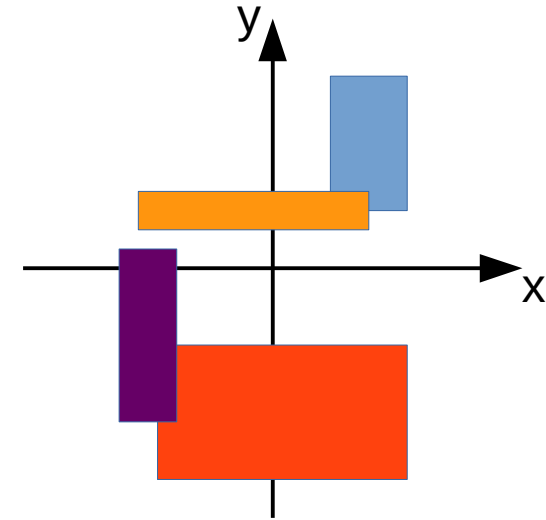


# VC-dimension – Example

$$D = \mathbb{R}^2$$

$F$  = all axis-aligned rectangles

Shattering 4 points? **easy!** Need 16 rectangles

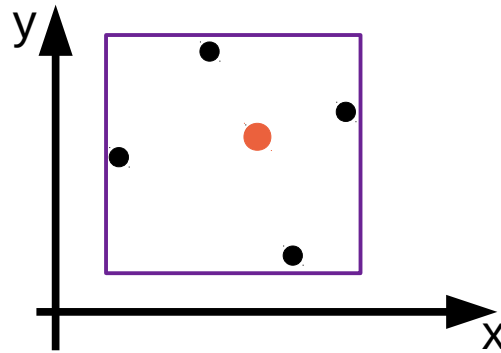


# VC-dimension – Example

$$D = \mathbb{R}^2$$

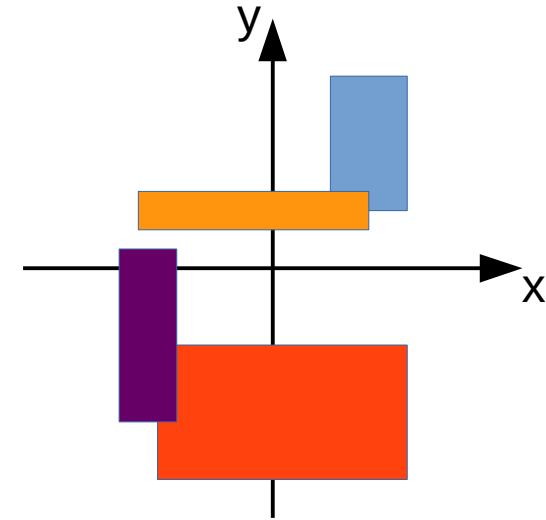
$F$  = all axis-aligned rectangles

Shattering 5 points? impossible!



One point is contained in all rectangles containing the other four

$$VC(D, F) = 4$$



# How does it relate to sampling?

**Theorem** ([Vapnik and Chervonenkis '71] [Li et al. '08])

- Fix  $0 < \varepsilon, \delta < 1$ , and assume  $\text{VC}(D, F) \leq d$
- $\pi$  : probability distribution on  $D$
- $\mathcal{S}$  : collection of samples from  $D$ , according to  $\pi$ , with

$$|\mathcal{S}| \geq \frac{1}{\varepsilon^2} \left( d + \log \frac{1}{\delta} \right)$$

- Then, with probability  $\geq 1 - \delta$

$$\left| \pi(R) - \frac{1}{|\mathcal{S}|} \sum_{a \in \mathcal{S}} \mathbb{1}_R(a) \right| \leq \varepsilon, \text{ for any } R \in F$$



# What do we need to use it?

- analytics task as probability estimation problem
- definition of  $D$  and  $F$
- probability distribution  $\pi$  on  $D$
- efficient procedure to sample from  $\pi$
- upper bound to  $VC(D, F)$ 
  - must be efficient to compute

# Outline

## ✓ Introduction

- ✓ Problem
- ✓ Thesis statement
- ✓ Contributions
- ✓ VC-dimension



## Estimating betweenness centrality

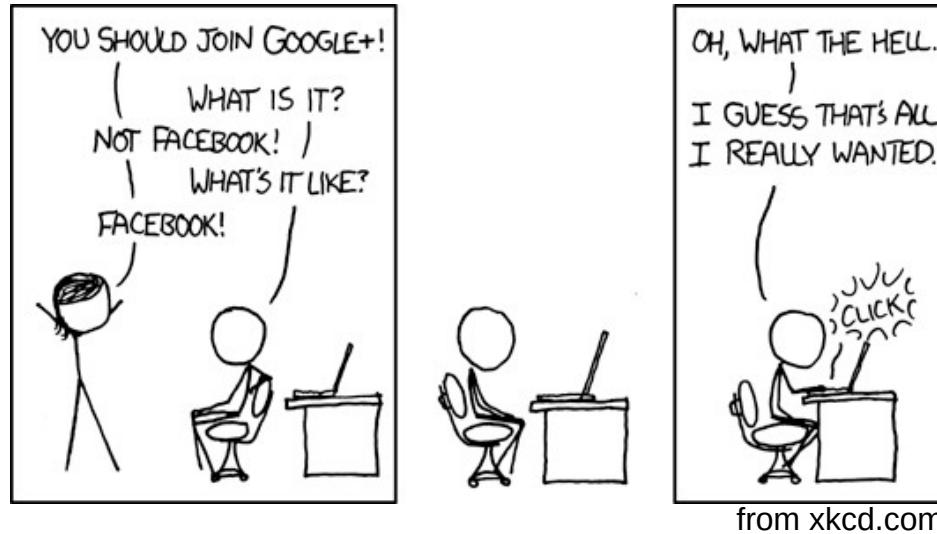
- Rangeset and bounds
- Algorithms

## Conclusions



# What's the setting?

Take a **social network**. Even Google+



What do you do with it? You **analyze** it

- If the NSA does it, it must be useful, right?

What are you analyzing about?

# What vertices in a graph are important?

**Betweenness centrality:** measure of vertex importance

- fraction of **shortest paths** that go through vertex

Graph  $G = (V, E)$   $|V| = n$   $|E| = m$

$$b(v) = \frac{1}{n(n-1)} \sum_{p_{uw} \in \mathbb{S}_G} \frac{\mathbb{1}_{\mathcal{T}_v}(p_{uw})}{\sigma_{uw}}$$

$\mathbb{S}_G$  = all shortest paths in  $G$

$\mathcal{S}_{uv}$  = set of shortest paths from  $u$  to  $v$  ( $\sigma_{uv} = |\mathcal{S}_{uv}|$ )

$\mathcal{T}_v = \{p \in \mathbb{S}_G : v \in \text{Int}(p)\}$

# How can we compute it?

**Naïve algorithm:** all pairs shortest paths + aggregation

- Aggregation part dominates. Complexity:  $\Theta(n^3)$

**[Brandes '01]:**

- aggregation after each Single Source Shortest Path computation
- Complexity:  $O(nm)$  or  $O(nm + n^2 \log n)$

**Too much** for networks with  $10^9$  vertices,  $10^{10}$  edges



what to sample?  
how much?

**Solution:** fewer SP computations using sampling!

# What do we want to get?

Probabilistic guarantees on approximation

$(\varepsilon, \delta)$ -approximation: values  $(\tilde{b}(v))_{v \in V}$  such that

$$\Pr \left( \exists v \in V : |\tilde{b}(v) - b(v)| > \varepsilon \right) < \delta$$

accuracy

confidence

Trade-off: smaller  $\varepsilon$  or  $\delta$ , higher number of samples



# A first sampling algorithm

[BrandesPich '08]:

- $r \leftarrow \frac{1}{2\varepsilon^2} \left( \ln n + \ln 2 + \ln \frac{1}{\delta} \right)$
- for  $i \leftarrow 1, \dots, r$ 
  - $v_i \leftarrow$  **random** vertex
  - Perform SSSP from  $v_i$
  - Perform partial aggregation for  $\tilde{b}(u), u \in V$  (like in exact algorithm)
- output  $\tilde{b}(v), \forall v \in V$

# How do they compute the sample size?

- Hoeffding bound for **single** vertex

$$\Pr(|\tilde{\mathbf{b}}(v) - \mathbf{b}(v)| > \varepsilon) < 2e^{-2r\varepsilon^2}$$

Wassily Hoeffding

- **union bound** over  $n$  vertices: we want

$$2e^{-2r\varepsilon^2} \leq \frac{\delta}{n}$$

- sample size for  $(\varepsilon, \delta)$ -approximation:

$$r \geq \frac{1}{2\varepsilon^2} \left( \ln n + \ln 2 + \ln \frac{1}{\delta} \right)$$

# What's wrong with this?

Size depends on  $\ln n$

- loose, due to union bound
- not the right quantity
- should be characteristic quantity of graph

At each iteration, algorithm performs SSSP

- full exploration of the graph (no locality)

# What can we do?

## Our algorithm [RiondatoK14]

- uses VC-dimension
- sample size depends on vertex-diameter of  $G$
- at each step, single s-t shortest path computation
  - fewer edges touched
  - more locality
  - can use bidirectional search

# Our algorithm

- $\text{VD}(G) \leftarrow$  **vertex-diameter** of  $G$
- $r \leftarrow (1/2\varepsilon^2)(\lfloor \log_2(\text{VD}(G) - 2) \rfloor + 1 + \ln(1/\delta))$
- $\tilde{\mathbf{b}}(v) \leftarrow 0, \forall v \in V$
- **for**  $i \leftarrow 1, \dots, r$ 
  - $(u, v) \leftarrow$  **random pair of vertices**
  - $\mathcal{S}_{uv} \leftarrow$  all SPs from  $u$  to  $v$  (Dijkstra, trunc. BFS, bidirect. Search)
  - $p \leftarrow$  random element of  $\mathcal{S}_{uv}$
  - $\tilde{\mathbf{b}}(w) \leftarrow \tilde{\mathbf{b}}(w) + 1/r, \forall w \in \text{Int}(p)$
- **output**  $\tilde{\mathbf{b}}(v), \forall v \in V$

# What is the vertex-diameter?

$VD(G)$ : max no. of vertices in a SP

$$VD(G) = \max\{|p| : p \in \mathcal{S}_G\}$$

- small in social networks

$G$  not weighted:  $VD(G) = \Delta_G + 1$

- otherwise no relationship in general

Computation:

- $G$  unweighted, undirected: 2-approx via SSSP
- otherwise: size of largest WCC



# What do we get?

- $\text{VD}(G) \leftarrow$  vertex-diameter of  $G$
- $r \leftarrow (1/2\varepsilon^2)(\lfloor \log_2(\text{VD}(G) - 2) \rfloor + 1 + \ln(1/\delta))$
- $\tilde{b}(v) \leftarrow 0, \forall v \in V$
- for  $i \leftarrow 1, \dots, r$ 
  - $(u, v) \leftarrow$  random pair of vertices
  - $\mathcal{S}_{uv} \leftarrow$  all SPs from  $u$  to  $v$  (Dijkstra, trunc. BFS, bidirect. Search)
  - $p \leftarrow$  random element of  $\mathcal{S}_{uv}$
  - $\tilde{b}(w) \leftarrow \tilde{b}(w) + 1/r, \forall w \in \text{Int}(p)$
- output  $\tilde{b}(v), \forall v \in V$

**Theorem:**  $(\tilde{b}(v))_{v \in V}$  is a  $(\varepsilon, \delta)$ -approximation

# How can we prove it?

Define rangeset

Define probability distribution

Define betweenness as probability estimation problem

Show upper bound to VC-dimension

- Bonus: show tightness + variants

Apply VC-dimension sampling theorem

# What are the ranges set and the probability?

$D = \mathbb{S}_G =$  **all SPs** in  $G$

**Let**  $T_v = \{p \in \mathbb{S}_G : v \in \text{Int}(p)\}$

$F = \{T_v, v \in V\}$

**Probability distribution**  $\pi$  **on**  $\mathbb{S}_G$ :

$$\pi(p_{uw}) = \frac{1}{n(n-1)} \frac{1}{\sigma_{uw}}$$

- algorithm samples paths according to  $\pi$

- $\pi(T_v) = \frac{1}{n(n-1)} \sum_{p_{uw} \in T_v} \frac{1}{\sigma_{uw}} = \frac{1}{n(n-1)} \sum_{p_{uw} \in \mathbb{S}_G} \frac{\mathbb{1}_{p_{uw}}(v)}{\sigma_{uw}} = \mathbf{b}(v)$

# What is the VC-dimension of our rangeset?

**Theorem:**  $VC(\mathbb{S}_G, F) \leq \lfloor \log_2 VD(G) - 2 \rfloor + 1$

## Proof

- To shatter  $A \subseteq \mathbb{S}_G$ ,  $|A| = d$ ,
  - need  $2^d$  different ranges
  - any  $p \in A$  must appear in  $2^{d-1}$  different ranges
- Any  $p$  appears only in the ranges  $T_v$  such that  $v \in \text{Int}(p)$
- i.e., it appears in  $|\text{Int}(p)| \leq VD(G) - 2$  ranges
- To shatter  $A$ , must be  $2^{d-1} \leq VD(G) - 2$

# How to use the bound?

$\tilde{b}(v)$  = empirical average for  $b(v)$   
Sampling done according to  $\pi$   
Know upper bound to  $VC(\mathbb{S}_G, F)$  } We can apply the **VC sample theorem**

$$\text{If } r \geq \frac{1}{\varepsilon^2} \left( \lfloor \log_2(\text{VD}(G) - 2) \rfloor + 1 + \ln \frac{1}{\delta} \right)$$

then  $(\tilde{b}(v))_{v \in V}$  is an  $(\varepsilon, \delta)$ -approximation:

$$\Pr \left( \exists v \in V : |\tilde{b}(v) - b(v)| > \varepsilon \right) < \delta$$

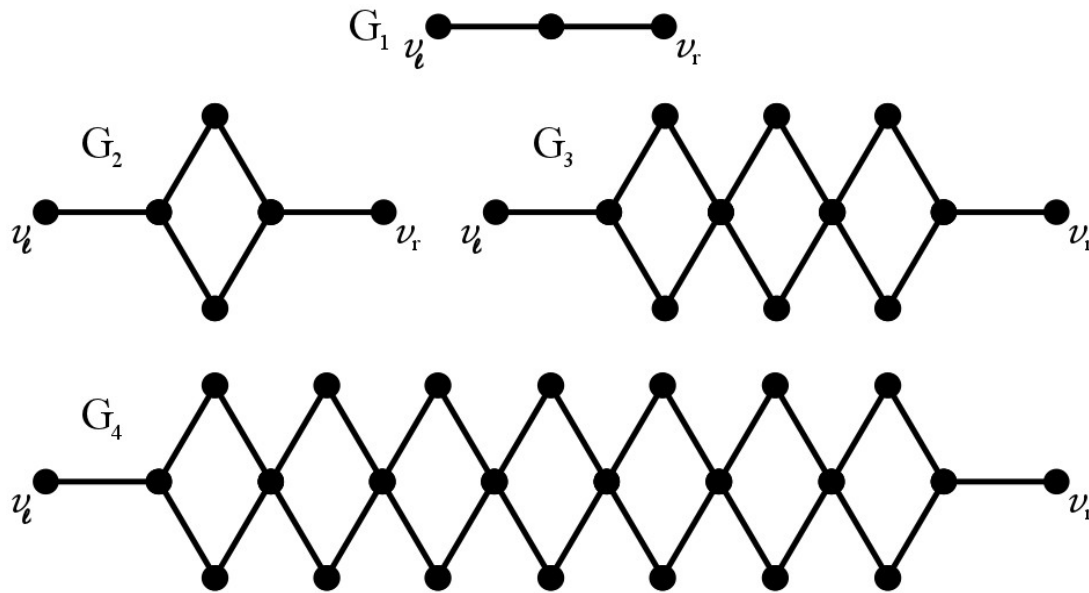
# Roadmap

- ✓ Define rangeset
- ✓ Define probability distribution
- ✓ Define betweenness as probability estimation problem
- ✓ Show upper bound to VC-dimension
  - Bonus: show tightness + variants
- ✓ Apply VC-dimension sampling theorem

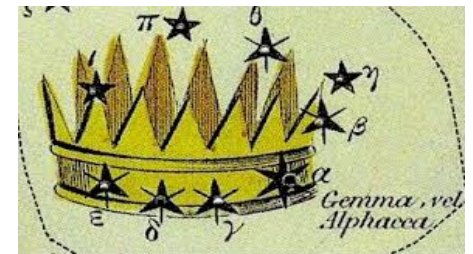


# Is the bound tight?

## Concertina graphs $(G_i)_{i \in \mathbb{N}}$



Concertina, musical instrument



Corona Borealis

**Theorem:**  $VC(\mathbb{S}_{G_i}) = \lfloor \log_2(\text{VD}(G_i) - 2) \rfloor + 1 = i$

# Is the vertex diameter the right quantity?

No.

If

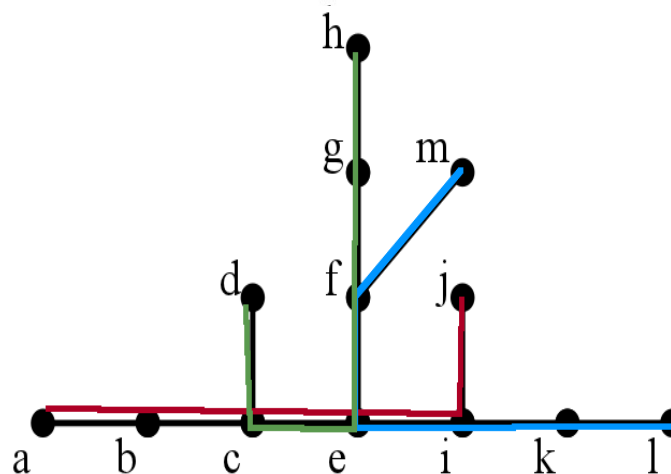
- $G$  is **undirected**
- for every connected pair  $(u, v)$  there is a **unique** SP, then

$$VC(\mathbb{S}_G, F) \leq 3$$

Proof: two SPs that meet and separate **can't meet again**

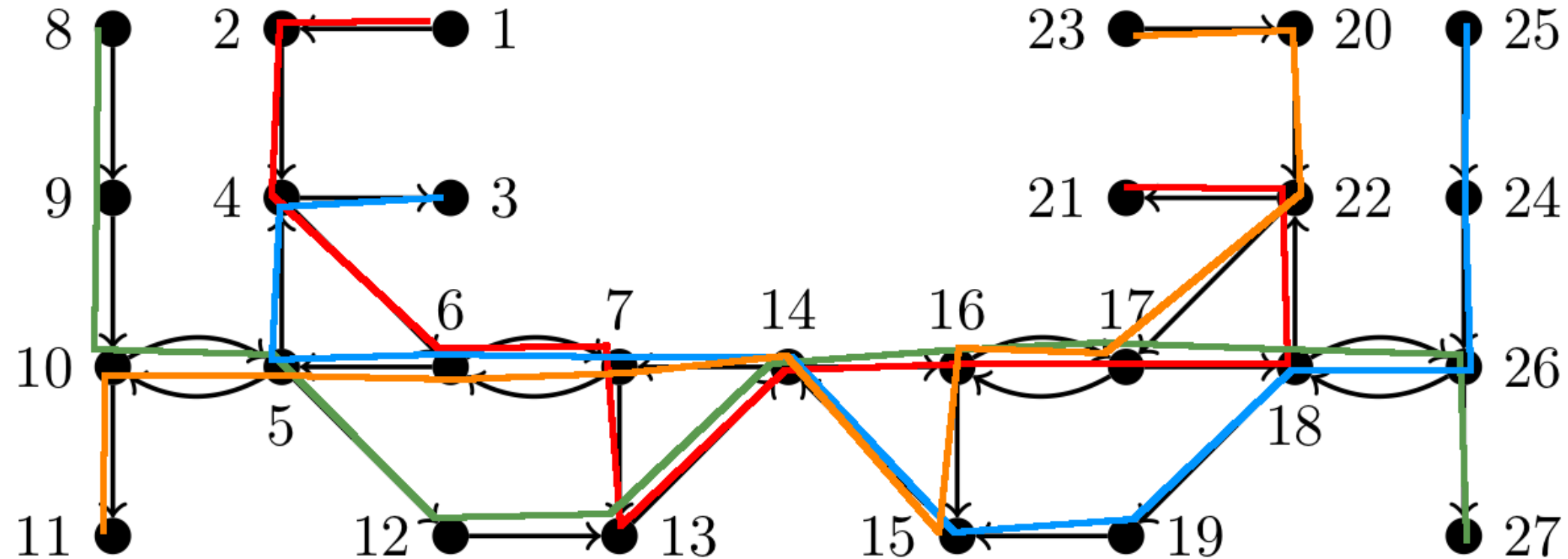
- + case analysis

Tight? Yes



# Is this true for directed graphs?

No. Can shatter 4 SP!



Open question: still a constant for directed graphs? (6?)

# What about relative guarantees?

$b^{(K)}$ :  $K^{\text{th}}$  highest betweenness, ties broken arbitrarily

**Top-K vertices:**  $T(K, G) = \{v \in V : b(v) \geq b^{(K)}\}$

**Relative approximation:**  $C = \{(v, \tilde{b}(v))\}$  s.t.

$$(1 - \varepsilon)b(v) \leq \tilde{b}(v) \leq (1 + \varepsilon)b(v), \forall v \in C$$

Algorithm:

- run additive approximation algorithm
- $\tilde{b}^{(K)} \leftarrow$  **lower bound** to  $b^{(K)}$
- use  $\tilde{b}^{(K)}$  and **relative-guarantees version of VC sample theorem** to compute sample size for relative approximation for  $T(K, G)$

# How good is the algorithm in practice?

C implementation as patch to igraph

Graphs: **real** (snap.stanford.edu) + **artificial** BarabasiAlbert

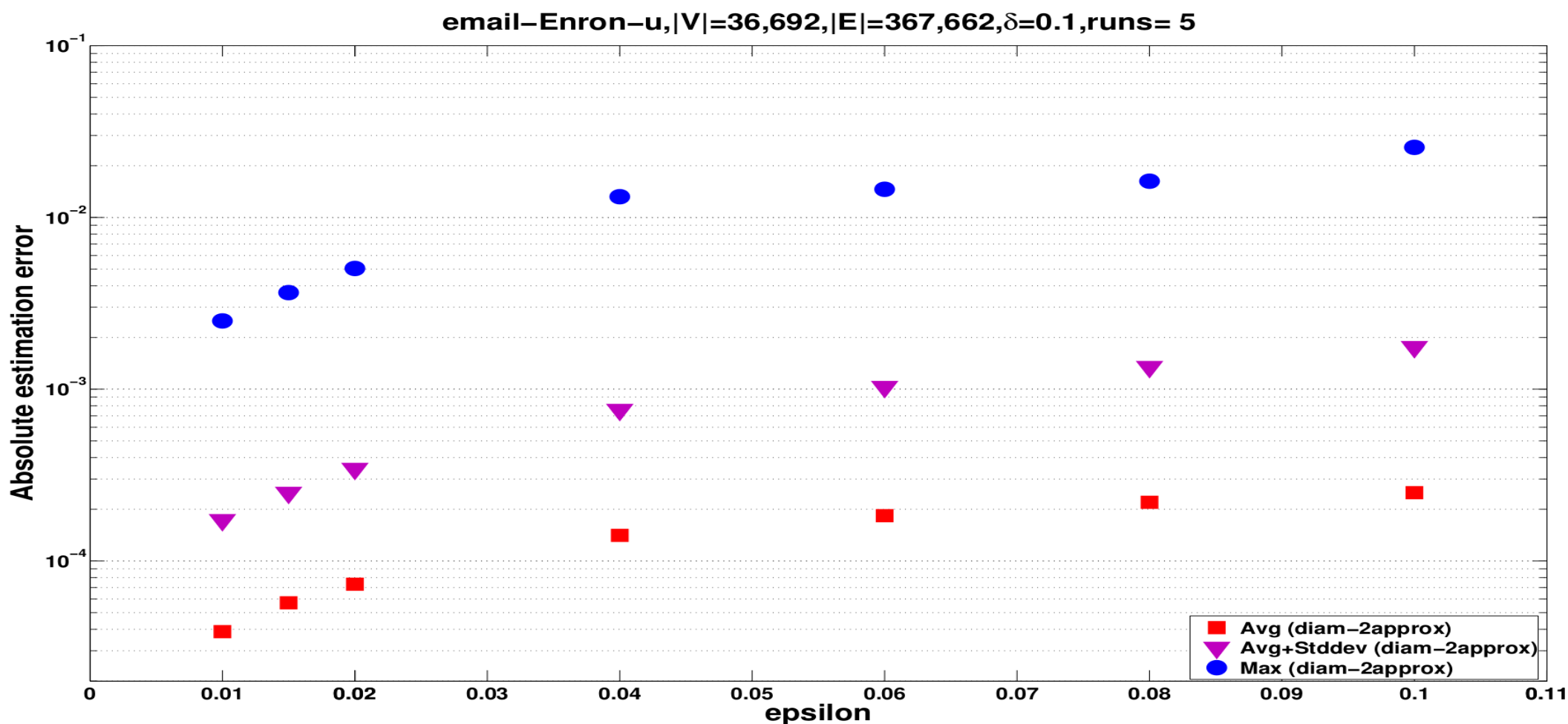
- social networks, road networks, ...

**Goals:** evaluate { accuracy  
speed  
scalability

# How accurate is our algorithm?

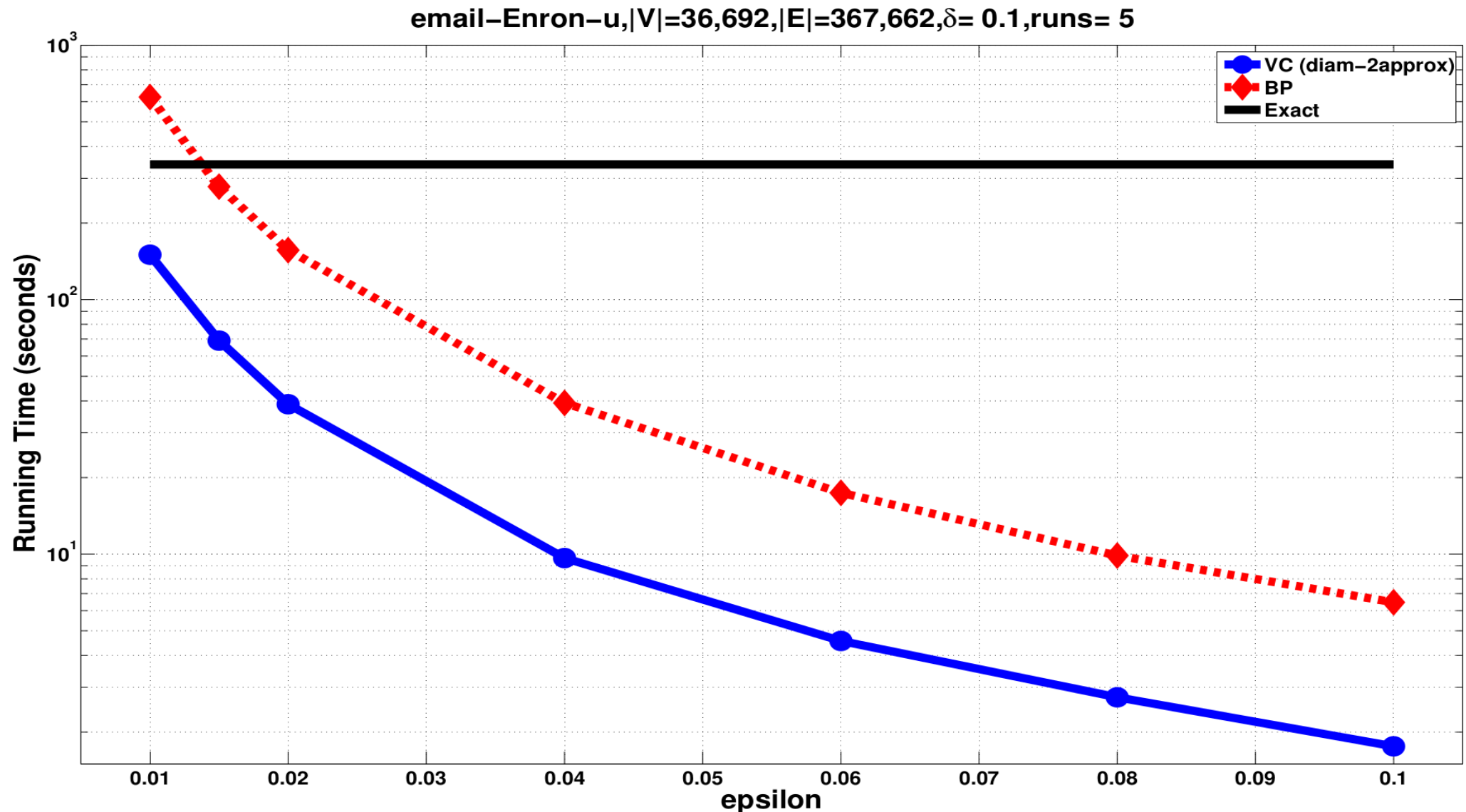
$|\tilde{b}(v) - b(v)|$  **always**  $\leq \varepsilon$  ( $O(10^3)$  runs on different graphs)

**Accuracy ~8x better** than guaranteed

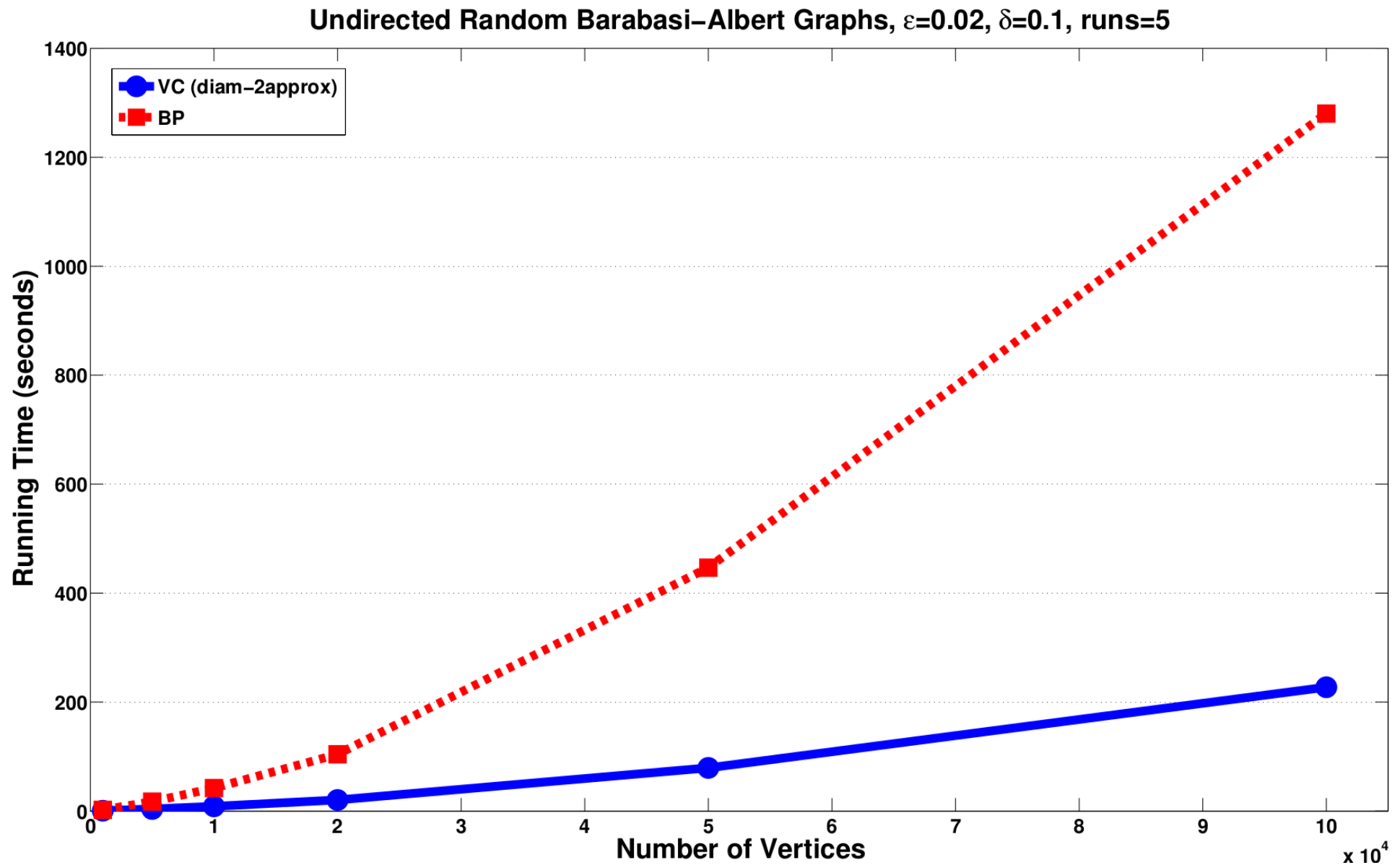


# How fast is our algorithm?

~8x faster than simple sampling algorithm



# How well does it scale?





# Am I telling the truth?

Yes, but.

- $\tilde{\mathbf{b}}_s(v)$  : estimator of simple sampling alg. w/ same no. of samples
- **Theorem:**  $\text{Var}[\tilde{\mathbf{b}}_s(v)] \leq \text{Var}[\tilde{\mathbf{b}}(v)], \forall v \in V$ 
  - does not imply that it computes a  $(\varepsilon, \delta)$ -approximation
- **Emphasis on different aspects:**
  - Ours: speed and scalability
  - Theirs: accuracy

# What did I show you?

Two sampling based algorithms for betweenness estimation

- Top-K algo is **first to achieve high relative guarantees**
- Much **smaller sample size** than previously known
- **Fewer computations** than existing work = faster

Characterizing graph problems through VC-dimension is challenging, but interesting

- and rewarding



Published at **ACM WSDM'14**, journal subm. in preparation

# Outline

## ✓ Introduction

- ✓ Problem
- ✓ Thesis statement
- ✓ Contributions
- ✓ VC-dimension



## ✓ Estimating betweenness centrality

- ✓ Rangeset and bounds
- ✓ Algorithms



## Conclusions

- Limitations of sampling
- Directions for further research



# What did we learn?

We can approximate many data analytics tasks using sampling

- size depends on **bound to VC-dim.**, not no. of questions (Variety)
  - characteristic quantity of the dataset / problem
- lower cost(Volume)
- sample **fits into memory** of single machine
  - can use **MapReduce** for boosting-like approach (many samples in parallel)
- can use “backwards” to derive **statistical tests** for false positives



# What are the limitations?

Need **efficient-to-compute** bound on VC-dimension

Need **efficient** sampling procedure

Need for **independent sampling**

- some new developments here

Dependency on  $\varepsilon$





# Where to go from here?

## Smaller samples

- pseudodimension, shatter coefficients, covering numbers, ...

## Progressive sampling

- Rademacher averages bounds

## Statistical testing

- False Discovery Rate rather than Family-Wide Error Rate

## New technology / computational platforms

- Spark, Pregel, ...

# Did we publish?

- R., Akdere, Çetintemel, Zdonik, Upfal. “The VC-dimension of SQL queries and selectivity estimation through sampling”. ECML-PKDD'11.
- R., Upfal. “Efficient discovery of Association Rules and Frequent Itemsets through sampling with tight performance guarantees”. ECML-PKDD'12, ACM TKDD'14.
- R., DeBrabant, Fonseca, Upfal. “PARMA: a parallel randomized algorithm for approximate association rule mining in MapReduce”. ACM CIKM'12.
- R., Vandin. “Finding the True Frequent Itemsets”. SIAM SDM'14.
- R., Kornaropoulos. “Fast approximation of betweenness centrality through sampling”. ACM WSDM'14
- Others:
  - Pietracaprina, R., Upfal, Vandin. “Mining top-k Frequent Itemsets through progressive sampling”. DMKD'10.
  - Akdere, Cetintemel, R., Upfal, Zdonik. “The case for predictive database systems: opportunities and challenges”. CIDR'11
  - Akdere, Cetintemel, R., Upfal, Zdonik. “Learning-based query performance modeling and prediction”. IEEE ICDE'12.
  - Pietracaprina, Pucci, R., Silvestri, Upfal. “Space-round tradeoffs for MapReduce computations”. ACM ICS'12

# Can you tell us more?

- 189 pages in 8 chapters
- 6 publications with 8 coauthors
- 211 references
- 9 quotes in 9 languages and 3 alphabets
- 180 GitHub commits (and counting)

algorithm approximation  
association betweenness bound  
collection data database dataset definition different  
distribution error estimation fi frequency  
frequent graph itemsets large lemma log  
mining number obtain pages pair Parma paths  
presented probability problem queries random  
range relative results rules sample  
selectivity shortest size space tables  
transactions values vc VC-  
dimension vertices work



# Who deserves all the credit?

- Eli
- Uğur, Basilis
- Andrea, Geppino, Stan, Rodrigo, Fabio, Francesco, Aris, Luca
- Olya, Andy, Justin, Evgenios, and all other PhDs
- Mackenzie, Michela, Marco, Bernardo, Robyn, Andrew, ...
- Lauren and astaff@
- tstaff@ for the grid + problems@

“Not he who begins, but he who keeps going”  
Leonardo da Vinci

