Distributed Data Mining: Current Pleasures and Emerging Applications

Hillol Kargupta
University of Maryland, Baltimore County and AGNIK
www.cs.umbc.edu/~hillol

Acknowledgements: Wes Griffin, Souptik Datta, Kanishka Bhaduri, Kamalika Das, Ran Wolff, Chris Giannella

Roadmap

- Distributed Data Mining: Why Bother?
- Some Emerging Applications
- Local Algorithms
  - Exact Local Algorithms
  - Approximate Local Algorithms
- Resources

Data Mining and Distributed Data Mining

- Data Mining: Scalable analysis of data by paying careful attention to the resources:
  - computing,
  - communication,
  - storage, and
  - human-computer interaction.
- Distributed data mining (DDM): Mining data using distributed resources.

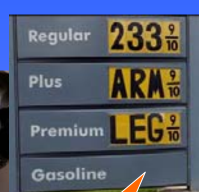
Data Mining for Distributed and Ubiquitous Environments: Applications

- Mining Large Databases from distributed sites
  - Grid data mining in Earth Science, Astronomy, Counter-terrorism, Bioinformatics
- Monitoring Multiple time critical data streams
  - Monitoring vehicle data streams in real-time
  - Monitoring physiological data streams
- Analyzing data in Lightweight Sensor Networks and Mobile devices
  - Limited network bandwidth
  - Limited power supply
- Preserving privacy
  - Security/Safety related applications
- Peer-to-peer data mining
  - Large decentralized asynchronous environments

Vehicles: Source of High Volume Data Streams

- Vehicles generate tons of data
- Hundreds of different parameters from different subsystems
- High throughput data streams
- So what?

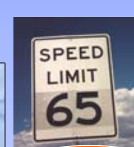
Why Mine Vehicle Data?



High gas prices

- Fuel consumption analysis
- Fleet analytics
- Vehicle benchmarking
- Predictive health-monitoring
- Driver behavior analytics

Breakdowns cost thousands of dollars



Bad driving costs money---fuel, brake shoe, insurance, law-suits

From Concept to Commercial Product

Circa 2001

Circa 2005

Circa 2007

- First prototype -- PDA-based platform
- Other choices:
  - Cell phones and
  - Low-cost, less powerful embedded devices

- Market Entry Point
  - Location management companies
  - M2M companies
- Low Cost Embedded GPS Devices
- Resource constrained
- 3-4K run time memory
- 250K footprint
- Resource sharing with GPS program

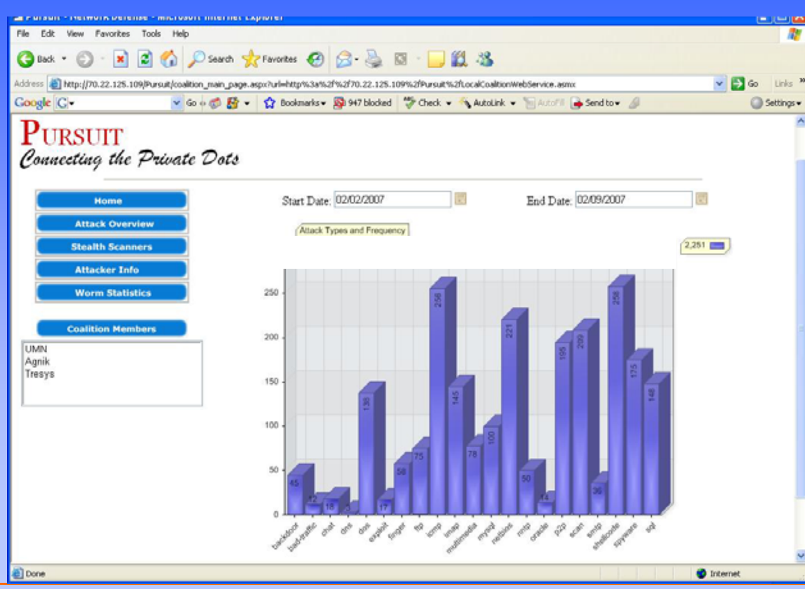
Private & Secure Data Mining from Multi-Party Distributed Data

- Compute global patterns without direct access to the multi-party raw distributed data
- Minimize communication cost
- Must come with provably correct guarantees with respect to a given privacy model
- Must be scalable with respect to
  - number of data sites
  - size of the data
- Privacy-preserving data mining
  - Blends in ``pattern-preserving'' transformations with data analysis

How PURSUIT Works for the User

- Need to have your own sensor such as SNORT, MINDS
- Download PURSUIT plug-in for the sensor and install
- PURSUIT plug-in offers
  - A stand-alone interface for processing your alerts from the sensor and cross-domain analysis
  - Web account for detailed cross-domain statistics
  - Optional distributed collaboration management module for managing the threats and archiving forensics

PURSUIT Web Site



Peer-to-peer (P2P) Networks

- Relies primarily on the computing resources of the participants in the network rather than a relatively low number of servers.
- P2P networks are typically used for connecting nodes via largely ad hoc connections.
- No central administrator/coordinator
- Peers simultaneously function as both "clients" and "servers"
- Privacy is an important issue in most P2P applications

Where do we find P2P Networks?

- Applications:
  - File-sharing networks: KaZAa, Napster, Gnutella
  - P2P network storage, web caching,
  - P2P bio-informatics,
  - P2P astronomy,
  - P2P Information retrieval
- P2P Sensor Networks?
- P2P Mobile Ad-hoc NETwork (MANET)?
- Next Generation:
  - P2P Search Engines, Social Networking, Digital libraries, P2P "YouTube"?

P2P Web Mining

- Web mining in a sever-less environment

Useful Browser Data

- Web-browser history
- Browser cache
- Click-stream data stored at browser (browsing pattern)
- Search queries typed in the search engine
- User profile
- Bookmarks

- Challenges
  - Indexing, clustering, data analysis in a decentralized asynchronous manner
  - Scalability
  - Privacy

P2P NASA Astronomy Data Mining

- Virtual Observatories
  - Client-server architecture
  - Consider Sloan Digital Sky Survey:
    - 2M hits per month
    - traffic is doubling every 15 months
  - Need better scalability
- MyDB: Download and locally manage your data
- Network of such databases
- Searching, clustering, and outlier detection in P2P virtual observatory data network.
- NASA AIST Project at UMBC

DDM Applications: Typical Characteristics

- Distributed computing environment
- Heterogeneous communication links with bandwidth constraints
- Distributed data
- Continuous data streams
- Multi-party data, sometimes privacy sensitive (difficult to centralize)
- Server-free networks
- Resource constraints (e.g. energy consumption)

Data Communication

- Case I: Participating nodes are connected by high speed networks and efficient redistribution of data is possible.
- Case II: Nodes are connected by low speed networks and data redistribution is difficult to support.
- Case III: Combination of I and II.

Global Function Computation

- Each vertex v in a graph holds an input X_{v_i}
- Compute some global function $f(X_{v_1}, X_{v_2}, \ldots .. X_{v_n})$

Locality Sensitive Computation

- Global vs. Local
- Main problems of the global algorithms:
  - Every node needs to maintain information about the entire network
  - Maintaining this information is resource intensive for large networks

Data Mining as Function Computation

- Most data mining problems can be viewed as function computations
- Examples
  - Classification
  - Predictive modeling
  - Clustering
  - Outlier detection

DDM: Defining the Problem

- Let $G=(V, E)$ be a graph
- Let Ω_k be the set of all neighboring nodes of the k-th node $v_k \in V$

- Need a decomposable representation where $f(V)$ can be computed from "locally" computed functions $\Phi_k(\Omega_k)$
- Example:

$$f(V) = \sum_k w_k \Phi_k(\Omega_k)$$

Homogeneous Data Sites

Account Number	Amount	Location	History	Earning
11992346	99.84	Seattle	Good	High
12999333	29.33	Seattle	Good	High
45633341	34.89	Portland	Okay	Low
55567999	980	Spokane	Good	Low

Account Number	Amount	Location	History	Earning
87992364	20	Chicago	Good	Low
67845921	447	Urbana	Good	Low
85621341	19.78	Chicago	Okay	High
95345998	800	Peoria	bad	High

Different sites observe same features for different events

Heterogeneous Sites

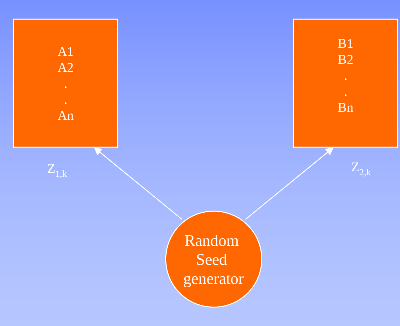
State	Movie	Rating	Revenue
WA	Hyper Space	A+	6M
ID	Once Upon a Time	B-	2M
BC	The King and the Liar	B+	8M
CA	The Shepard	A-	10M

City	State	Size	Avg. earning	Teen pop.
Lewiston	ID	Small	Low	5K
Spokane	WA	Medium	Medium	30K
Seattle	WA	Large	High	250K
Portland	OR	Large	High	200K
Vancouver	BC	Medium	Medium	199K

Different sites observing different feature sets

Distributed Randomized Inner Product Computation

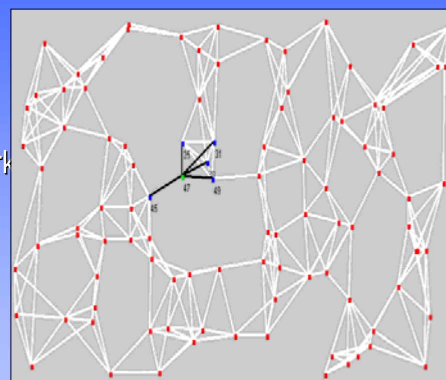
Node 1 Node2



- Node 1 computes $Z_{1,k}$
  - $Z_{1k}=A1.J_1+..+An.J_n$
  - $J_i \in \{+1,-1\}$ with uniform probability
- Node 2 calculates $Z_{2,k}$
  - $Z_{2k}=B1.J_1+..+Bn.J_n$
- Compute $z_{1,k}.z_{2,k}$ for a few times and take the average

Locality Sensitive Distributed Algorithms

- Global algorithms: Know everything about the entire network
  - Every node needs to maintain information about the entire network
  - Maintaining this information is resource intensive for large networks
- Local algorithms: Communicate only with the local neighborhood.
- Does locality imply efficiency?



Bounded Communication Local Algorithms

- Every node communicates with its local neighborhood bounded by path-length of α
- In addition, the total amount of communication with its neighbors is also bounded by some γ
- $(\alpha, \gamma)-$ Local algorithms

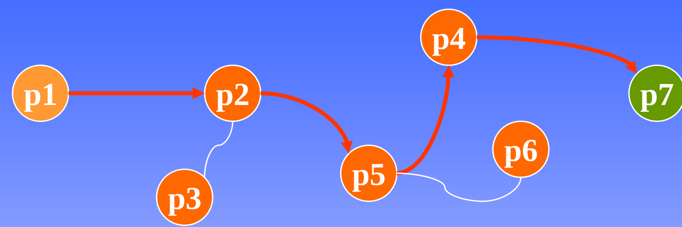
Approaches

- Functions computation through decomposable representations
  - Approximations
    - Randomized techniques
    - Sampling-based approximations
    - Variational approximations
  - Exact decompositions
    - Deterministic techniques

Approximation

- Estimate $\Phi_k(\Omega_k)$
  - Cardinal sampling
  - Ordinal relaxation
    - Interested in constructing an ordering
    - Find the ones that rank high

Sampling in Distributed Environments



- Uniform data sample often good representative of data
- Collecting uniform sample in asynchronous networks is challenging
  - Varying degrees of nodes
  - Skewed data distribution

Challenges

- How to collect random-uniform sample of data from an asynchronous network?
- How to make sampling communication efficient and fast?
- Asynchronous algorithms

Varying Degrees of Connectivity in Large Communication Networks

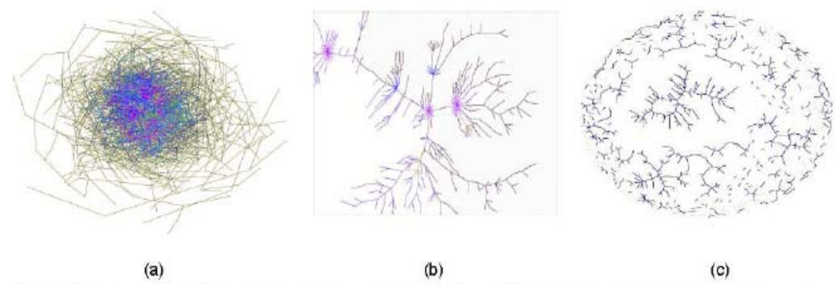


Fig. 17. *Left*: Topology of the Gnutella network as of February 16, 2001 (1771 peers); *Middle*: Topology of the Gnutella network after a random 30% of the nodes are removed; *Right*: Topology of the Gnutella network after the highest-degree 4% of the nodes are removed

Source: *Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble*. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia Syst.*, *2003*

Problem Definition: Uniform Data Sampling

- Data is homogeneously distributed among peers

  $X = X^1 \cup X^2 \cup \ldots \cup X^n$
- Data distribution is non-uniform

  $|X^i| \neq |X^j|$ for $i \neq j$
- Uniform sampling of peers results in biased data sampling
- Problem: How to collect a uniform-random sample x of total data X from the network?

Random Walk and Markov Chain

- Random walk on Graph: visits nodes in a sequence where at each step, the next destination node is selected using transition probability of current node – Markov process

$$\pi(t+1)^T = \pi(t)^T P$$

  - P = Transition Probability Matrix
  - π (t) = Probability Distribution of State at *t*
- i^{th} element of stationary distribution $\pi_i = d_i/2m$
- Mixing-time of Markov Chain: Length of walk to converge to stationary distribution [Sinclair, 1992]

$$\tau = O\left(\log (n)/(1-|\lambda_2|)\right)$$

  - $|\lambda_2|$= SLEM (Second Largest Eigenvalue Modulus) of P
  - Aperiodic graphs

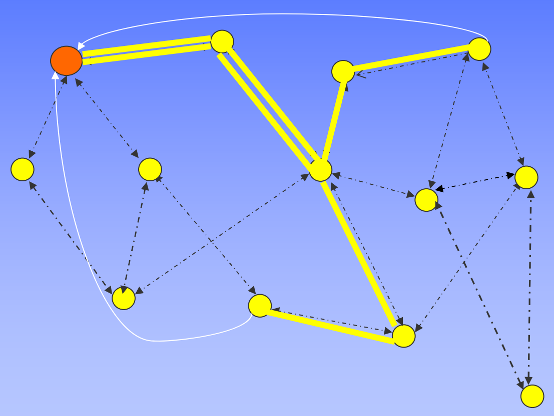
Uniform Sampling of Peers in P2P

- Random walk with degree correction helps uniform sampling of peers
  - Maximum Degree
  - Metropolis-Hasting
  - Random-weight Distribution
- Metropolis-Hasting:

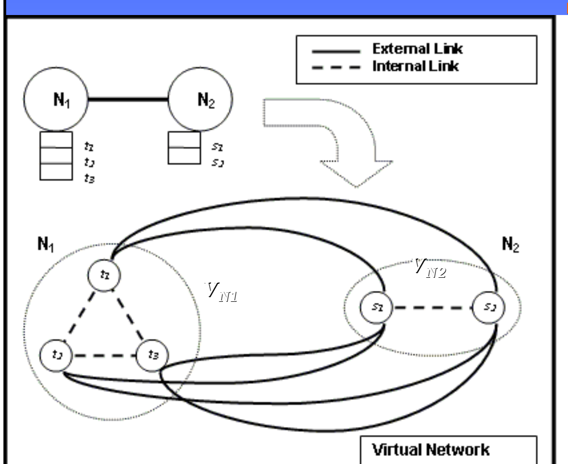
$$p_{ij}^{mh} = \begin{cases} 1/Max(d_i, d_j) & \text{if } i \neq j \text{ and } j \in \Gamma^{(i)} \\ 1 - \sum_{j \in \Gamma^{(i)}} p_{ij}^{mh} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

- Source node applies modified r.w. of length L_{walk} to pick-up one peer uniformly
  - Walk-length L_{walk} = *O* (*log* (Total Network Size))

Sampling of Peers by Random Walk



Data Sampling Concept: Virtual Network



- Transform to virtual network with single data-tuple per virtual node
- Data-tuples held by same real node are *'fully-connected'*.
- Apply Metropolis-Hasting on virtual network.
- Communication saved on 'virtual- walk' on *internal* links.

Metropolis-Hasting on Virtual Graph

- To achieve uniformity, P should meet the following conditions on virtual graph

$$P\mathbf{1} = \mathbf{1}, \mathbf{1}^T = \mathbf{1}^T P, P \geq 0, P = P^T,$$

Symmetric, Non-negative, Double-stochastic

- Transition probability between data-tuple K and L on virtual graph:

$$p^V_{KL} = \begin{cases} 1/Max((n_{i,K\in V_{N_i}} - 1 + \sum_{g\in\Gamma^{(i)},K\in V_{N_i}} n_g), \\ (n_{j,L\in V_{N_j}} - 1 + \sum_{h\in\Gamma^{(j)},L\in V_{N_j}} n_h)) \\ \quad \mathit{if}\ L \neq K\ \mathit{and}\ \overline{E}_{KL} \in \overline{E} \\ 1 - \sum_{\hat{L}\in\bar{\Gamma}^{(K)}} p^V_{K\hat{L}}\ \ \mathit{if}\ L = K \\ 0\ \ \mathit{Otherwise}, \end{cases}$$

Algorithm

- Initialization: Each node N_i knows
  - Immediate neighborhood : $\Gamma^{(i)}$
  - Total data-size of neighbors: $\sum_{j\in\Gamma^{(i)}} n_j$
- Transition Probability on real graph at N_i

$$\hat{p}_{ij} = \begin{cases} n_i/(n_i - 1 + \sum_{g\in\Gamma^{(i)}} n_g) \\ \text{probability of going to another} \\ \text{data tuple randomly in current node } N_i \\ n_j/Max((n_i - 1 + \sum_{g\in\Gamma^{(i)}} n_g), \\ (n_j - 1 + \sum_{h\in\Gamma^{(j)}} n_h)) \\ \text{probability of going to a data tuple in node } N_j \\ 1 - \sum_{j\in\Gamma^{(i)}} \hat{p}_{ij} \text{ probability of doing nothing} \\ 0 \text{ for } N_j\text{-s not in } \Gamma^{(i)} \end{cases}$$

Case I

Case II

Case III

Performance Analysis

- Arbitrarily selected 'source-node' (N_S) launches *s* random walks
- Random-walk terminates after L_{walk} steps

$$L_{walk} = O(\log(\text{Datasize})/1-|\lambda_2|)$$

- The data-tuple t_i on which walk terminates marked as a uniform random sample
- t_i sent back to N_S

Estimating Random-walk length

- $L_{walk} = O\,(\log(\text{Datasize})/(\text{spectral gap}))$
  - Spectral gap = ($1-|\lambda_2|$)
- Source-node can over-estimate datasize
  - Logarithmic effect on the walk-length
- Computing 'spectral-gap' exactly is communication and computation intensive.
- A lower-bound of spectral-gap gives upper-bound of walk-length

Bounding the Spectral-gap

- Neighbor data ratio is important $\rho_i = \frac{\sum_{j \in \Gamma(i)} n_j}{n_i}$
- For a network of size *n*, lower bound of spectral gap

$$1 - |\lambda_2| \geq 2 - \sum_{i=1}^{n} \frac{1}{1+\rho_i}$$

- For each node, if $\rho_i \geq \rho_T$, a universal threshold value:

$$\frac{1}{1-|\lambda_2|} \leq \frac{1}{2 - \frac{n}{1+\rho_T}}$$

- If $\rho_i = O(n)$ for all nodes, then $\frac{1}{1-|\lambda_2|} = O(1)$
- Hence, $L_{walk} = O(\log(\text{Datasize}))$

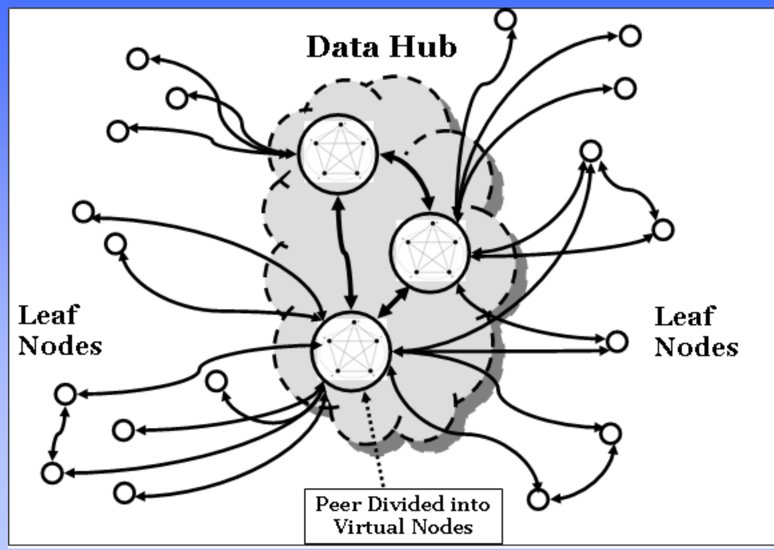
Effect on Communication Topology

- For all nodes N_i in the network, $\rho_i = O(n)$ implies :

Total Data Contained by Neighbors (n_j in Γ_i) $\geq O(n)$ times local data

- Real world network data distribution often follows power-law *("Measuring and Analyzing the Characteristics of Napster and Gnutella hosts" by Stefan Saroiu et. al. , 2003)*
  - Majority of the data content by few peers forming a 'data-hub'
- Peers with small amount of local data connecting to 'data-hub' achieves $O(n)$ neighbor data ratio
  - Communication topology: A central hub consisting of few peers sharing most of the data, and rest of the peers sharing few data are directly connected to this hub.

Communication Topology



Communication Complexity

- Communication cost
  1. Discover a uniform sample
  2. Transport sampled data to N_S
- Assumption: Network protocol takes care of the peer-to-peer communication between two nodes
- P2P-Sampling Initialization Cost: $2\times|E|$ integer bytes
- Communication to discover one sample

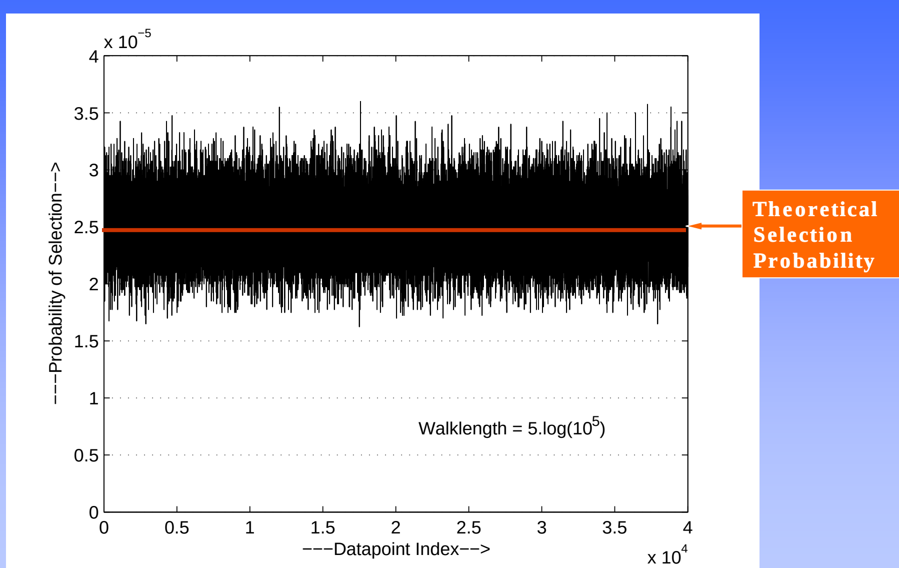
  $ά \times L_{walk} \times (đ+2)$ integer bytes

  - đ = Average degree of connectivity = Constant
  - ά = Average probability of going to a different node in one step of random walk ($1 \geq ά \geq 0$)

Experimental Setup

- Network topology generated by
  - BRITE (Boston University Representative Internet Topology Generator)
- Router level *Barabasi-Albert* model for power-law topology
- P2P network with 1,000 and higher nodes
- Total data = 40×network size
- Arbitrarily selected node conducts P2P-Sampling
- Data distribution: Non-uniformly distributed

Uniformity of Sampling



Ordinal Relaxation

- Let X be a continuous random variable
- Let ξ_p be the population percentile of order p, i.e. $\Pr\{x \le \xi_p\} = p$
- Let $x_1<x_2<\ldots<x_N$ be N independent samples from X
- We have

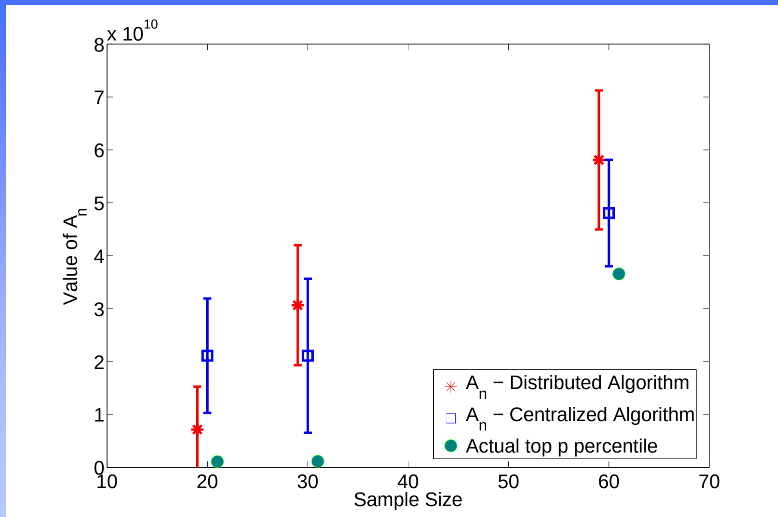
$$\Pr\{x_N > \xi_p\} > q \Rightarrow N \ge \left\lceil \frac{\log(1-q)}{\log p} \right\rceil$$

- Example:
  - q=95% and p =80% ➔ N=14
  - If we took 14 independent samples from any distribution, we can be 95% confident that 80% of the population would below x_{14}.

Ordinal Inner Product Computation

- Each node has a vector X_i
- Compute the Inner Product Matrix
  - Every node needs X_i from every node.
- How about finding just the top-k entries of the inner product matrix?

Ordinal Identification of Significant Entries from the Inner Product Matrix



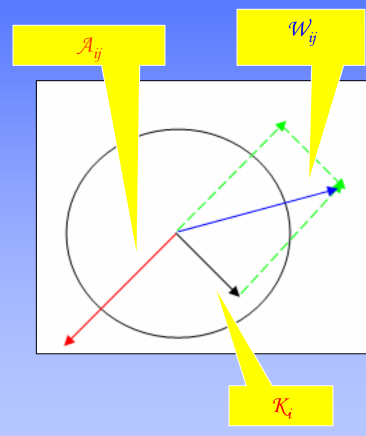
Bhaduri, Das, Kargupta. (2006). An Ordinal Approach for Detecting Feature-Interaction in a Peer-to-Peer Network

Majority Vote Computation Algorithm

- Each node has a number
- Check if the summation of the numbers at all nodes is greater than or equal to 0.
- Another variant: Check if the sum is greater than a certain threshold.

Notations

- $P_1, \ldots, P_n$ – set of peers
- P_i's local vectors -
  - $\mathcal{S}_i$ – data at time t
  - $\mathcal{X}_{ij}$ – sent by P_i to P_j
  - $\mathcal{K}_i$ – *knowledge* $(\mathcal{S}_i + \Sigma \mathcal{X}_{ji})$
  - $\mathcal{A}_{ij}$ – *agreement* $(\mathcal{X}_{ij} + \mathcal{X}_{ji})$
  - $\mathcal{W}_{ij}$ – ***withheld*** $(\mathcal{K}_i - \mathcal{A}_{ij})$
  - $\mathcal{G}$ – average of all peers



All vectors computations are local to a peer

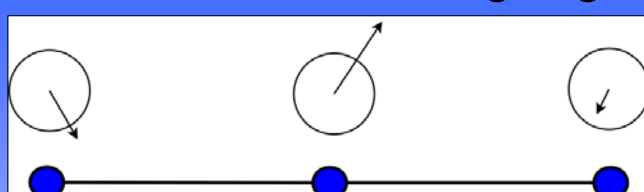
One-dimensional Example: Majority Vote

- Input to P_i : a real number $(\mathcal{S}_i)$
- Goal: Find if $\Sigma \mathcal{S}_i > 0$
- Output: 1 if $\mathcal{K}_i > 0$, 0 otherwise
- Simple stopping rule:
  - If $(\mathcal{A}_{ij} > 0$ and $\mathcal{A}_{ij} > \mathcal{K}_i)$ ⇨ Communicate
  - If $(\mathcal{A}_{ij} < 0$ and $\mathcal{A}_{ij} < \mathcal{K}_i)$ ⇨ Communicate
- If communicate
  - Set $\mathcal{X}_{ij} = \mathcal{K}_i - \mathcal{X}_{ji}$

Applications: L2 Norm Monitoring

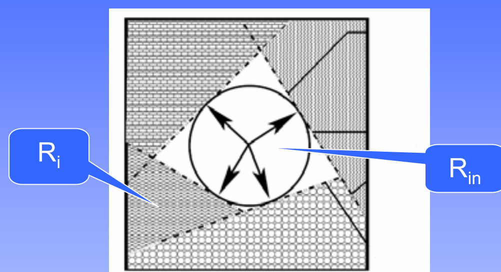
- Initial setup: each peer has
  - A data vector
- Monitoring Problem:
  - is $||\mathcal{G}||<\varepsilon$?

Local L2 Norm Monitoring Algorithm



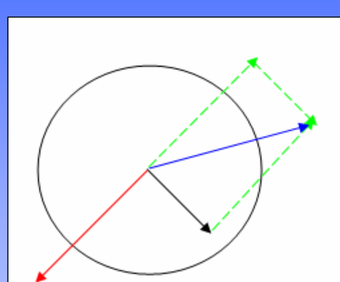
- Initial setup: each peer has
  - A data vector
  - Some global pattern vector
- Monitoring Problem:
  - is the L2 norm of the distance between the average data vector and the pattern vector greater than a given constant ε
- Applications:
  - Centroid monitoring
  - Eigenvector monitoring

Specifications: L2 Norm Monitoring



- Region of $\mathcal{F}$ true (R_{in}): inside circle, convex
- Region of $\mathcal{F}$ false: outside circle, non-convex convex
- Use tangent planes (R_i) to cover domain
- $C = \{R_{in}, R_1, R_2, \ldots\}$

Local Vectors



- For peer P_i
  - Own estimate of global average (X)
  - Agreement with neighbor P_j (Y)
  - Withheld knowledge w.r.t neighbor P_j (Z=X-Y)

Theorem

- If for every peer and each of its neighbours both the agreement and the withheld knowledge are in a convex shape (here a circle) - then so is the global average
- Wolff, Bhaduri, Kargupta, 2005

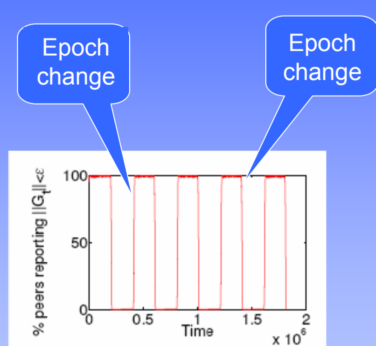
L2 Experimental Setup

- Simulator
  - Distributed Data Mining Toolkit (DDMT)
- Topology
  - BRITE Internet Topology generator
  - Realistic edge delays
- Input data
  - Mixture of correlated Gaussians in R^d with 10% noise
- Epoch change: Change of the means of Gaussians at fixed time intervals

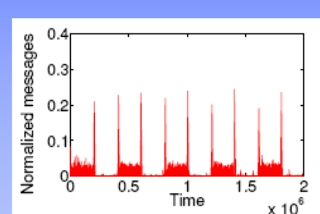
L2 Experimental Setup

- Quality: Percentage of peers correctly computing alert –
  - $\|\mathcal{K}\|<\varepsilon$ when $\|\mathcal{G}\|<\varepsilon$
  - $\|\mathcal{K}\|>\varepsilon$ when $\|\mathcal{G}\|>\varepsilon$
- Cost: Messages per peer per unit of leaky bucket

L2 Experiments: Results



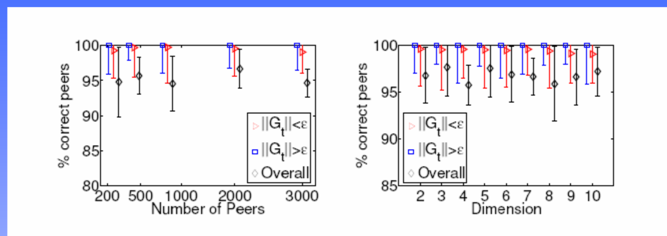
Quality



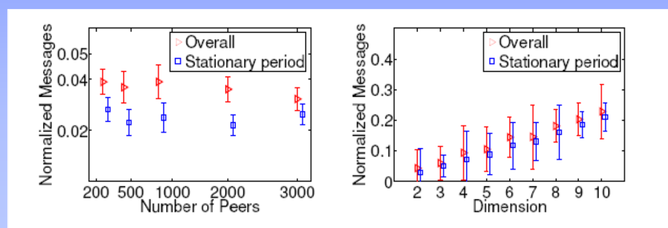
Cost

For broadcast-based algorithms normalized messages = 2

L2 Experiments: Scalability



Quality



Cost

Resources

- DDMWiki (http://www.umbc.edu/ddm/wiki/)
- DDMBib (http://www.cs.umbc.edu/~hillol/DDMBIB/)
- Full DDM Course Web Site (http://www.cs.umbc.edu/~hillol/CLASSES/DDM)