Monitoring Large, Distributed, Dynamic Systems

Systor '11, IBM Haifa

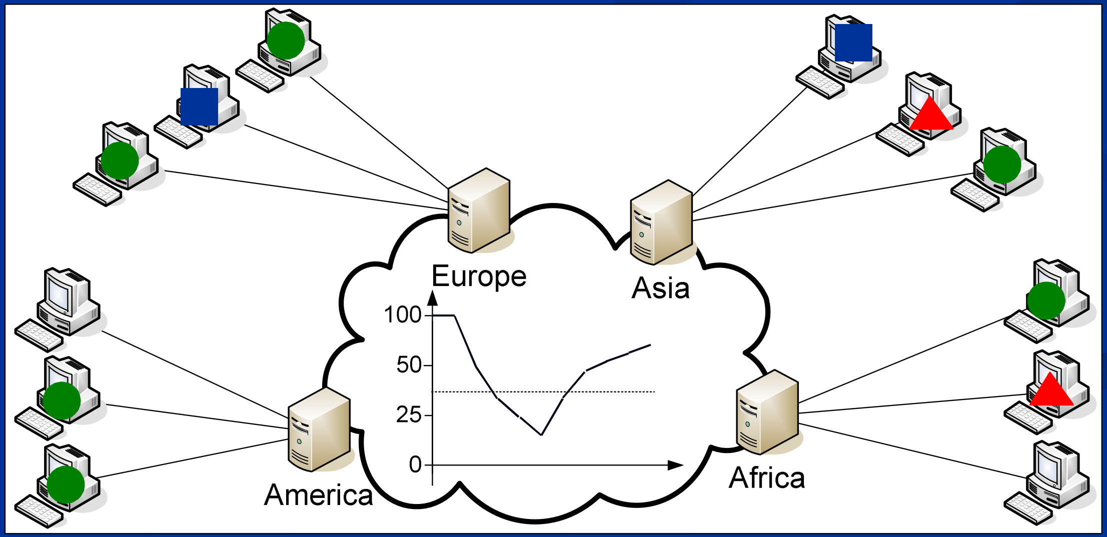
Izchak Sharfman, Guy Sagy, Avishay Livne, Amir Abboud, Daniel Keren, Assaf Schuster

- *A Geometric Approach to Monitoring Distributed Data Streams.* SIGMOD'06 (Honorable Mention for Best Paper Award), ACM TODS'07
- *Aggregate Threshold Queries in Sensor Networks.* IPDPS'07
- *Shape Sensitive Geometrical Monitoring.* PODS'08, IEEE TKDE 2011
- *Top-k Vectorial Aggregation Queries in a Distributed Environment.* JPDC 2010
- *Distributed Threshold Querying of General Functions by a Difference of Monotonic Representation.* PVLDB'11
- Optimal Local Constraints for Distributed Stream Monitoring, submitted.
- EU FP7 Project "LIFT" *(USING LOCAL INFERENCE IN MASSIVELY DISTRIBUTED SYSTEMS)*
- EU FP7 Project "DATA SIM" *(DATA SCIENCE FOR SIMULATING THE ERA OF ELECTRIC VEHICLES)*

Web Page Frequency Counts

- Mirrored web site
- Mirrors record the frequency of requests for pages
- Detect when the global frequency of requests for a page exceeds a predetermined threshold

● Req #1
■ Req #2
▲ Req #3



Running example: monitoring cloud health

- Small computer cloud (two machines)
- Want to submit an alert when **average** load is ≥ 80%.
- But – don't want to keep reporting individual loads.
- Trivial solution: every machine keeps silent as long as its load is < 80%.
- Better – set different thresholds (robust machine reports when load ≥ 90%, weak machine when load ≥ 70%).

We're scraping the surface of the *Distributed Monitoring Problem*

- Slightly modifying the problem makes it far harder…
- Want to monitor *uniformity*
- Three machines, loads are $L_1, L_2, L_3 \;\; (\overline{L} = \text{average load})$
- Want to submit an alert when

$$\left(\overline{L} - L_1\right)^2 + \left(\overline{L} - L_2\right)^2 + \left(\overline{L} - L_3\right)^2 \geq T$$

- *Local conditions* for submitting an alert are harder to define!

- Loads at all machines may be increasing but in a uniform fashion, hence no alert should be sent, etc.

- A huge range of problems…

- Simplest – average (linear function) of two scalar parameters.
- Most general and difficult – many nodes, each holds a (dynamic) vector of parameters, need to monitor a general function of them all. The value of this function may indicate a problem, an abnormality, or a phase change.

The need has been there for a while!

- "More than Sum: Complex Aggregates. Network aggregation infrastructures can support a large class of functions for merging data... Up to now we have focused on SUM... Standard database languages offer other aggregates including AVERAGE, STDEV, MAX and MIN. Given a constraint on one of these global aggregates **(e.g. 'ensure that the STDEV of latency is < 1 second')**, it is not immediately clear what ***local event*** should trigger global constraint checks."
  – From "A Wakeup Call for Internet Monitoring Systems: The Case for Distributed Triggers", Ankur Jain, Joseph M. Hellerstein, Sylvia Ratnasamy, David Wetherall, **HOTNETS04**.

- No general solution yet.
- Required: a paradigm for compiling *local* conditions at the nodes, such that:
  - Every *global* event is captured (i.e. it results in the violation of at least one local condition).
  - Communication is minimal.
- Why? Reduce communication, avoid false alerts, maintain privacy…

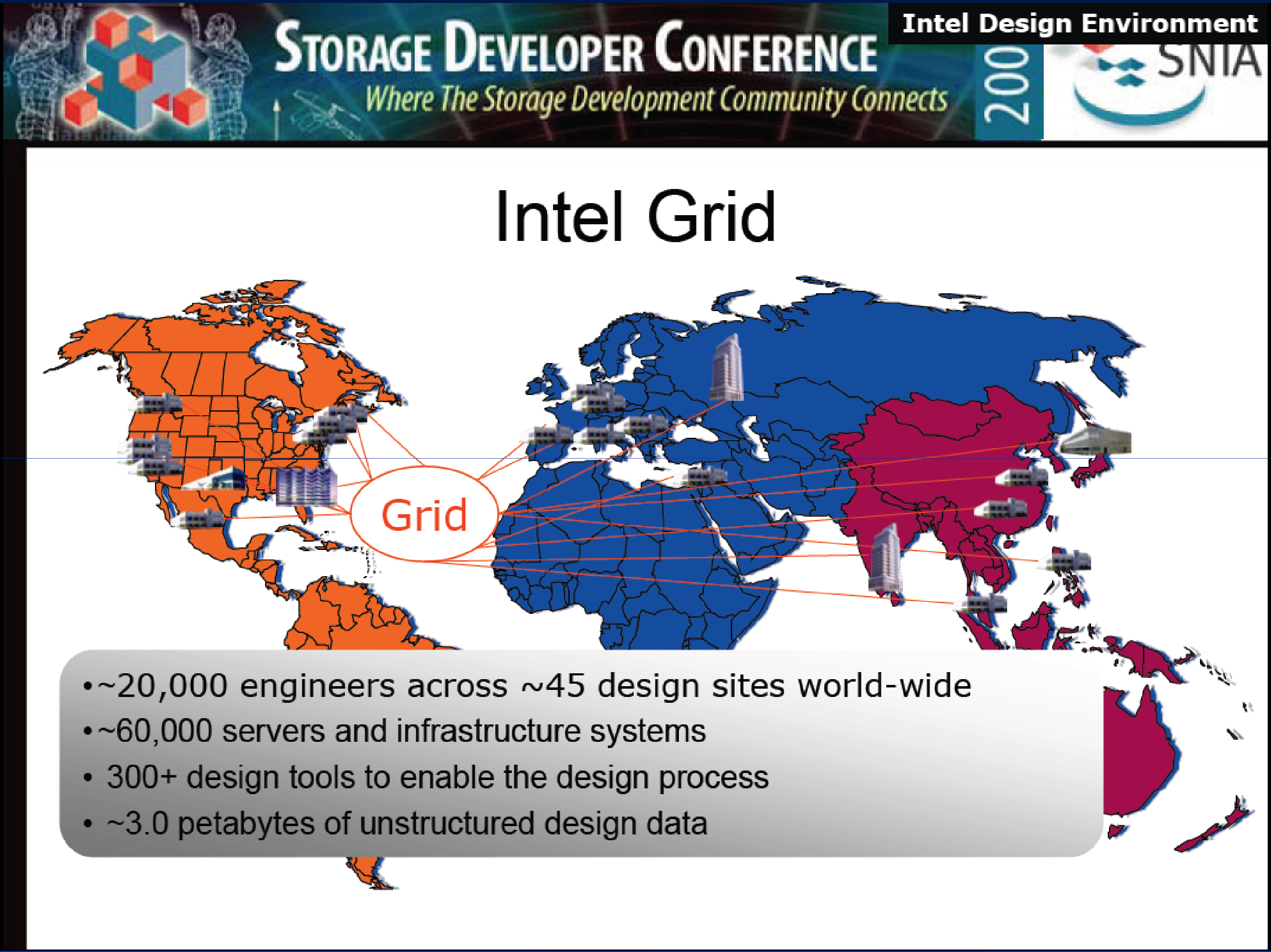
Problem Definition – Streams

- A set of data sources
  - Distributed
  - Dynamic
- A data vector is collected from each stream
- Given:
  - A function over the union of data vectors
  - A threshold *T*
- Continuous query: alert when the GLOBAL function crosses *T*
- Goal: minimize communication during query execution

Search Engines

- Distributed datacenter/warehouse
  - "Our logs are larger than any other data by orders of magnitude. They are our source of truth." Sridhar Ramaswamy. SIGMOD'08 keynote on "Extreme Data Mining"
- Mining the logs: Compute pairs of keywords for which the correlation index is high
- "Network bandwidth is a relatively scarce resource in our computing environment". Dean and Ghemawat. MapReduce paper, OSDI'04

Intel Design Environment



STORAGE DEVELOPER CONFERENCE

Where The Storage Development Community Connects

200

SNIA

Intel Grid

- ~20,000 engineers across ~45 design sites world-wide
- ~60,000 servers and infrastructure systems
- 300+ design tools to enable the design process
- ~3.0 petabytes of unstructured design data

Monitoring Cloud Health

- Amazon’s Elastic Compute Cloud – EC2
- Amazon’s Simple Storage Service – S3

Amazon Web Services » Service Health Dashboard

Amazon S3 Availability Event: July 20, 2008

Amazon S3 Availability Event: July 20, 2008

“At 8:40am PDT, error rates in all Amazon S3 datacenters began to quickly climb and our alarms went off. By 8:50am PDT, error rates were significantly elevated and very few requests were completing successfully. By 8:55am PDT, we had multiple engineers engaged and investigating the issue. Our alarms pointed at problems processing customer requests in multiple places within the system and across multiple data centers. While we began investigating several possible causes, we tried to restore system health… At 9:41am PDT, we determined that servers within Amazon S3 were having problems… By 11:05am PDT, all server-to-server communication was stopped, request processing components shut down, and the system's state cleared…. “

Ad-Hoc Mobile P2P Networks

Peer-to-peer network invites drivers to get connected

CarTorrent could smarten up our daily commute, reducing accidents and bringing multimedia journey data to our fingertips

- Laura Parker
- The Guardian,
- Thursday January 17 2008

"The name BitTorrent has become part of most people's day-to-day vernacular, synonymous with downloading every kind of content via the internet's peer-to-peer networks. But if a team of US researchers have their way, we may all be talking about CarTorrent in the not too distant future.....

Researchers from the University of California Los Angeles are working on a wireless communication network that will allow cars to talk to each other, simultaneously downloading information in the shape of road safety warnings, entertainment content and navigational tools...."

Problem Model – Monitored Function

- Need to define a problem which is more general than what was done so far (mostly, linear/monotonic functions, aggregates).
- But also a problem which is tractable and relevant to practical applications.
- A satisfactory choice is

$$f\left(\frac{x_1+\ldots+x_n}{n}\right)$$

f a general function,

x_i the data vectors at the nodes

Problem Model – Monitored Function

- Broad enough to cover many interesting problems, including maximum, top-*k*, variance, effective dimension…

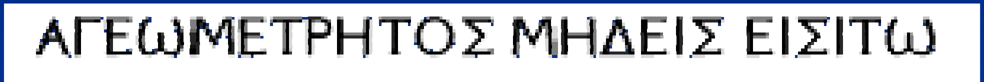
- Maximum: augment local vectors by their powers and use the fact that

$$\max\{x_i\} \cong \left(x_1^n + \ldots + x_k^n\right)^{\frac{1}{n}}$$

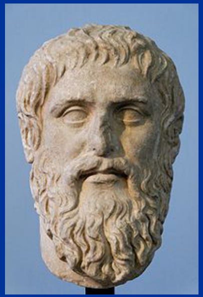
Other work

- Communication-efficient distributed monitoring of thresholded counts: R. Keralapura, G. Cormode, J. Ramamirtham, SIGMOD 2006.
- Algorithms for distributed functional monitoring: G. Cormode, S. Muthukrishnan, Ke Yi, SODA 2008.
- Handling Non-linear Polynomial Queries over Dynamic Data: S. Shah, K. Ramamritham, ICDE 2008.
- Communication-efficient online detection of network-wide anomalies: L. Huang, X.L. Nguyen, M. Garofalakis, J.M. Hellerstein, INFOCOM 2007.
- Efficient Detection of Distributed Constraint Violations: S. Agrawal, S. Deb, K.V.M. Naidu, R. Rastogi, ICDE 2007.
- Efficient Constraint Monitoring Using Adaptive Thresholds: S. Kashyap, J. Ramamirtham, R. Rastogi, P. Shukla, ICDE 2008.
- Approximate Decision Making in Large-Scale Distributed Systems: L. Huang, M. Garofalakis, A.D. Joseph, N. Taft, MLSys'2007.

New Approach – Based on Geometry


ΑΓΕωΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤω

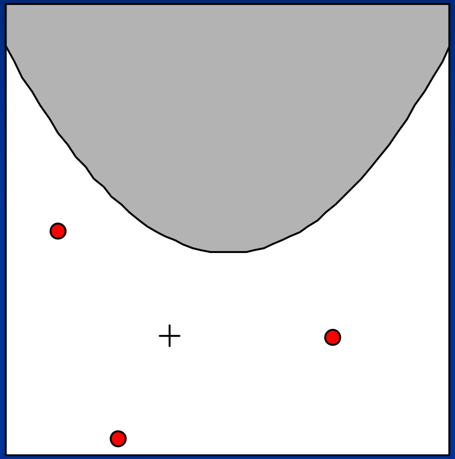
"Let no one ignorant of geometry enter!"



- We argue that for general functions, one must monitor the *domain* of the function and not its *range.*

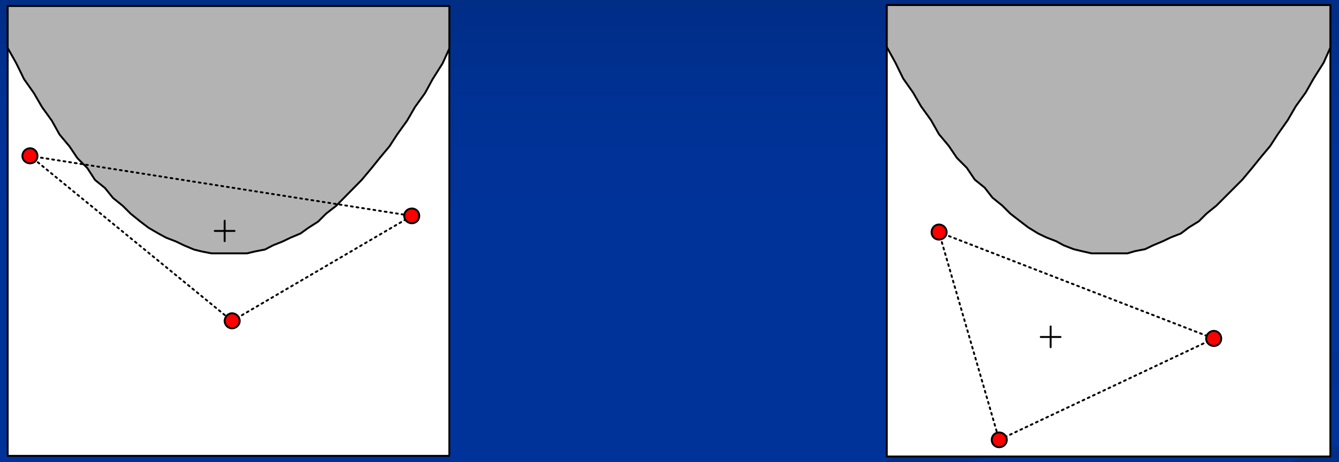
Geometric Approach

- Geometric Interpretation:
  - Each node holds a data vector
  - Coloring the data space
    - Grey: function > threshold
    - White: function < threshold



- Goal: determine *color* of global data vector (average).
- General function – not necessarily linear, monotonic, convex… implies a general subset of Euclidean space in which function > threshold.

Bounding the Convex Hull

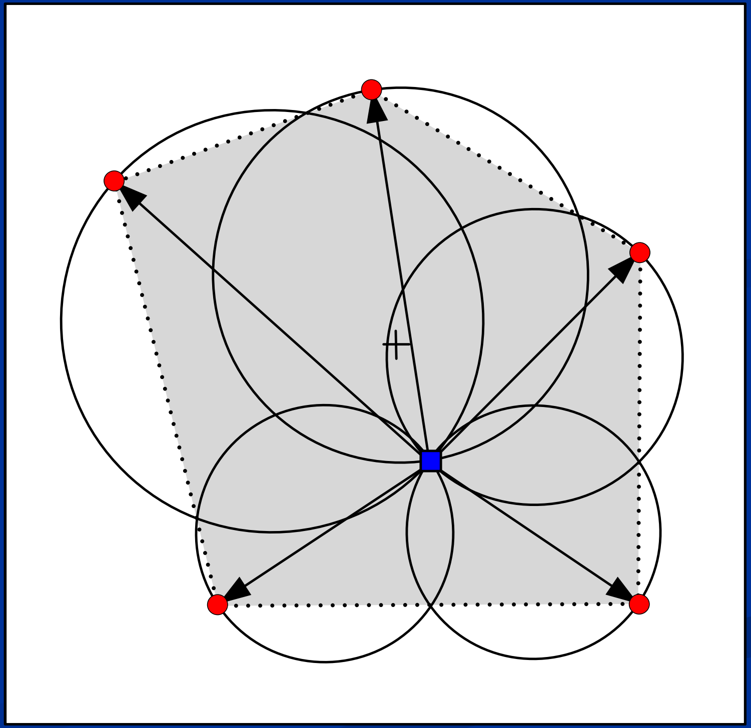


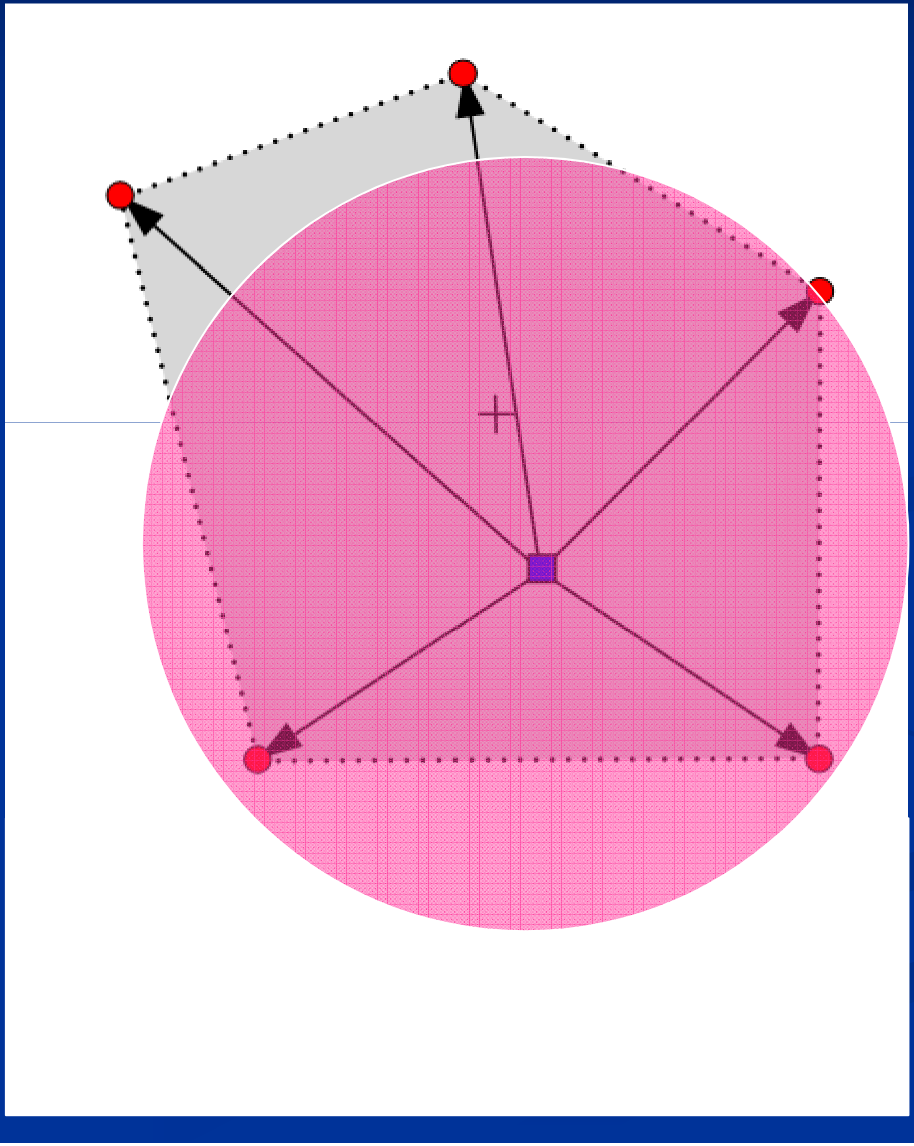
- Observation: average is in the convex hull
- If convex hull is monochromatic, we know what happens at the average vector
- Problem - convex hull may become large

DataMovie.avi

The Bounding Theorem

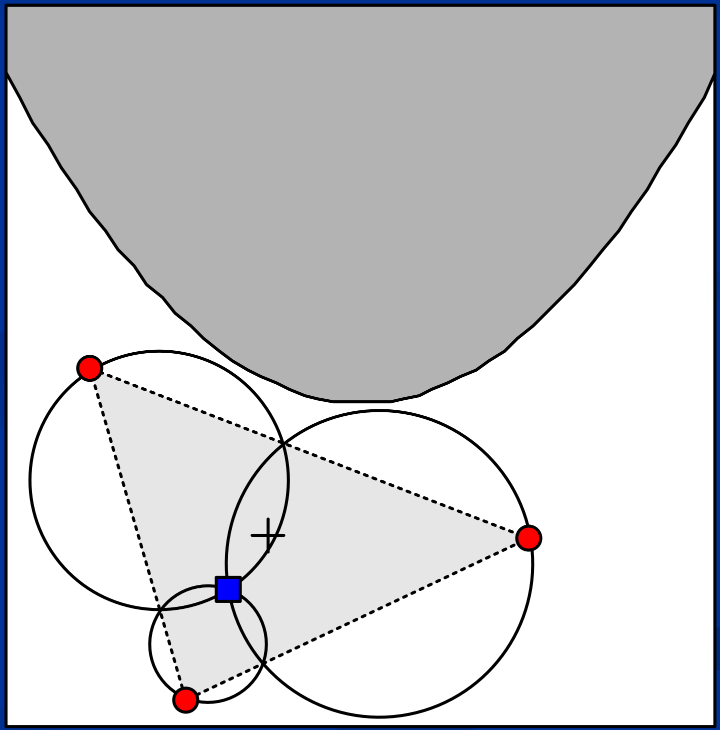
- A reference point is known to all nodes
- Each vertex constructs a sphere
- Theorem: convex hull is bounded by the union of spheres
- → Local constraints!





Basic Algorithm

- An initial estimate vector is calculated
- Nodes check color of drift spheres
    - Drift vector is the diameter of the drift sphere
- If any sphere is non-monochromatic: node triggers re-calculation of estimate vector

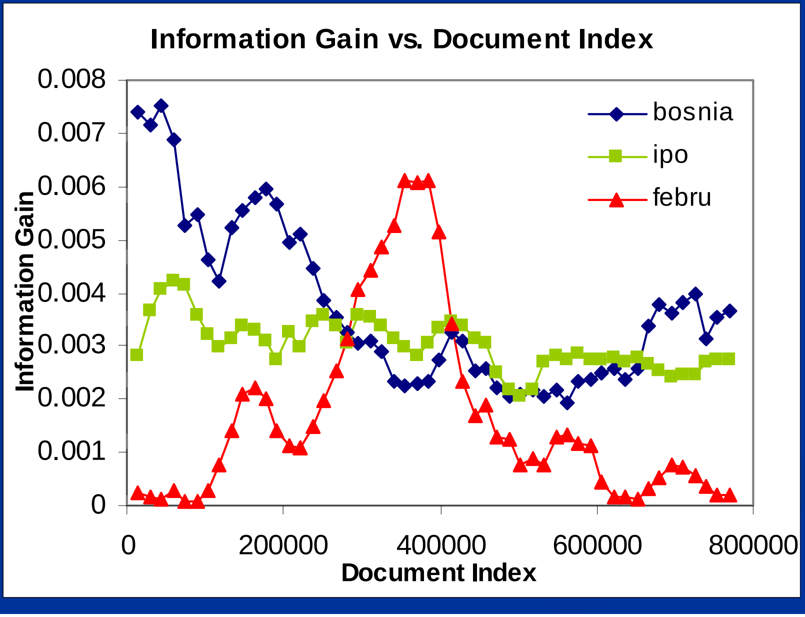


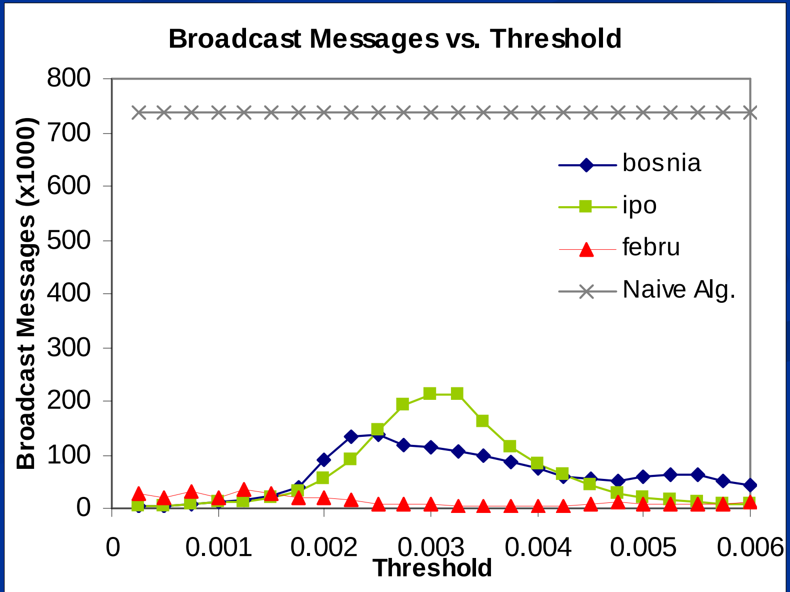
DataConvMovie.avi

DriftMovie.avi

Experiments: Reuters Corpus (RCV1-v2)

- 800,000+ news stories
- Aug 20 1996 -- Aug 19 1997
- Corporate/Industrial tagging simulates spam
- Information Gain $IG(C) = \sum_{i\in\{1,2\}} \sum_{j\in\{1,2\}} c_{i,j} \log\left(\frac{c_{i,j}}{(c_{i,1}+c_{i,2})(c_{1,j}+c_{2,j})}\right)$

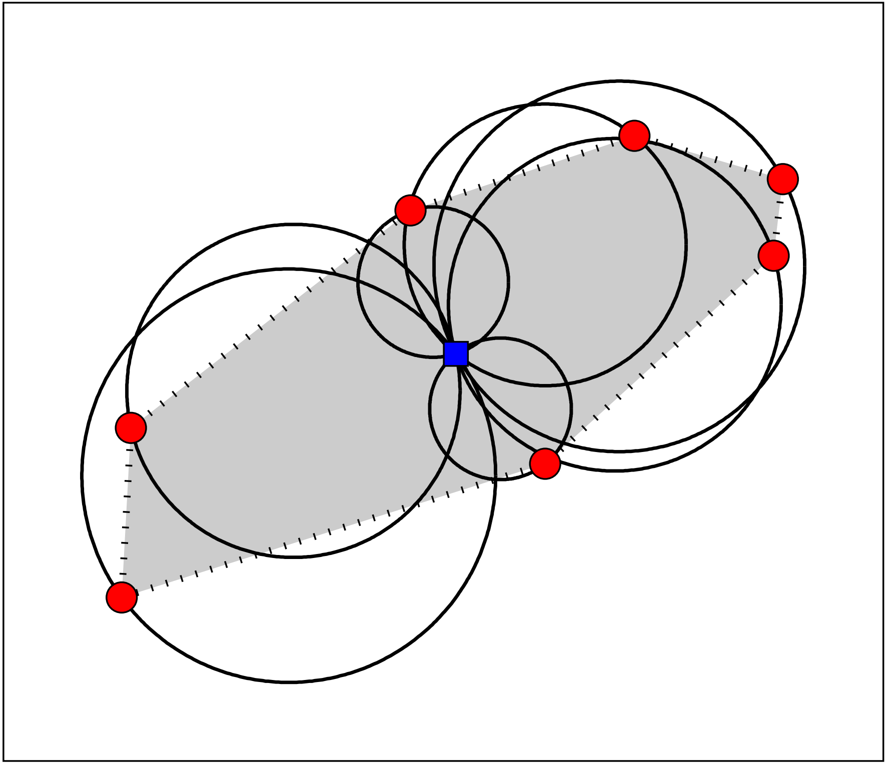


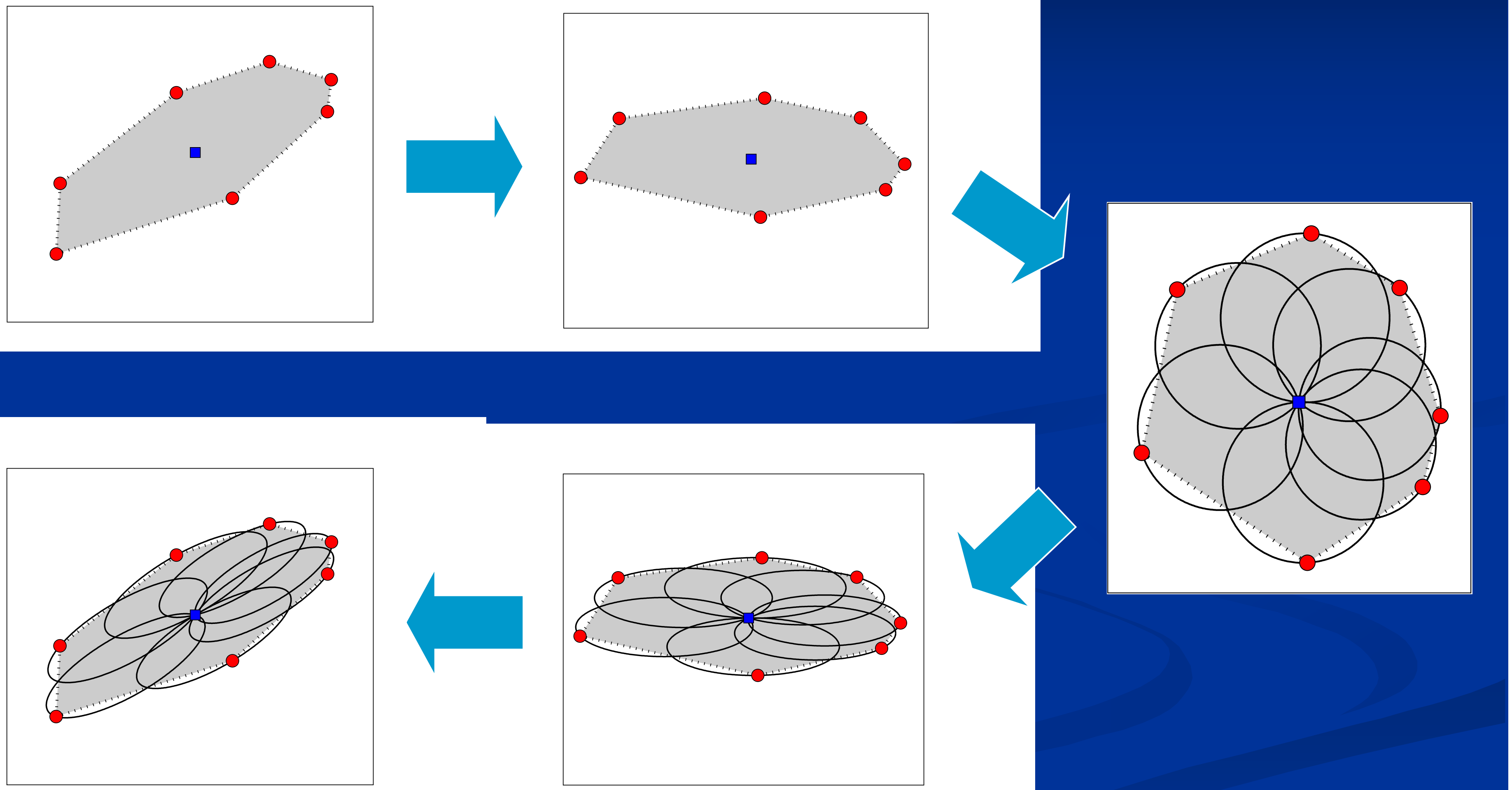


Issues

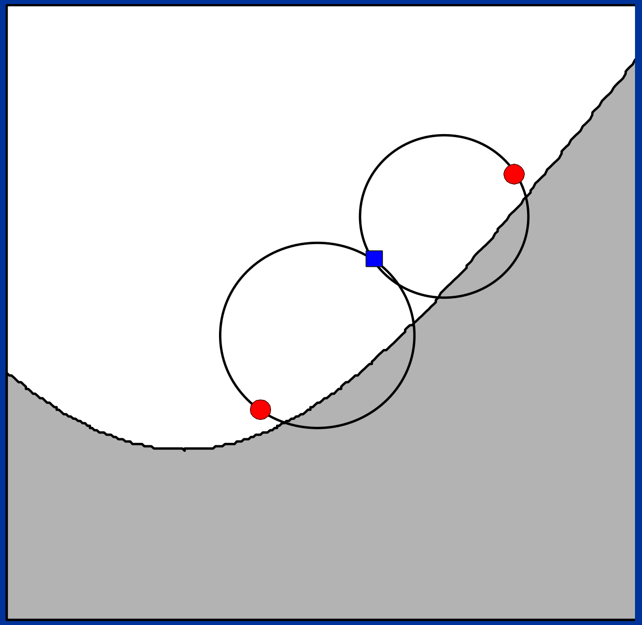
- A general solution, but it lacks…
  - Rigorous optimality measure
  - Adapt to specific instance problems
  - Reference point determination
  - “A theory!!!”
- Monochromaticity checking

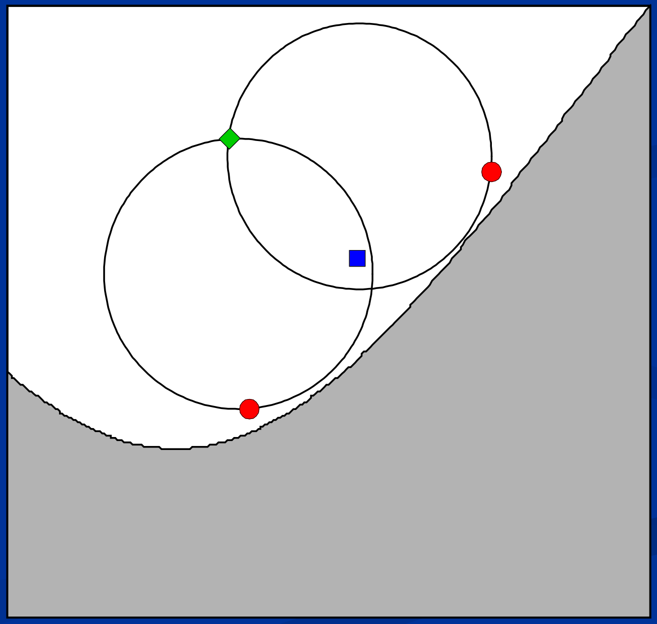
Improvement: fit bounding volumes to data distribution



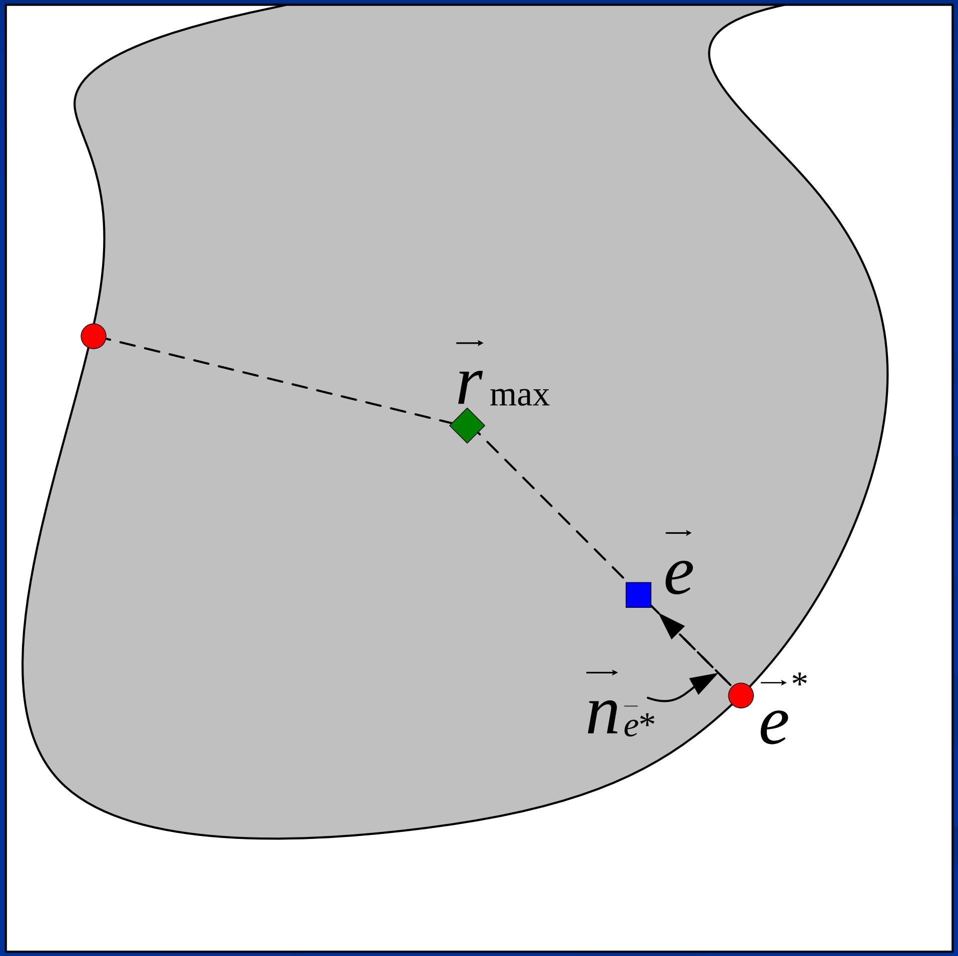


Improvement: larger (but better) bounding volumes via "inner" reference vectors

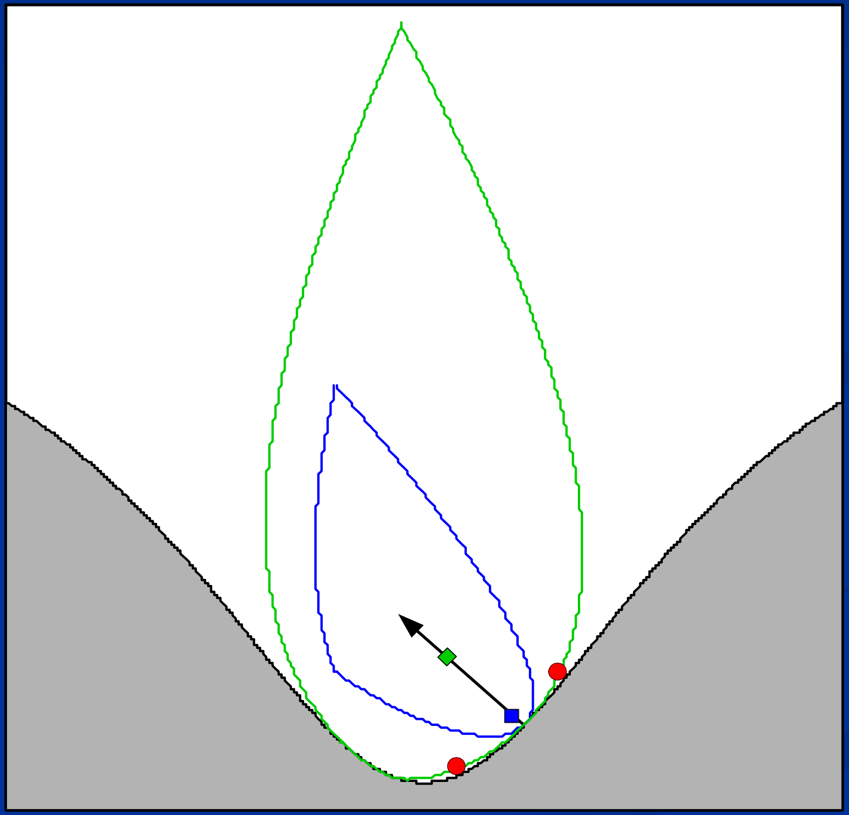




Determining the new Reference Vector



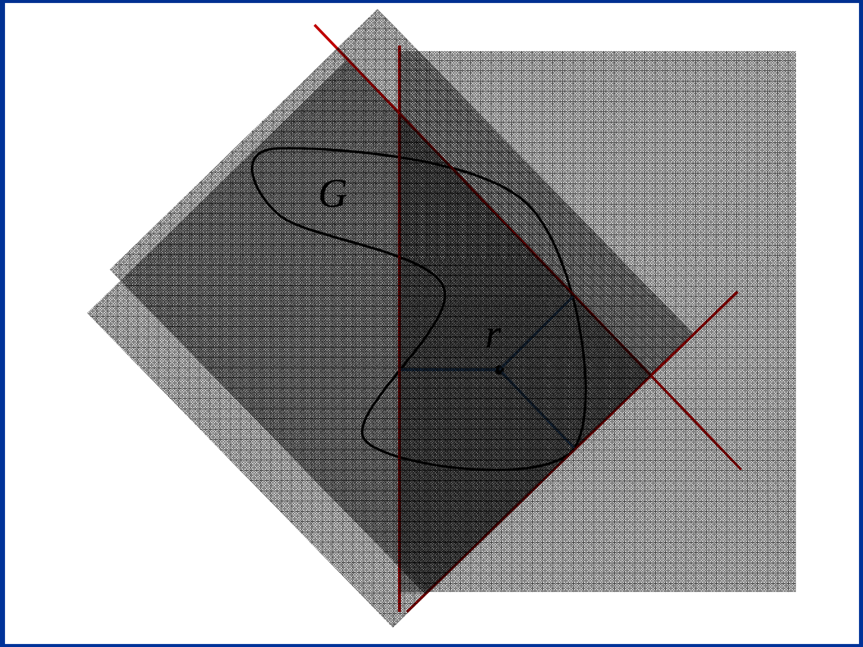
Inner reference vector

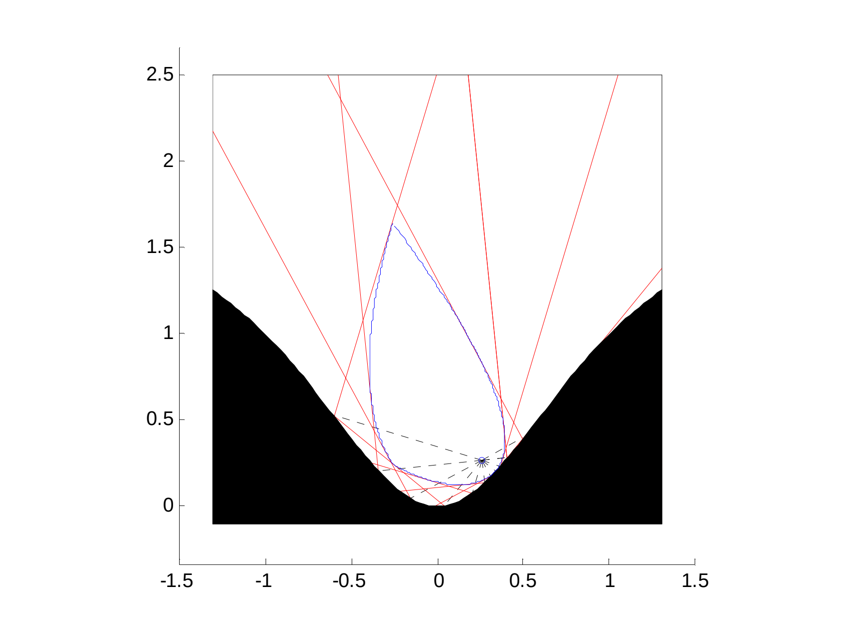


Convexity

- The spherical constraints define regions in which the local vectors can roam freely (no alerts required).
- These regions turn out to be *convex*.

Proof: for two points, the region is a half-plane. Therefore, any region is the intersection of half-planes, which are convex.





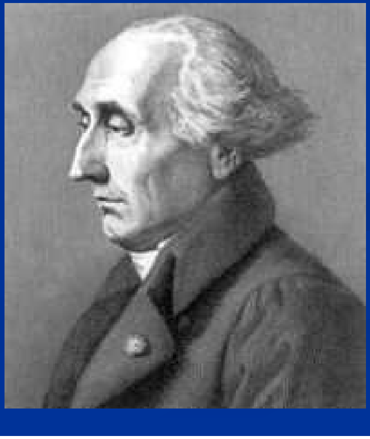
And vice-versa – it is easy to see that any convex subset will do the job (since it is closed under averages).

So – why not look for an *optimal* convex region?

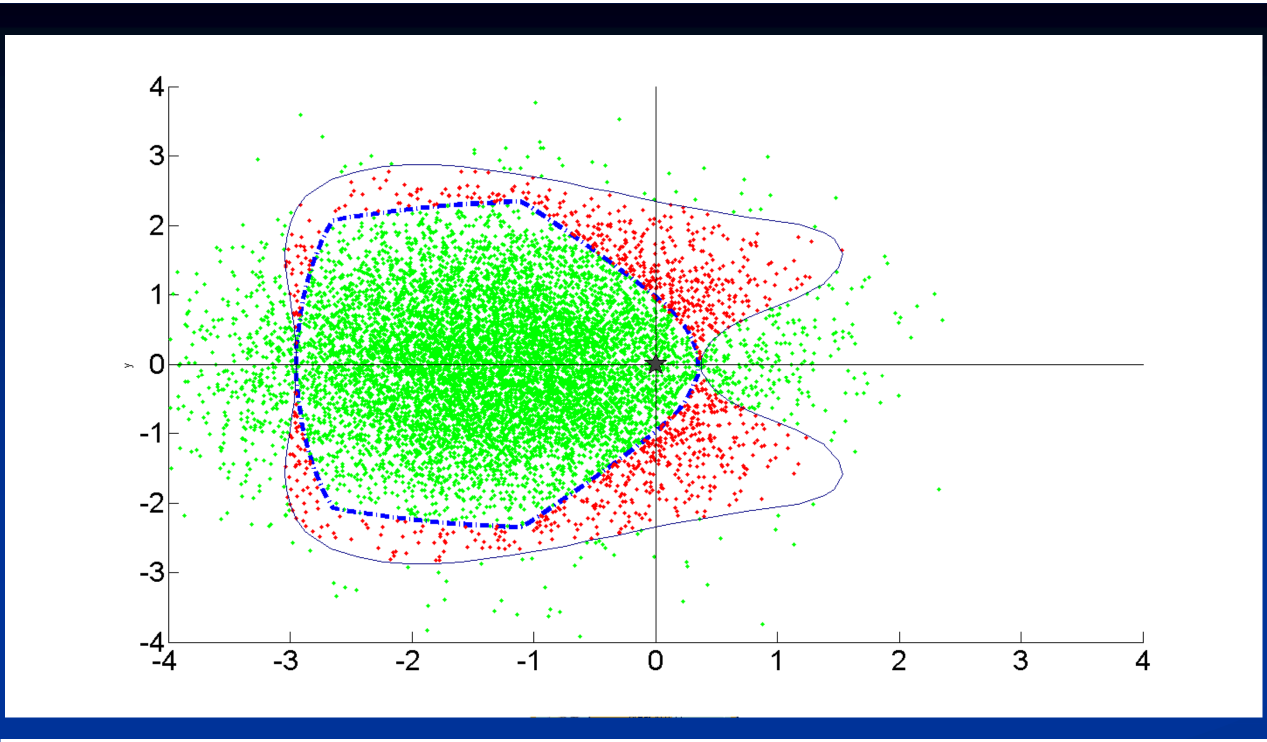
- Very difficult (infinite dimensional, non-linear) optimization problem: find a maximal (in the probabilistic sense) convex subset.

- A more general approach, requiring additional machinery (optimization, probability, algorithms).

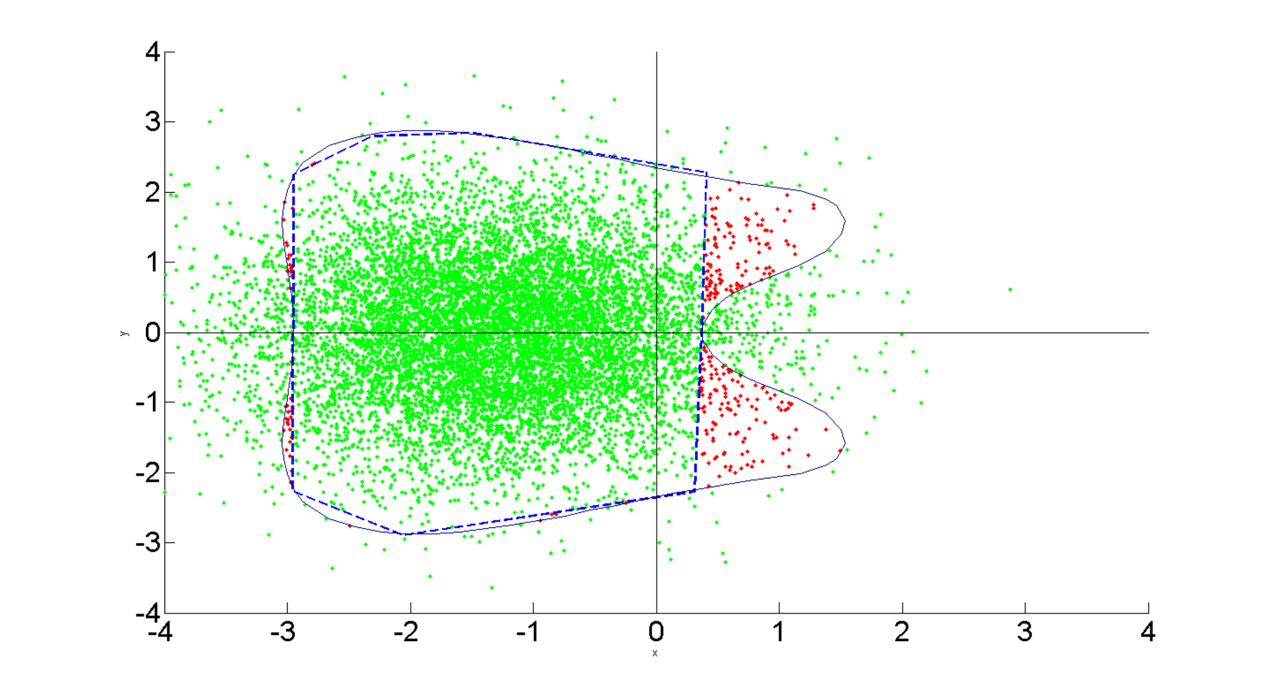
"As long as algebra and geometry have been separated, their progress have been slow and their uses limited, but when these two sciences have been united, they have lent each mutual forces, and have marched together towards perfection."



Spherical constraints



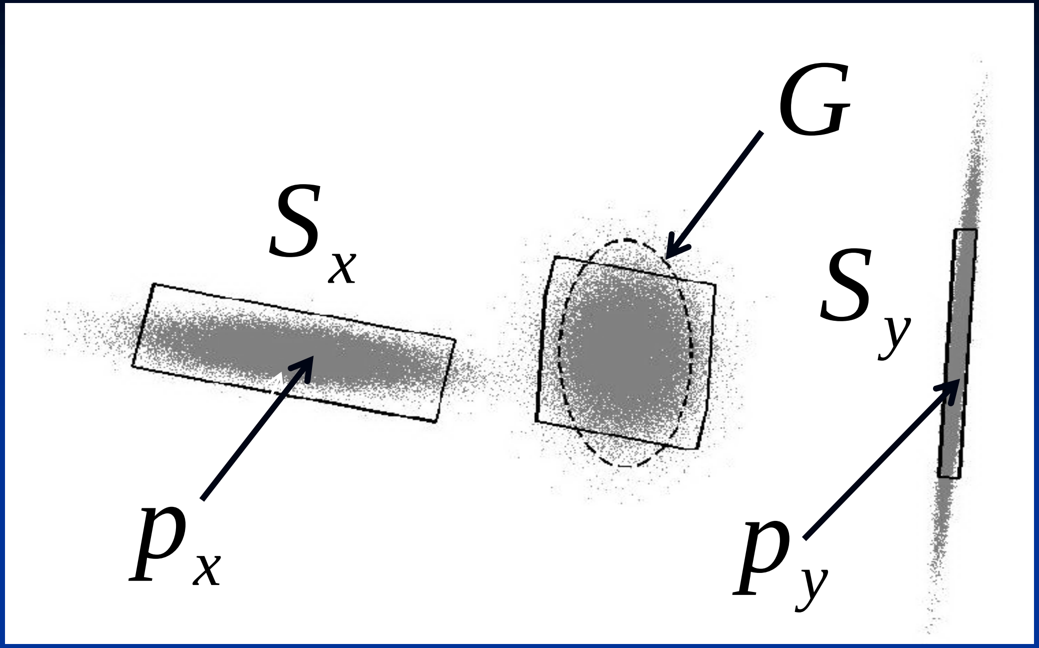
Optimal convex octagon

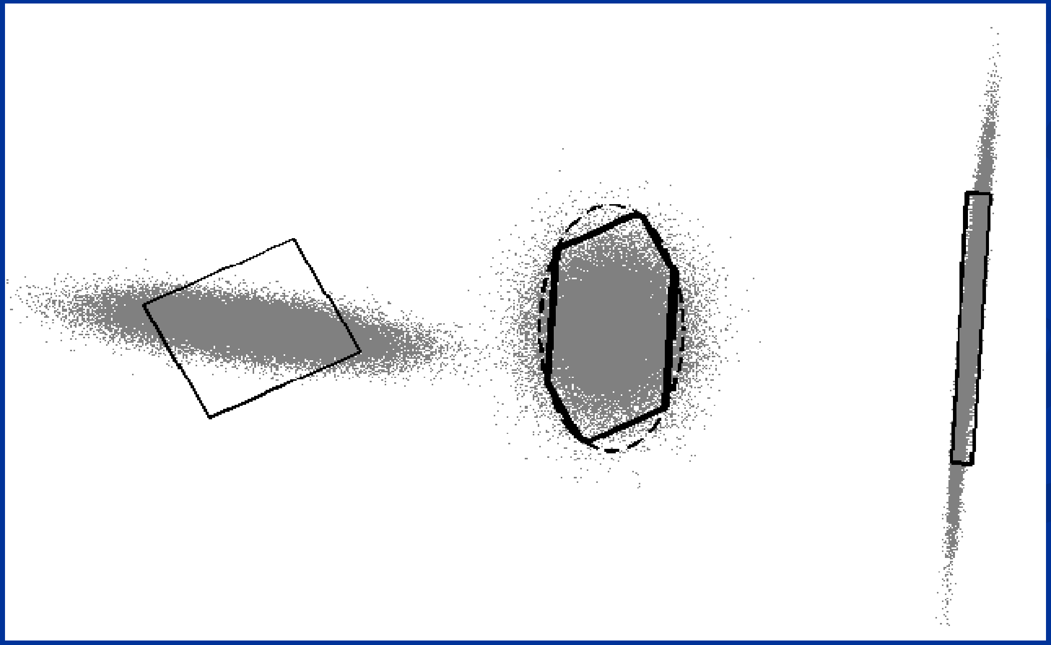


- So far, we assumed all nodes have the same distribution, and therefore they are all assigned the same region!
- General solution – assign each node its own region, such that:

i. Each node is assigned a region which fits its data distribution.

ii. The average of vectors from the distinct regions satisfies the threshold constraint (it lies inside the set of points at which the function is smaller than the threshold).


G
S_x
S_y
p_x
p_y



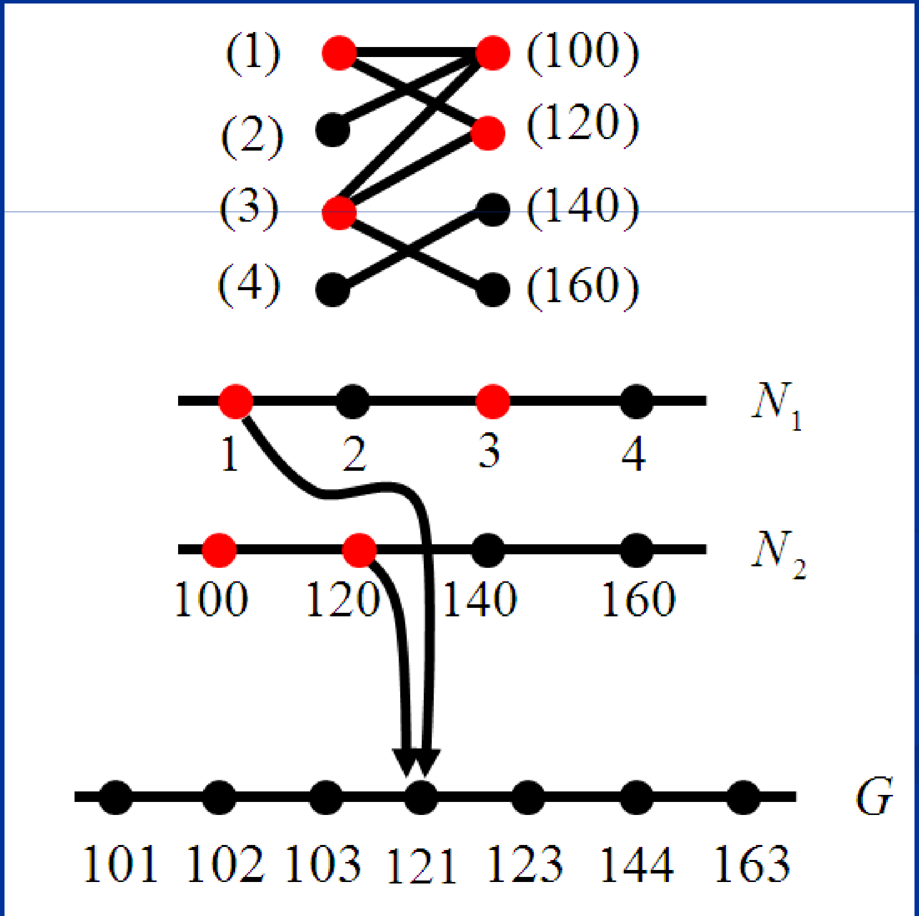
Minkowski sum:

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

Optimization problem:

Maximize $\int_{S_x} p_x dx \int_{S_y} p_y dy$ subject to $\frac{S_x \oplus S_y}{2} \subset G$

- This problem is NP-hard even for the simplest case: two nodes, one-dimensional data (reducible to maximal biclique):

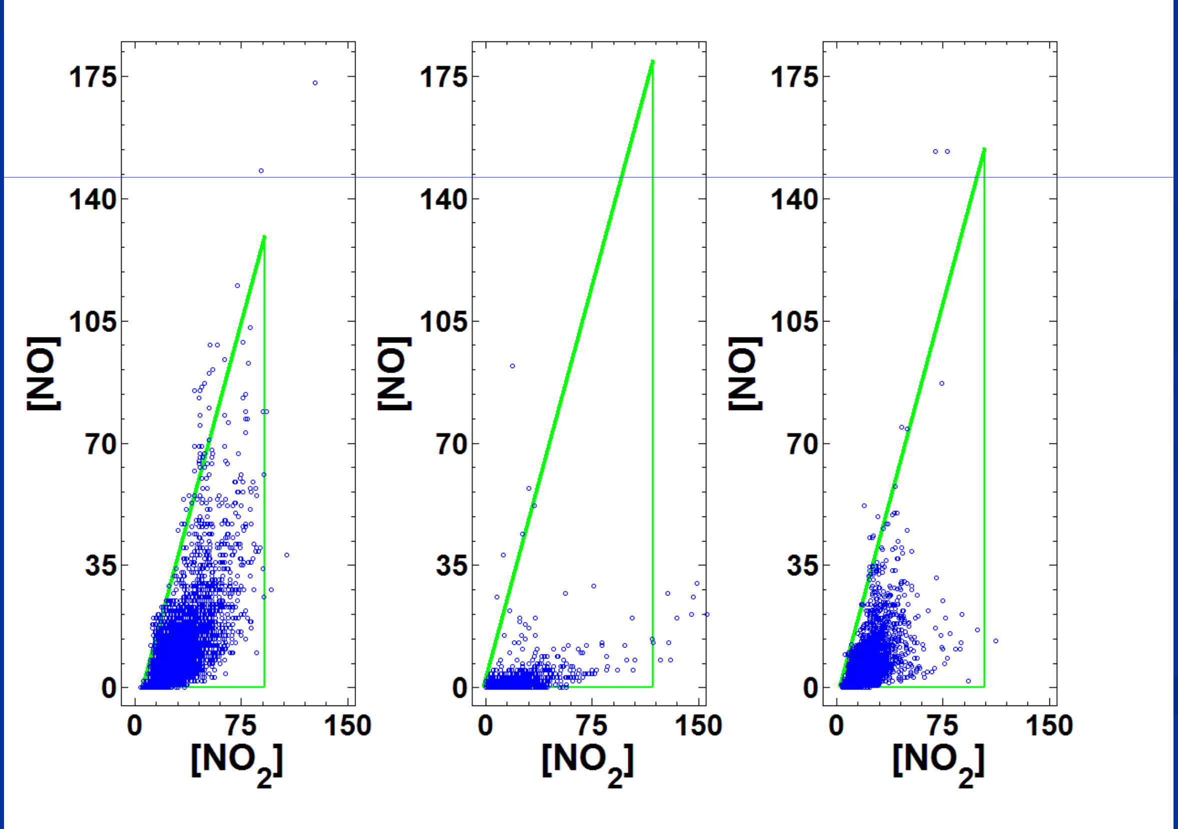


Making it work:

- Hierarchical clustering of nodes.

- Various computational tricks to quickly test the constraints and compute the target function.

❑ Violation recovery – find optimal pairs of nodes which “balance” each others


[NO]
$[NO_2]$