Subscription Covering Detection in Publish/Subscribe Systems

A Random Projection Approach

Duc A. Tran

Computer Science
University of Massachusetts, Boston

Outline

Publish/Subscribe Systems

- A network system of
  - Subscribers: the consumers of the data
    - Subscription: a query submitted by the subscriber to the network to specify a data interest. E.g., “I am very interested in “blah, blah, blah", whenever somebody has it, please send it to me"
  - Publishers: the producers of the data
    - Event: the data information submitted by the publisher to the network to find consumers. E.g., “I have this “blah, blah, blah". Please let those persons interested in it know that I have their data"
- Enable subscribers and publishers, which do not know each other, to find each other quickly

Applications: social networking, e-commerce, event monitoring, anomaly detection, etc.

Pub/Sub vs. Traditional Search

- Traditional search = Request/Reponse
  - Data must exist already
  - Queries are submitted with expectation to receive matching data immediately
  - Data is stored in the network in advance, not queries
    - Data indexing and storage = an important problem
- Pub/sub:
  - Data may currently exist, currently not exist, or never exist
  - Queries are sent in advance with expectation to be notified whenever matching data will be found
  - Queries must be stored in the network in advance
    - Subscription indexing and storage = an important problem

Pub/Sub vs. Traditional Search

- Traditional search = Request/Reponse
  - Data must exist already
  - Queries are submitted with expectation to receive matching data immediately
  - Data is stored in the network in advance, not queries
    - Data indexing and storage = an important problem
- Pub/sub:
  - Data may currently exist, currently not exist, or never exist
  - Queries are sent in advance with expectation to be notified whenever matching data will be found
  - Queries must be stored in the network in advance
    - Subscription indexing and storage = an important problem

Assumptions

Event and Subscription

- Event: $x = (x_1, x_2, ..., x_d) \in R^d$
  - d: number of attributes associated with each event
- Subscription:
  - Rectangular: $S = [min_1, max_1] \times [min_2, max_2] \times ...[min_d, max_d]$
    - x matches S iff $min_i \leq x_i \leq max_i$
  - Spherical: $S = (s, r)$ (a sphere centered at $s \in R^d$ with radius $r \in R^+$
    - x matches S iff $\| x - s \| \leq r$
  - Subscription S_1 covers subscription S_2 iff $S_1 \supseteq S_2$

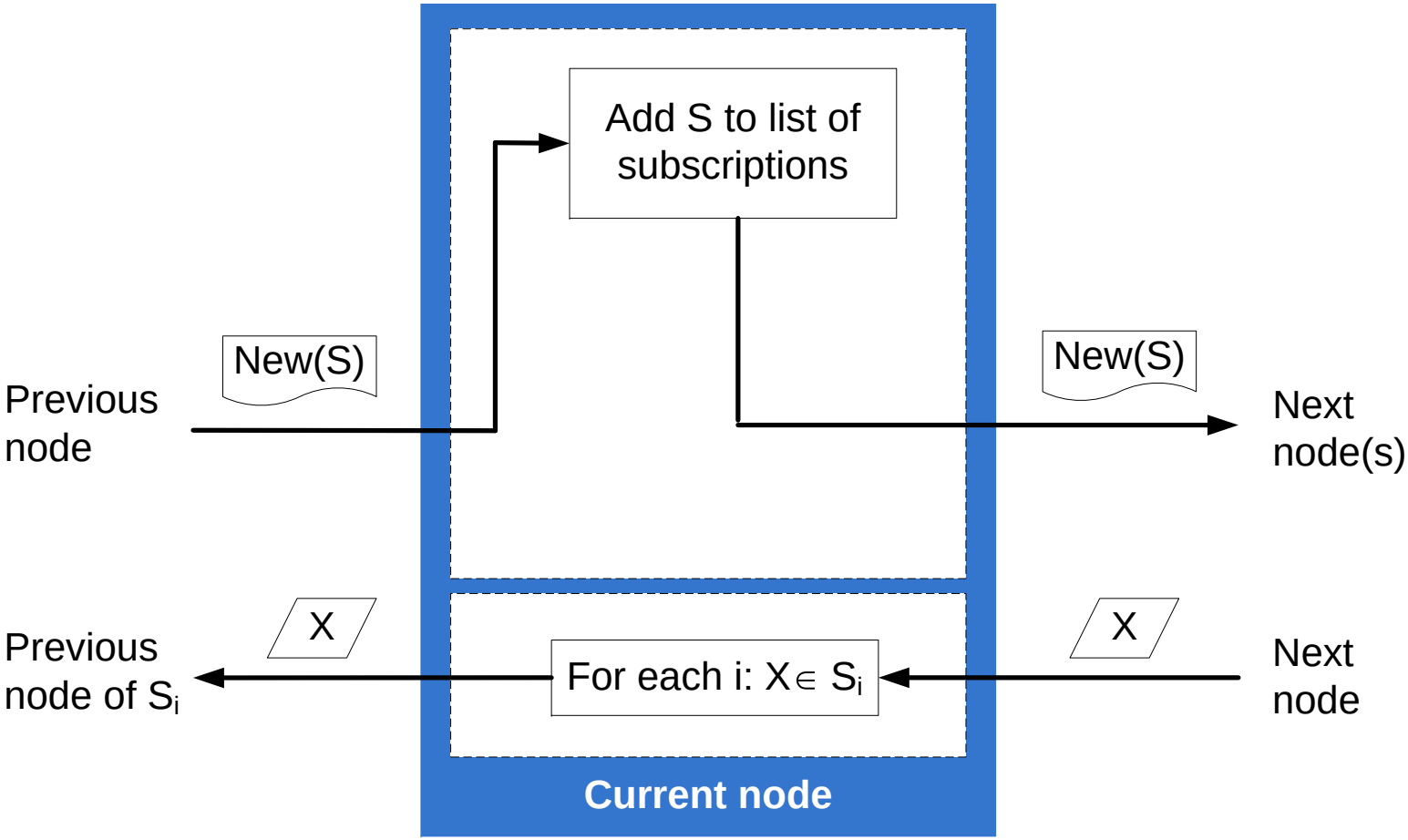
E.g. a pub/sub weather forecast network

- Subscription = (location $=$ *, temp $\geq 100^o$, humidity $\geq 50\%$)
- Event 1 = (Boston, 50^o, 60%) not matching above subscription
- Event 2 = (Phoenix, 110^o, 80%) matching above subscription

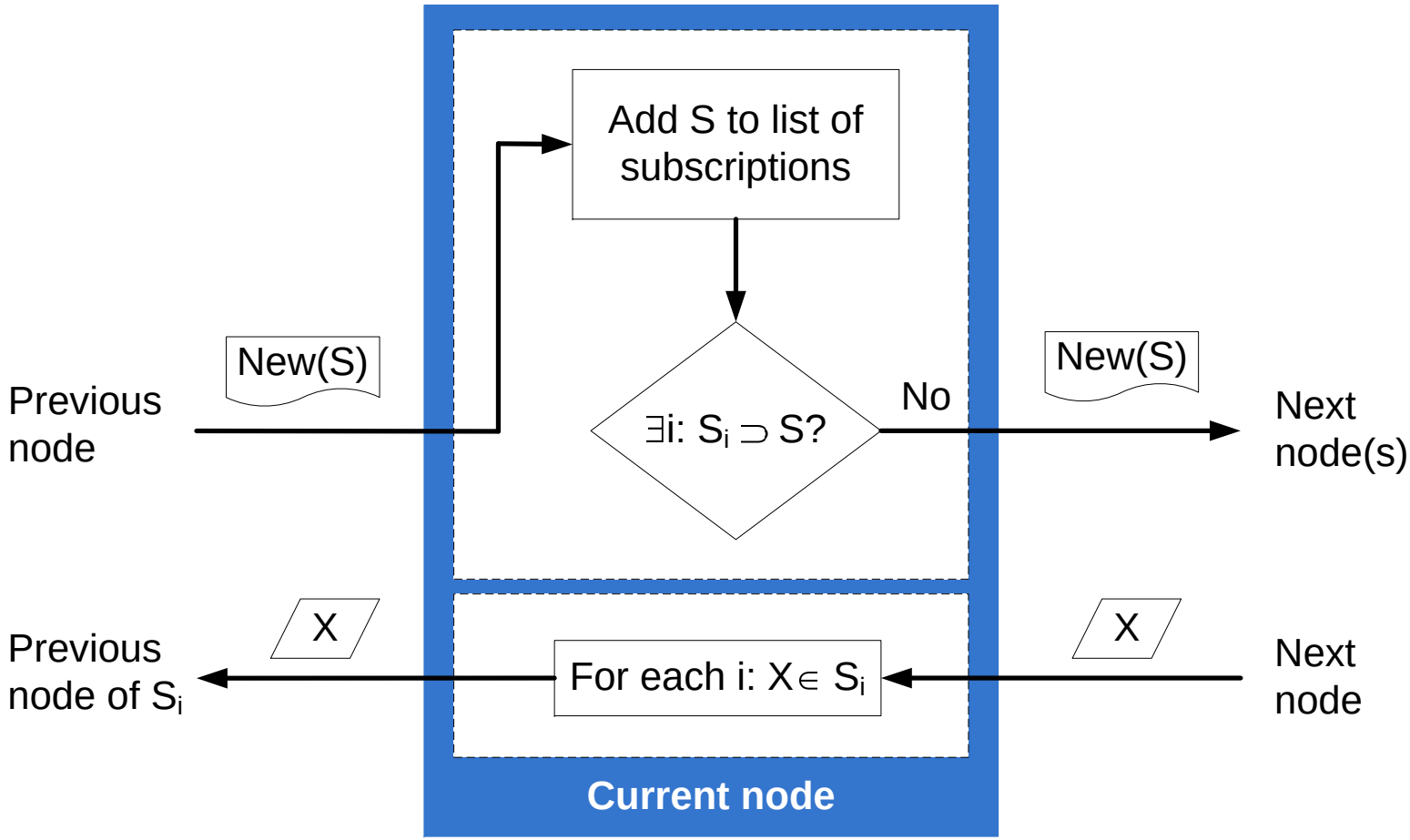
Pub/Sub Issues

- Communication
  - Need mechanisms to disseminate subscriptions and events in the network
- Storage Structure
  - Need efficient data structures to store subscriptions and events
  - Need fast and efficient algorithms to match subscriptions with events

Subscription/Event Routing

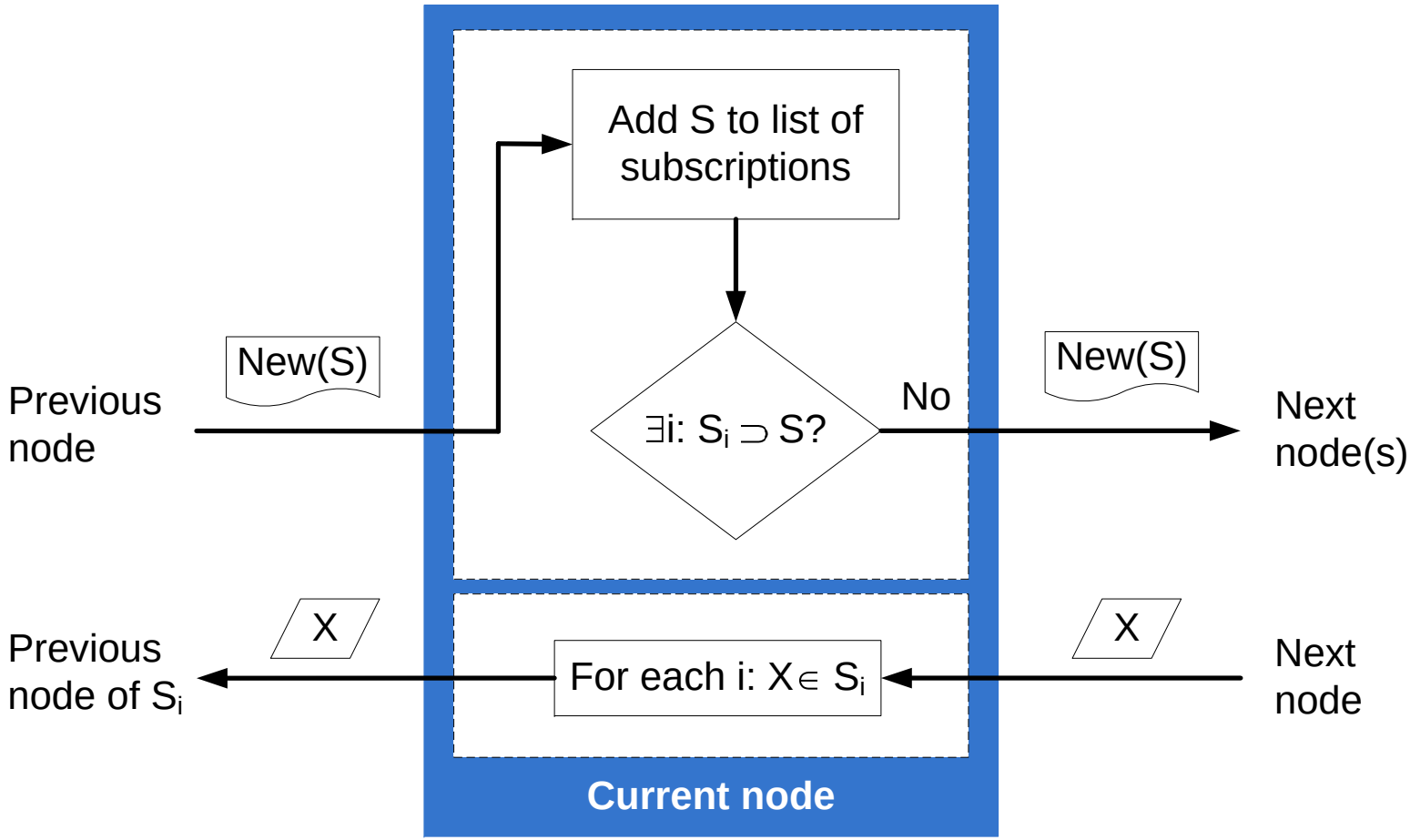


Sub/Event Routing: Use Subscription Covering



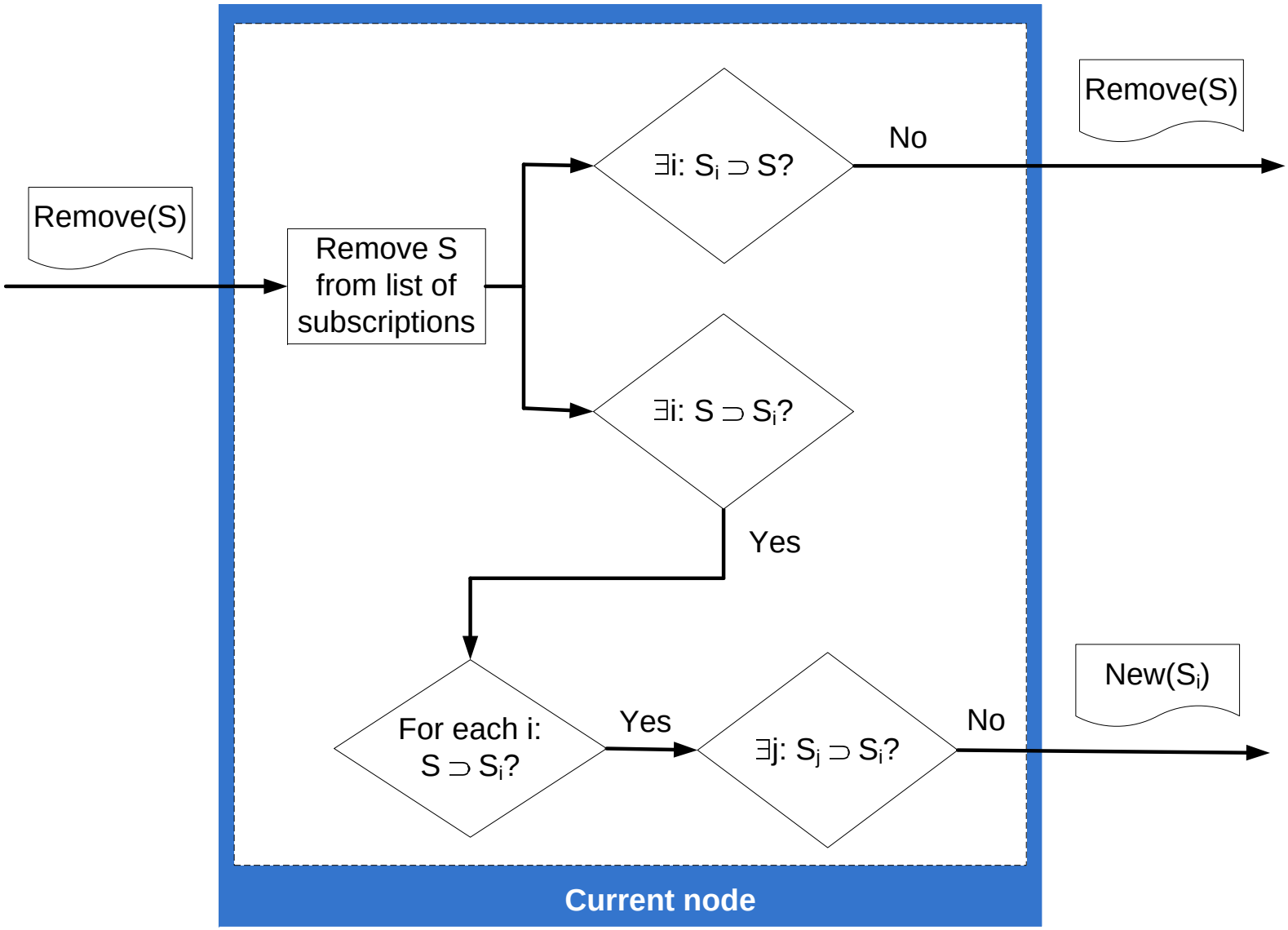
- Reduce traffic of subscription advertising and event notification
- Reduce the number of subscriptions stored at each node

Sub/Event Routing: Use Subscription Covering



- Reduce traffic of subscription advertising and event notification
- Reduce the number of subscriptions stored at each node

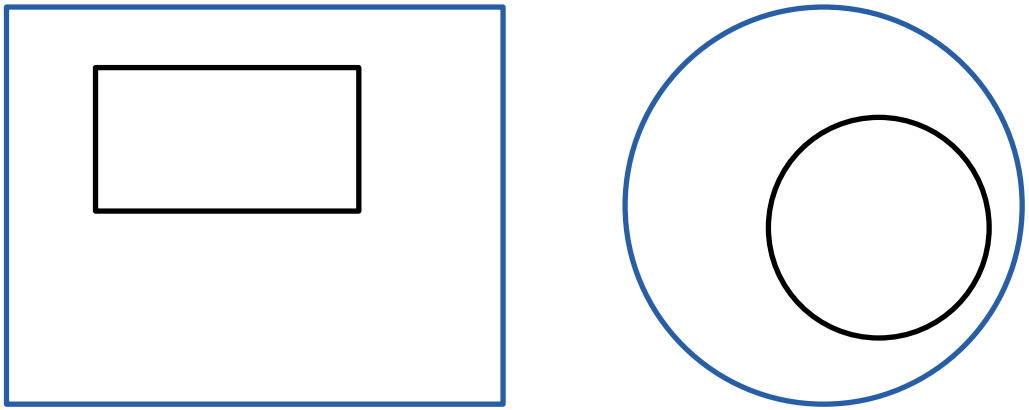
Subscription Covering: Use in Un-Subscription



Subscription Covering: Good but S..l..o..w

Checking subscription covering relationship among a large set of subscriptions (n) in high dimension (d) is very slow

- Curse of dimensionality!



- Given a subscription, time to find its coverings
  - $O(\log^{2d-1} n)$ (rectangular subscriptions)
  - $O(nd)$ (spherical subscriptions)

The Subscription Covering Problem

Given a new subscription, if we mistakenly conclude that it is not covered by any existing subscription, this wrong decision does not harm the correctness of the system. What it does is to forward the subscription, resulting in just some unnecessary costs.

Problem

Finding the perfect solution is very slow. Instead, can we find an approximate solution that may produce false coverings but is much faster with a high approximation accuracy?

The Subscription Covering Problem

Given a new subscription, if we mistakenly conclude that it is not covered by any existing subscription, this wrong decision does not harm the correctness of the system. What it does is to forward the subscription, resulting in just some unnecessary costs.

Problem

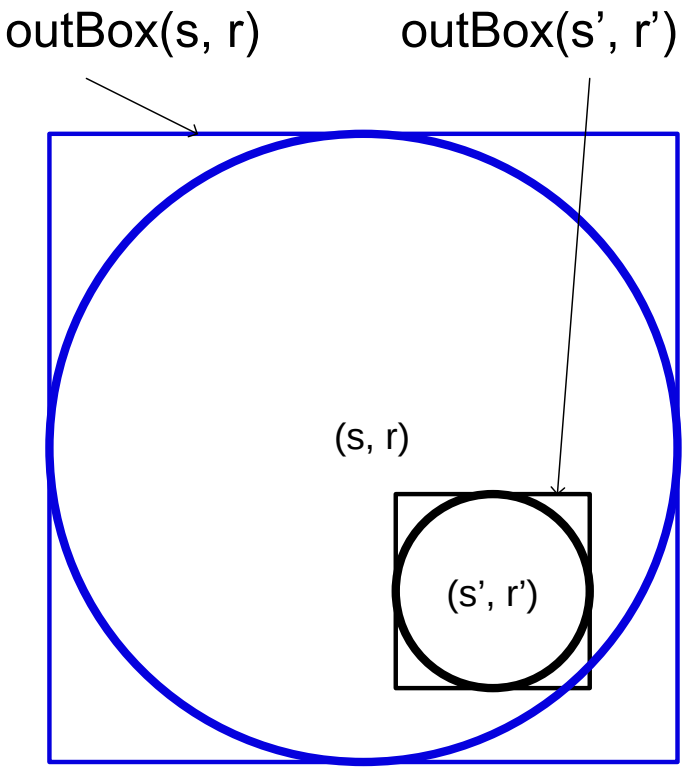
Finding the perfect solution is very slow. Instead, can we find an approximate solution that may produce false coverings but is much faster with a high approximation accuracy?

Our Result

- For spherical subscriptions,
  - Search time is $O(\log^{2k-1} n)$, where $k \ll d$
    - Much better than $O(nd)$
  - Accuracy exponentially approaches 100% as k increases
- For rectangular subscriptions,
  - will be discussed later...

Approximating Spheres with Circumscribed Boxes

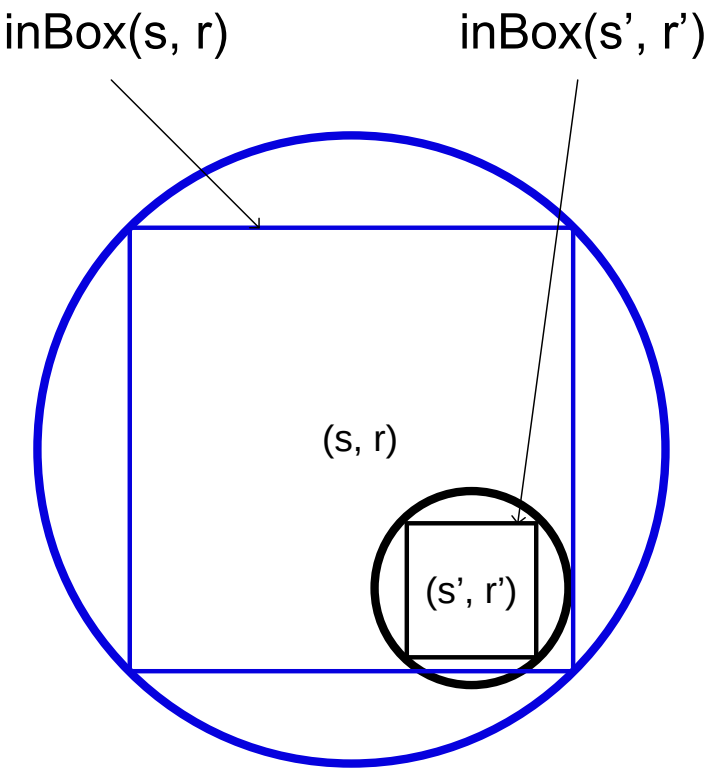
It is true that $(s, r) \supseteq (s', r') \Rightarrow outBox(s, r) \supseteq outBox(s', r')$



Condition 1:

$outBox(s, r) \not\supseteq outBox(s', r') \Rightarrow (s, r) \not\supseteq (s', r')$

Approximating Spheres with Inscribed Boxes

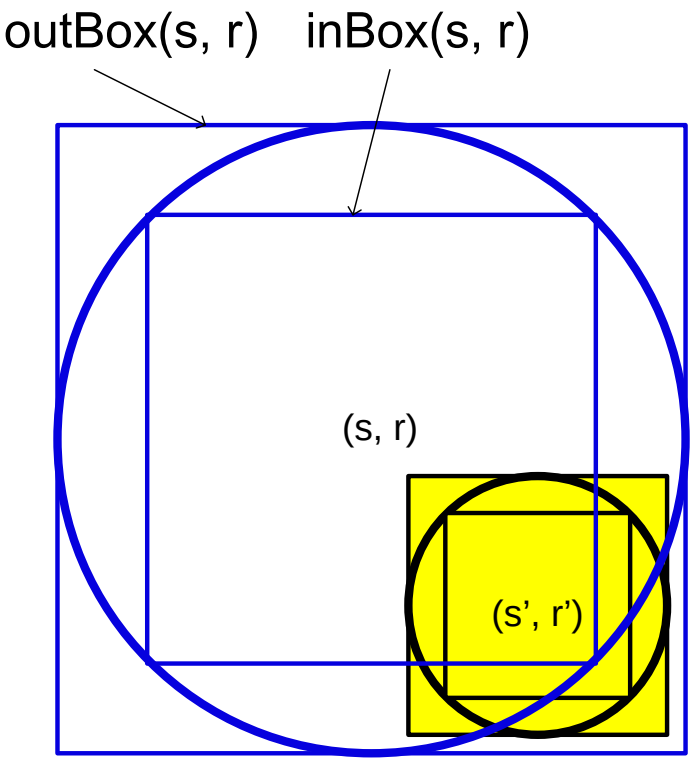


Condition 2:

$inBox(s, r) \supseteq inBox(s', r') \Rightarrow (s, r) \supseteq (s', r')$

Approximating Spheres with Boxes: Ambiguity

$outBox(s, r) \supseteq outBox(s', r')$ and $inBox(s, r) \not\supseteq inBox(s', r') \Rightarrow$ no conclusion possible



Assuming uniform distribution for the subscriptions

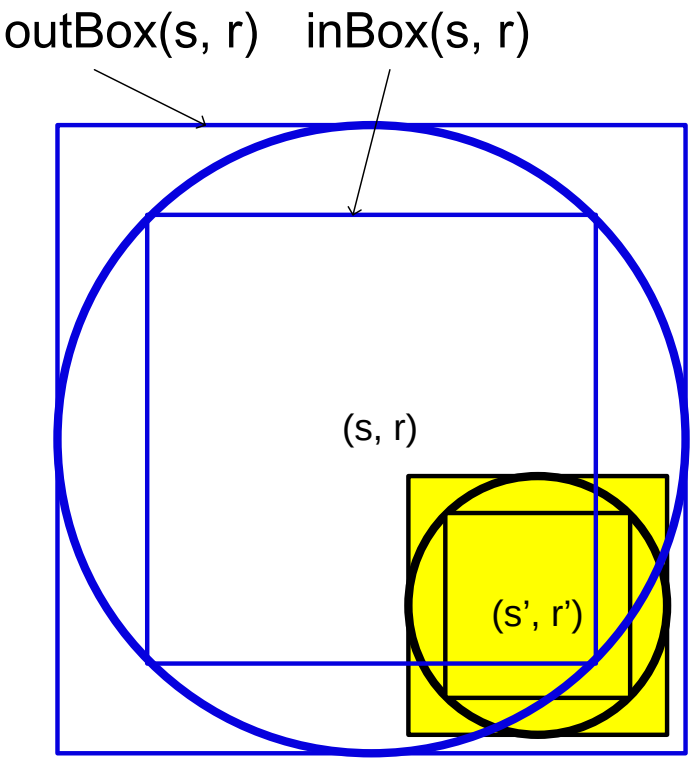
$$\begin{aligned} & Pr\{ambiguous\ case\} \\ = & \ 1 - \left(\frac{volume(inBox(s,r))}{volume(outBox(s,r))} \right)^d \\ = & \ 1 - \left(\frac{1}{\sqrt{2}} \right)^d \end{aligned}$$

$\rightarrow$ highly likely if d is a large number.

Can we do it better?

Approximating Spheres with Boxes: Ambiguity

$outBox(s, r) \supseteq outBox(s', r')$ and $inBox(s, r) \not\supseteq inBox(s', r') \Rightarrow$ no conclusion possible



Assuming uniform distribution for the subscriptions

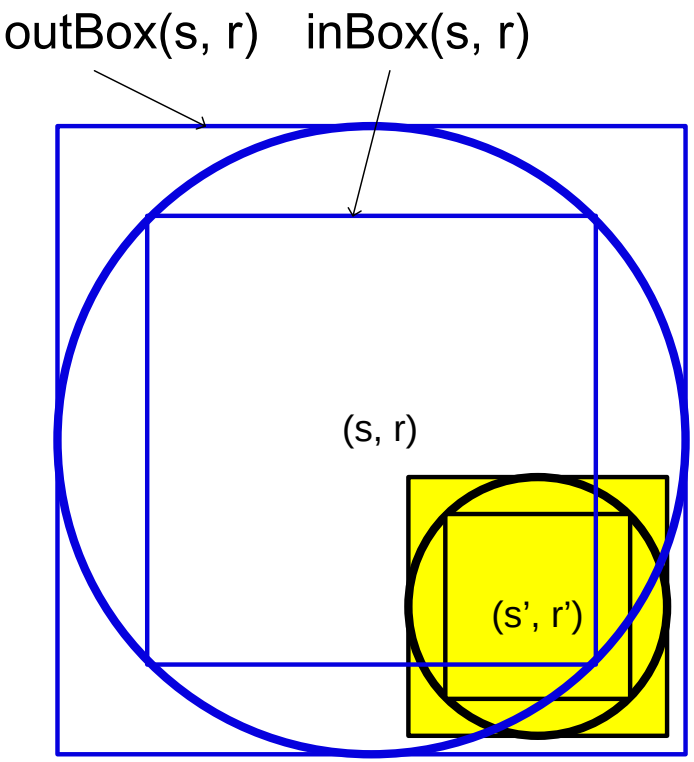
$$\begin{aligned} & Pr\{ambiguous\ case\} \\ = & \ 1 - \left(\frac{volume(inBox(s,r))}{volume(outBox(s,r))} \right)^d \\ = & \ 1 - \left(\frac{1}{\sqrt{2}} \right)^d \end{aligned}$$

$\rightarrow$ highly likely if d is a large number.

Can we do it better?

Approximating Spheres with Boxes: Ambiguity

$outBox(s, r) \supseteq outBox(s', r')$ and $inBox(s, r) \not\supseteq inBox(s', r') \Rightarrow$ no conclusion possible



Assuming uniform distribution for the subscriptions

$$
\begin{aligned}
& Pr\{ambiguous\ case\} \\
= \quad & 1 - \left(\frac{volume(inBox(s,r))}{volume(outBox(s,r))} \right)^d \\
= \quad & 1 - \left(\frac{1}{\sqrt{2}} \right)^d
\end{aligned}
$$

$\rightarrow$ highly likely if d is a large number.

Can we do it better?

Random Projection: Motivation

- Approximating a sphere with its axis-parallel inscribed and circumscribed boxes does not work
- What if we approximate a sphere with a random-dimension box?

Johnson-Lindenstrauss Lemma (1992)

Given $\varepsilon \in \mathbb{R}^+$ and $n \in \mathbb{N}^+$, let $k \in \mathbb{N}^+$ such that $k \geq k_0 = O\left(\frac{\log n}{\varepsilon^2}\right)$. For every set $P \subset \mathbb{R}^d$ of n points, there exists $f : \mathbb{R}^d \rightarrowtail \mathbb{R}^k$ such that for all $x, y \in P$

$$(1-\varepsilon) \parallel x-y \parallel^2 \leq \parallel f(x)-f(y) \parallel^2 \leq (1+\varepsilon) \parallel x-y \parallel^2$$

Random Projection: Motivation

- Approximating a sphere with its axis-parallel inscribed and circumscribed boxes does not work
- What if we approximate a sphere with a random-dimension box?

Johnson-Lindenstrauss Lemma (1992)

Given $\varepsilon \in \mathbb{R}^+$ and $n \in \mathbb{N}^+$, let $k \in \mathbb{N}^+$ such that $k \geq k_0 = O\left(\frac{\log n}{\varepsilon^2}\right)$. For every set $P \subset \mathbb{R}^d$ of n points, there exists $f : \mathbb{R}^d \rightarrowtail \mathbb{R}^k$ such that for all $x, y \in P$

$$(1-\varepsilon) \parallel x-y \parallel^2 \leq \parallel f(x)-f(y) \parallel^2 \leq (1+\varepsilon) \parallel x-y \parallel^2$$

J-L Lemma: Choosing Mapping f ([Achlioptas03])

Let

$$k_0 = \frac{4+2\beta}{\varepsilon^2/2 - \varepsilon^3/3} \log n$$

Let $U = \{u_{ij}\}_{d \times k}$ generated from either one of the following two probability distributions

$$u_{ij} = \begin{cases} +1 & \text{with prob } 1/2 \\ -1 & \text{with prob } 1/2 \end{cases}$$

$$u_{ij} = \sqrt{3} \times \begin{cases} +1 & \text{with prob } 1/6 \\ 0 & \text{with prob } 2/3 \\ -1 & \text{with prob } 1/6 \end{cases}$$

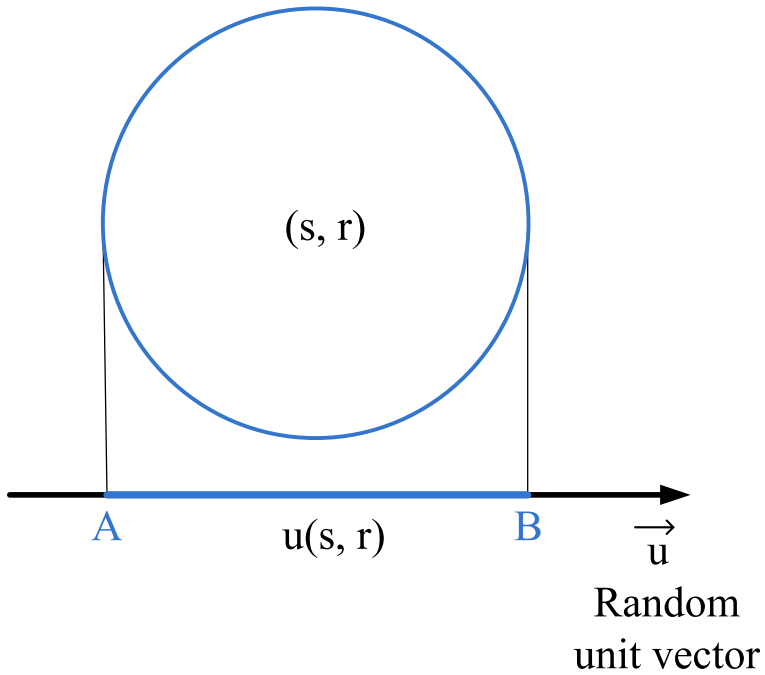
Choosing $f(x) = \frac{1}{\sqrt{k}} x^T U$, then with probability at least $1 - n^{-\beta}$

$$(1-\varepsilon) \parallel x - y \parallel^2 \leq \parallel f(x) - f(y) \parallel^2 \leq (1+\varepsilon) \parallel x - y \parallel^2$$

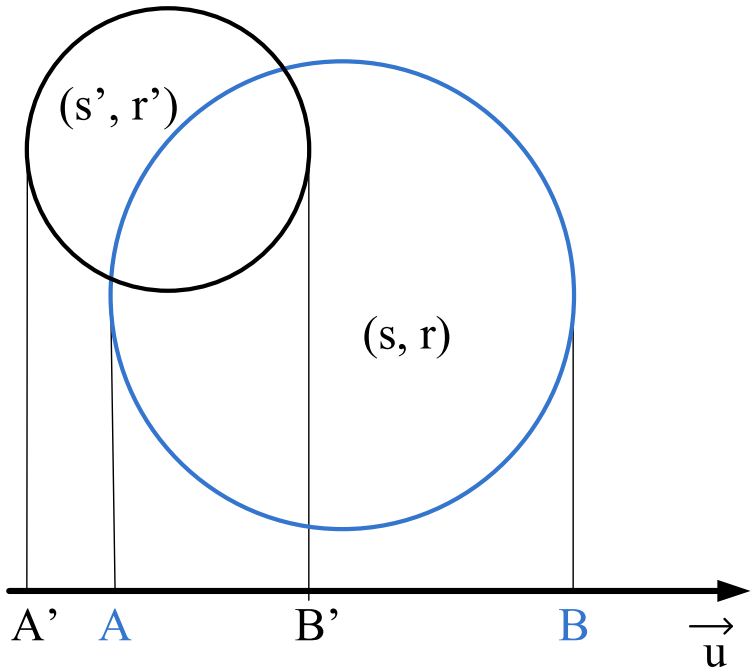
Random Projection: One Dimension

Let $\overrightarrow{u}$ be a random unit vector (i.e., $\| u \| = 1$)
Let the mapping be

$$(s, r) \rightarrow u(s, r) = [\langle u, s \rangle - r, \langle u, s \rangle + r]$$



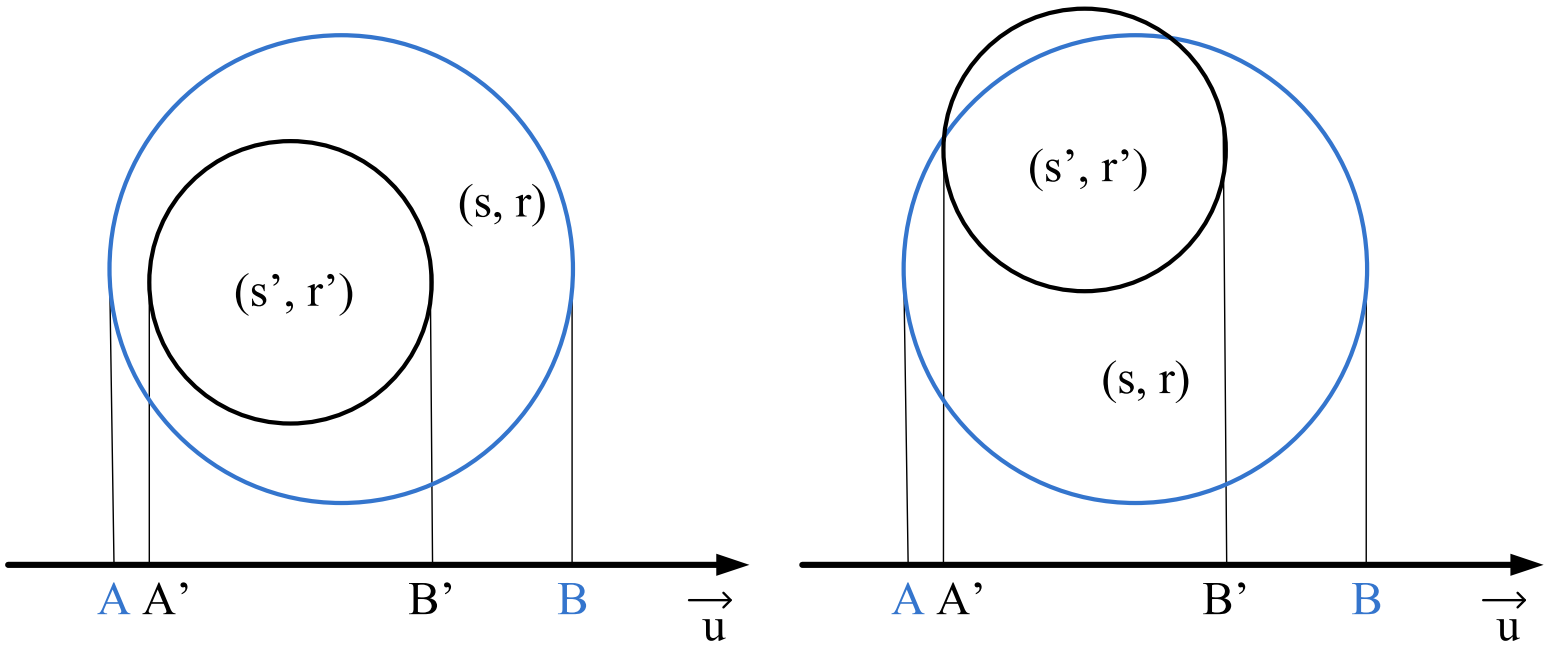
1-D Random Projection: Non-Ambiguous Case



Non-ambiguous case

If $u(s, r)$ and $u(s', r')$ do not have a covering relationship, neither do (s, r) and (s', r')

1-D Random Projection: Ambiguous Case



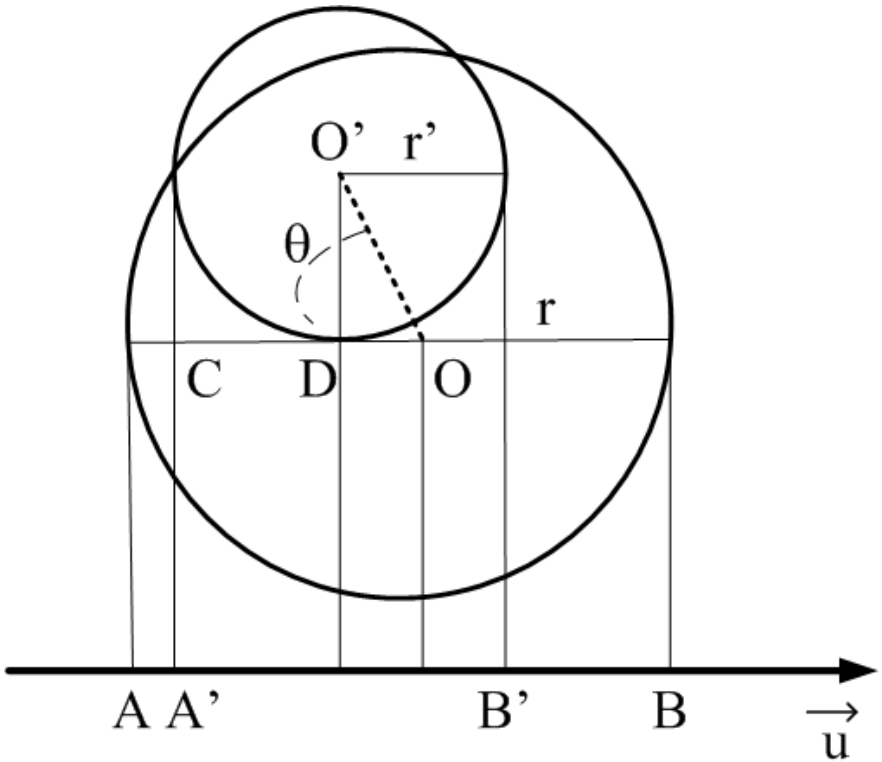
Ambiguous case

If $u(s,r) \supset u(s',r')$, cannot conclude that $(s,r) \supset (s',r')$

1-D Random Projection: Error Probability

Error decision

$u(s,r) \supseteq u(s',r') \Rightarrow (s,r) \supseteq (s',r')$ *but actually* $(s,r) \not\supseteq (s',r')$



$$AB = u(s,r) \supseteq u(s',r') = A'B'$$

$$\Leftrightarrow \quad \frac{r'-r}{\| s-s' \|} \leq \cos\theta \leq \frac{r-r'}{\| s-s' \|}$$

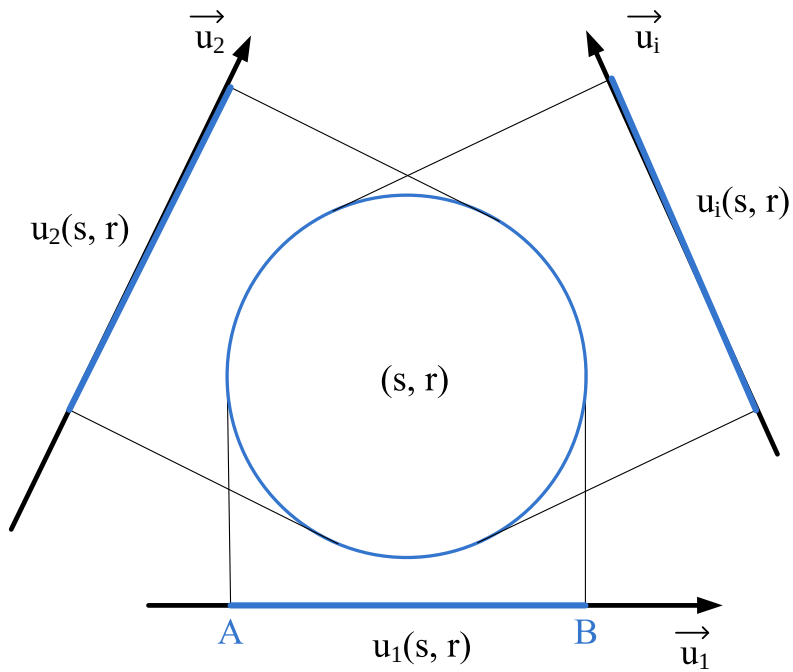
Assuming $\theta \approx$ uniform$[0, 2\pi]$:

$$\begin{aligned} Pr & \{error\} \\ = & \ 1 - \frac{2}{\pi} \arccos\left(\frac{r-r'}{\| s-s' \|}\right) \\ \simeq & \left(\frac{2}{\pi}\right) \frac{r-r'}{\| s-s' \|} \leq \frac{2}{\pi} \end{aligned}$$

Random Projection: k Dimensions

Let $\{\overrightarrow{u_1}, \overrightarrow{u_2}, ..., \overrightarrow{u_k}\}$ be k random unit vectors, i.i.d. generated
Let the mapping be

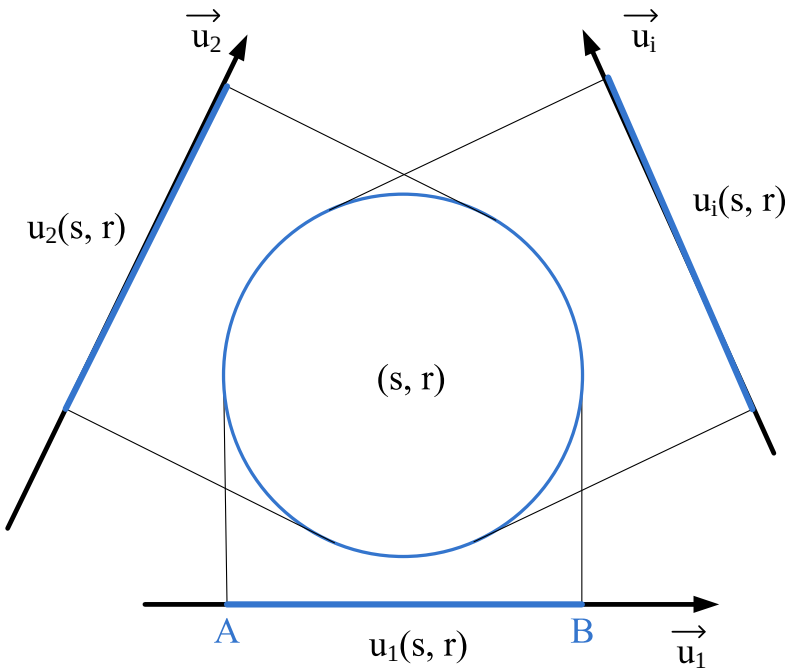
$$(s, r) \rightarrow \prod_{i=1}^{k} [\langle u_i, s \rangle - r, \langle u_i, s \rangle + r]$$



k-D Random Projection: Accuracy

Decision rules

1. If $\forall i$: $u_i(s,r) \supset u_i(s',r')$, conclude $(s,r) \supset (s',r')$
2. Else, conclude $(s,r) \not\supset (s',r')$

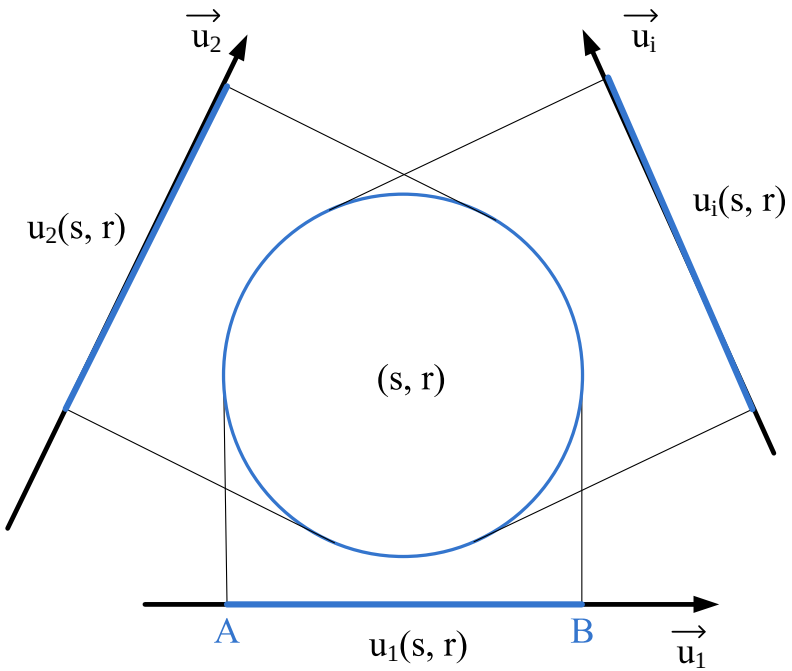


$$
\begin{aligned}
Pr & \{correct\} \\
= & \ Pr\{correct \text{ for at least one } u_i\} \\
= & \ 1 - Pr\{error \text{ for every } u_i\} \\
= & \ 1 - (2/\pi)^k \dashrightarrow 100\%
\end{aligned}
$$

k-D Random Projection: Accuracy

Decision rules

1. If $\forall i$: $u_i(s,r) \supset u_i(s',r')$, conclude $(s,r) \supset (s',r')$
2. Else, conclude $(s,r) \not\supset (s',r')$



$$
\begin{aligned}
Pr & \{correct\} \\
= & \ Pr\{correct \text{ for at least one } u_i\} \\
= & \ 1 - Pr\{error \text{ for every } u_i\} \\
= & \ 1 - (2/\pi)^k \dashrightarrow 100\%
\end{aligned}
$$

Data Structure

1. Index each subscription (s, r) by a $2k$-dimension point:

$$(s, r) \rightarrow idx(s, r) = \prod_{i=1}^{k} [\langle u_i, s \rangle - r, \langle u_i, s \rangle + r]$$

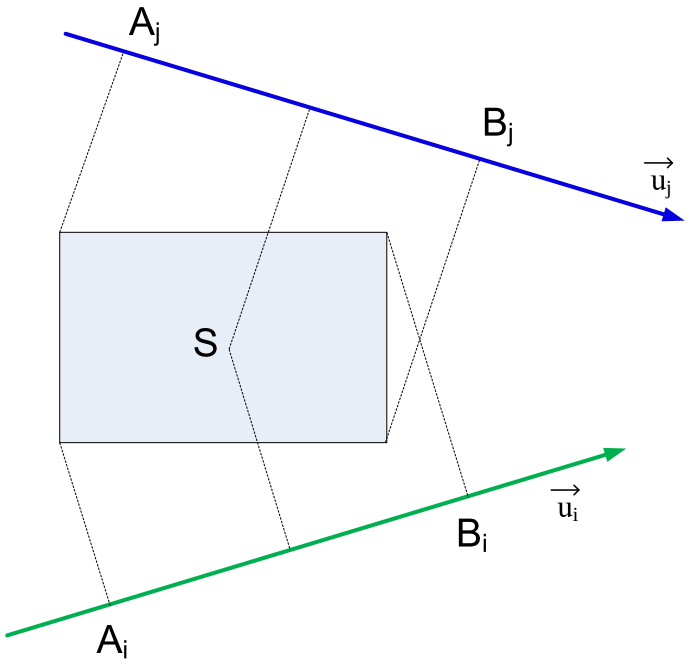
2. Store these indices using a data structure that supports orthogonal range searching in high dimension. If we use a $2k$-dimension layered range tree:
   - Building time for n subscriptions: $O(n \log^{2k-1} n)$
   - Update time to insert a new subscription or delete an existing subscription: $O(\log^{2k-1} n)$
   - Time to query coverings: $O(\log^{2k-1} n + m)$ where m is the number of coverings reported
   - Space complexity: $O(n \log^{2k-1} n)$

Algorithm

Input: A subscription (s, r)
Output: Whether to forward (s, r) or not

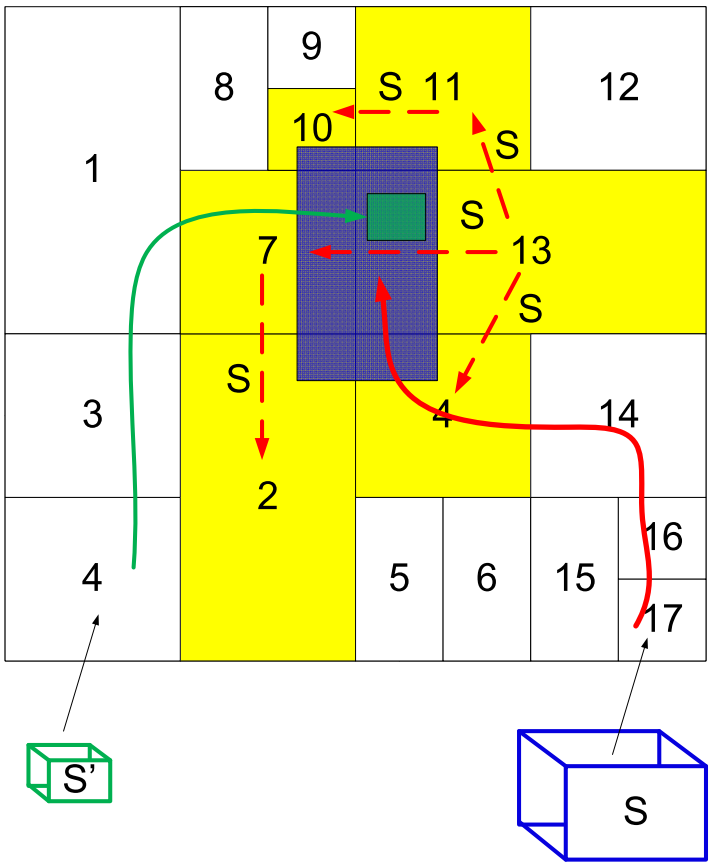
1. Compute $idx(s, r)$ and insert it to the index tree
2. Search the index tree to find *one* subscription (s', r') such that $idx(s', r')$ covers $idx(s, r)$
   1. IF no such (s', r') is found, forward (s, r)
   2. ELSE check the original covering condition
      1. IF $\| s - s' \| \leq r - r'$, stop forwarding (s, r)
      2. ELSE forward (s, r)
3. END

Rectangular Subscriptions



For a random unit vector u_i, a rectangular subscription is mapped to the smallest interval containing all the projections of the subscription vertices

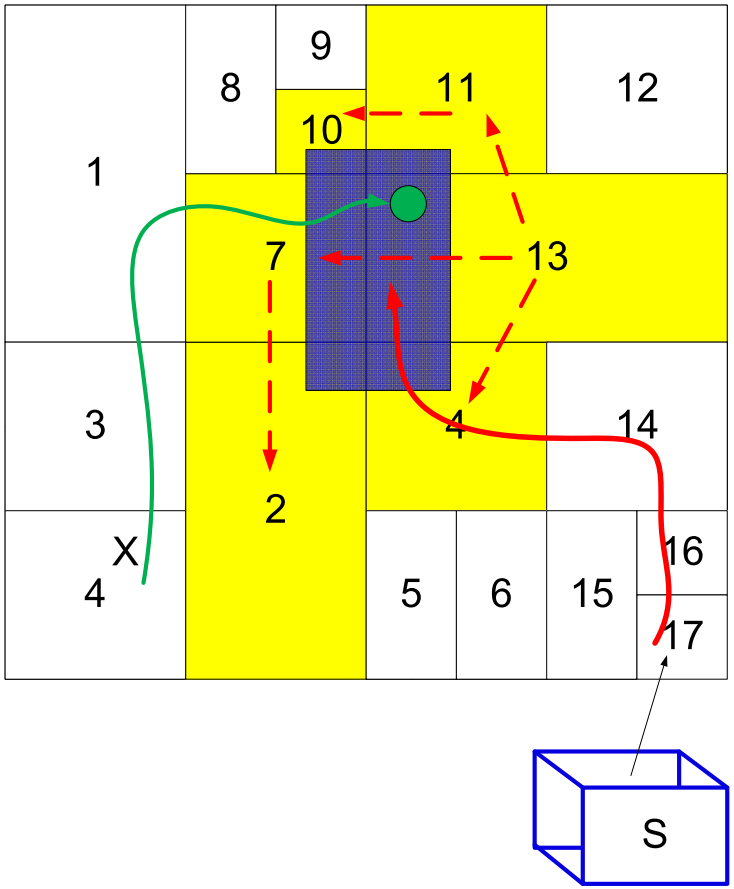
Subscription Routing



With random projection, we can map high-dimension subscription/event data into a low-dimension CAN-based P2P network

- A new sub S is routed to the node whose zone contains the center of $idx(S)$
- If sub S is not covered by any existing sub, it is also stored in all nodes whose zone intersects $idx(S)$
  - E.g., sub S is stored at node 13 and also nodes 2, 4, 7, 10, 11, but sub S' is stored at node 13 only

Event Routing



- A new event X is routed to the node whose zone contains $idx(X)$
- Thus, if an event X matches a sub S, they always find each other

Summary

- Subscription covering is an important optimization issue of content-based pub/sub networks
- Perfect detection of subscription covering in high dimensionality is inefficient

Contributions

- A random projection approach is proposed to detect subscription coverings approximately, however, very efficiently with high approximation accuracy
- A routing approach is proposed that takes into account subscription coverings

Future work:

- Need a better way to address rectangular subscriptions
- How is the solution applied to sensor networks?