Algorithms for the spherical coverage verification

Marko D. Petković[1], Dragoljub Pokrajac[2], Longin Jan Latecki[3], Janko Milutinović[2]

[1]Faculty of Science and Mathematics
University of Niš, Serbia
dexterofnis@gmail.com

[2]Department of Computer and Information Sciences
Delaware State University, Dover 19901, DE, USA
dragoljub.pokrajac@comcast.net, miljanko@yahoo.com

[3]Computer and Information Sciences Department
Temple University, Philadelphia, PA, USA
latecki@temple.edu

Theoretical Computer Science - From Foundation to Application
Niš, 2009

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Motivation

Reverse k-nearest neighbor problem.

Problem RkNN (Reverse k-nearest neighbor)

For the given set of points $p_1, p_2, \ldots, p_m \in \mathbb{R}^d$ and point $q \in \mathbb{R}^d$, determine all points p_i such that q is among k nearest neighbors of p_i.

Idea:

1. Find a 'small' set of candidate points S'.
2. For each point $x \in S'$, check if $q \in k - \mathrm{nearest\ neighbors}(x)$.

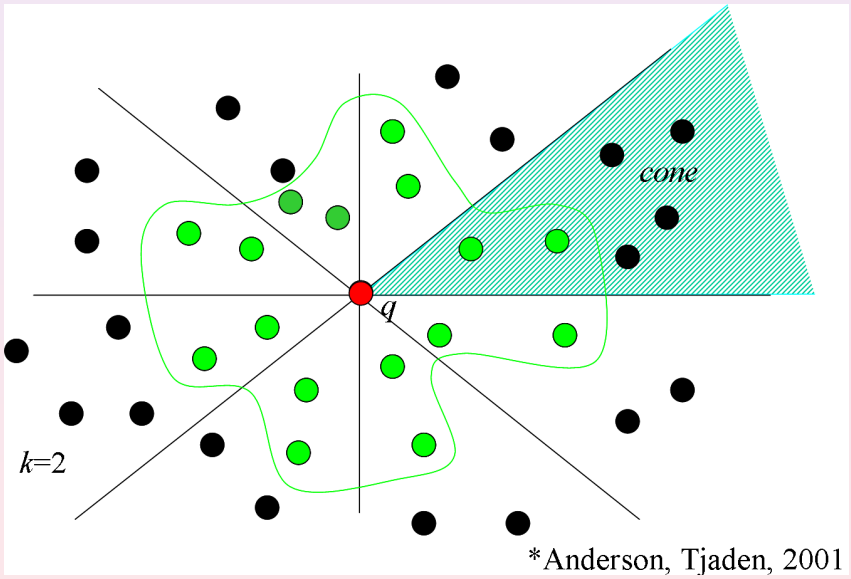
Points are stored using the indexing structures for spatial data (R-tree, R^*-tree ,X-tree, M-tree,...).

- J. Anderson, B. Tjanden, *The inverse nearest neighbor problem with astrophysical applications*, Proceedings of the 12th Symposium of Discrete Algorithms (SODA), Washington, DC, January 2001
- I. Stanoi, D. Agrawal, A. ElAbbadi, *Reverse Nearest Neighbor Queries for Dynamic Databases*, ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Dalas, TX, May 2000

Computation of candidate set

Divide space into the n cones C_i centered at the point q, with angle not larger than $\pi/6$.

In each cone C_i, find set of k points Q_i nearest to q. Set $S' = \bigcup_{i=1}^{n} Q_i$. Cardinality of S' is at most kn and does not depend on the total number of points m.



Formulation of the problem

Suppose that we have n cones $C_1, \ldots, C_n$ in d-dimensional space $\mathbb{R}^d$ centered at point $O = (0, \ldots, 0)$ and defined by:

$$C_i = C(t_i; \theta_i) = \left\{ x \in \mathbb{R}^d \mid \frac{(x, t_i)}{\|x\| \|t_i\|} \geq \theta_i \right\}.$$

Parameter θ_i is equal to $\theta_i = \cos(\alpha_i/2)$ where α_i is angle of the cone C_i.

Problem SphCoverVer (Spherical coverage verification)

Check if cones C_i cover the whole space $\mathbb{R}^d$. Equivalently, check if hypercaps $K_i = C_i \cap S_d(1)$ cover an unit hypersphere $S_d(1) = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$.

Set of the cones C_i covering the whole space $\mathbb{R}^d$ is called covering set (covering constellation). Otherwise it is called uncovering set (uncovering constellation)

We can assume that $\|t_i\| = 1$. Let $t_i = (t_{i1}, \ldots, t_{id})$. If $x \in S_d(1)$ then:

$$x \in K_i \iff \sum_{j=1}^{d} x_i t_{ij} \geq \theta_i$$

Minimal covering constellation for $\theta_i = \cos(\pi/6)$

Problem SphCoverMin

Find the minimal number of cones C_i centered at $O = (0, \ldots, 0)$, such that $\theta_i = \cos(\pi/6)$ ($\alpha_i = \pi/3$) and $\bigcup_{i=1}^{n} C_i = \mathbb{R}^d$.

Obviously, for $d = 2$ it is $n = 6$. For $d > 3$: hard problem!

Denote by $\mathbf{M}(d)$ solution of the Problem **SphCoverMin**. Our goal is to find lower bound $\mathbf{M}_l(d)$ for $\mathbf{M}(d)$.

Idea: consider the area of the hypercap with angle $\alpha = \pi/3$ and area of the hypersphere.

Theorem

A lower bound $\mathbf{M}_l(d)$ is given by

$$\mathbf{M}_l(d) = \frac{\int_0^{\pi} \sin^{d-2} \phi_1 d\phi_1}{\int_0^{\pi/6} \sin^{d-2} \phi_1 d\phi_1} = \Theta(2^d \sqrt{d})$$

Proof of the theorem

Using the hyperspherical coordinate system:

$$x_i = r\cos\varphi_i \prod_{j=1}^{i-1} \sin\varphi_j, \quad i = 1, \ldots, d-1, \quad x_d = r\prod_{j=1}^{d-1} \sin\varphi_j,$$

with Jacobian:

$$|J| = r^{d-1} \prod_{i=1}^{d-2} \sin^{d-1-i}\varphi_i$$

we obtain the following expression for the area of the hyperspherical cap with angle 2α:

$$\begin{aligned} A_{cap}(\alpha) &= \int_0^\alpha d\varphi_1 \int_0^\pi d\varphi_2 \cdots \int_0^\pi d\varphi_{d-2} \int_0^{2\pi} d\varphi_{d-1} \frac{|J|}{r^{d-1}} \\ &= \int_0^\alpha \sin^{d-2}\varphi_1 d\varphi_1 \underbrace{\int_0^\pi \sin^{d-3}\varphi_2 d\varphi_2 \cdots \int_0^\pi \sin\varphi_{d-2} d\varphi_{d-2} \int_0^{2\pi} d\varphi_{d-1}}_{C(d)} \\ &= C(d) \cdot \int_0^\alpha \sin^{d-2}\varphi_1 d\varphi_1 \end{aligned}$$

Proof of the theorem (continued)

Hence:

$$\mathbf{M}_l(d) = \frac{A_{cap}(\pi)}{A_{cap}(\pi/6)} = \frac{\int_0^{\pi} \sin^{d-2}\phi_1 d\phi_1}{\int_0^{\pi/6} \sin^{d-2}\phi_1 d\phi_1}$$

Numerator of the last expression can be exactly computed and is equal to:

$$\int_0^{\pi} \sin^{d-2}\phi_1 d\phi_1 = \begin{cases} 2\frac{(n-3)!!}{(n-2)!!}, & n \in 2\mathbb{N}+1, \\ \frac{(n-3)!!}{(n-2)!!}\pi, & n \in 2\mathbb{N} \end{cases}$$

Using Stirling's formula

$$\left(\frac{n}{e}\right)^n \sqrt{2n\pi} < n! < \left(\frac{n}{e}\right)^n \sqrt{2n\pi}\left(1 + \frac{1}{12n-1}\right),$$

we obtain $\int_0^{\pi} \sin^{d-2}\varphi_1 d\varphi_1 = \Theta\left(\frac{1}{\sqrt{d}}\right)$.

Proof of the theorem (continued)

On the other hand, we have

$$A_{cap}(\pi/6) = \int_0^{\pi/6} \sin^{d-2}\phi_1 d\phi_1 = \frac{1}{(d-1)2^{d-1}}\ {}_2F_1\left(\frac{1}{2}, \frac{d-1}{2}; \frac{d+1}{2}; \frac{1}{4}\right).$$

Using the properties of Gauss hypergeometric function ${}_2F_1$ and partial integration we can show that:

$$I_d = {}_2F_1\left(\frac{1}{2}, \frac{d-1}{2}; \frac{d+1}{2}; \frac{1}{4}\right) = \frac{2}{\sqrt{3}} - I'_d, \qquad I'_d = \frac{1}{8}\int_0^1 t^{(d-1)/2}\left(1-\frac{t}{4}\right)^{-3/2} dt.$$

Since $I'_d > 0$, there holds $I_d < \frac{2}{\sqrt{3}}$.
We prove $\lim_{d\to+\infty} I'_d = 0$ using Lebesgue's dominant convergence theorem.

Let $f_d(t) = t^{(d-1)/2}\left(1-\frac{t}{4}\right)^{-3/2}$ and $g(t) = \left(1-\frac{t}{4}\right)^{-3/2}$. Since $|f_d(t)| \le g(t)$ for every $d \in \mathbb{N}$ and $t \in [0,1]$, the sequence $(f_d(t))_{d\in\mathbb{N}}$ is dominated by $g(t)$. Moreover since $\int_0^1 g(t)dt = 8 - 4\sqrt{3}$, function $g(t)$ is integrable.

Proof of the theorem (continued)

Now we are ready to apply Lebesgue's dominant convergence theorem:

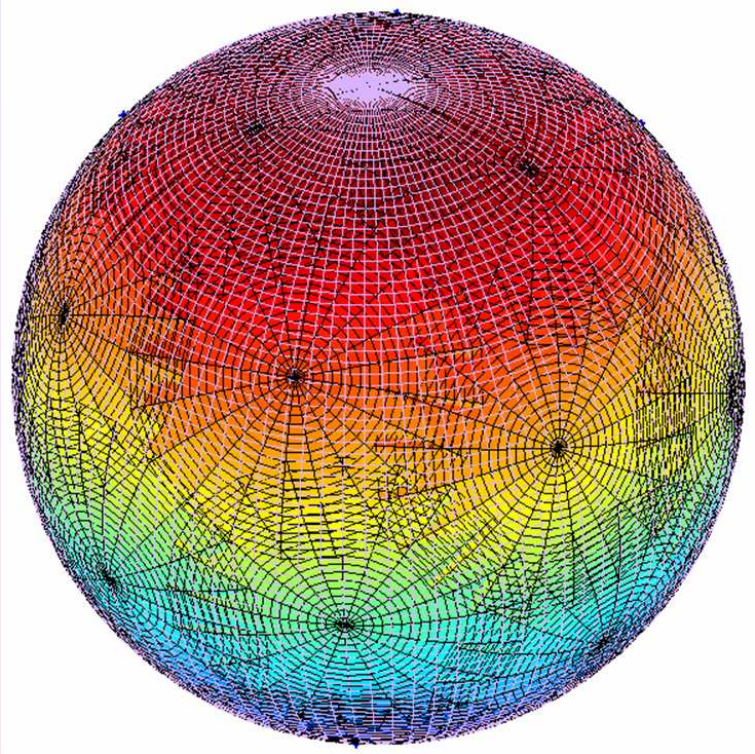
$$\lim_{d\to+\infty} I'_d = \lim_{d\to+\infty} \frac{1}{8}\int_0^1 f_d(t)dt = \frac{1}{8}\int_0^1 \lim_{d\to+\infty} f_d(t)dt = 0.$$

As a direct consequence we have $A_{cap}(d) = \Theta(d^{-1}2^{-d})$ and hence $\mathbf{M}_l(d) = \Theta(2^d\sqrt{d})$. This finishes the proof of the theorem. □

Corollary

Minimal number of cones with angle $\alpha = \pi/3$ required to cover $\mathbb{R}^d$ is $\mathbf{M}(d) = \Omega(2^d\sqrt{d})$.

Example



Picture shows the covering constellation of $n = 22$ cones for $d = 3$.
We need an effective method for checking if given constellation of cones $C_1, C_2, \ldots, C_n$ covers the space $\mathbb{R}^n$ or not!!!

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Reduction to quadratic programming

Recall that point x on the hypersphere $S_d(1)$ is not covered by cones C_i if and only if $(x, t_i) < \theta_i$ for all $i = 1, \ldots, n$. In other words, uncovered points are solutions of the following system:

$$\begin{aligned} x_1^2 + x_2^2 + \ldots + x_d^2 &= 1 \\ (x, t_1) = t_{11}x_1 + t_{12}x_2 + \ldots + t_{1d}x_d &< \theta_1 \\ &\vdots \\ (x, t_n) = t_{n1}x_1 + t_{n2}x_2 + \ldots + t_{nd}x_d &< \theta_n. \end{aligned} \tag{1}$$

First equation of (1) gives the objective function and other give constraints.

Let

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid (x, t_i) < \theta_i,\ i = 1, \ldots, n\}$$
$$\mathcal{S}' = \{x \in \mathbb{R}^n \mid (x, t_i) \leq \theta_i,\ i = 1, \ldots, n\}$$

Also let $f(x) = \|x\|^2$ and $M = f(x^*) = \max\{f(x) \mid x \in \mathcal{S}'\}$ (if $\mathcal{S}'$ is unbounded, then $M = +\infty$).

Reduction to quadratic programming

Theorem

System of equations and inequations (1) *has solutions if and only if* $M > 1$.

According to the last theorem, spherical coverage problem (Problem **SphCoverVer**) is reduced on the following quadratic programming (QP) problem:

$$\begin{aligned} \max M &= x_1^2 + x_2^2 + \ldots + x_d^2 \\ \text{s.t. } & t_{11}x_1 + t_{12}x_2 + \ldots + t_{1d}x_d \leq \theta_1 \\ & \vdots \\ & t_{n1}x_1 + t_{n2}x_2 + \ldots + t_{nd}x_d \leq \theta_n. \end{aligned} \tag{2}$$

Uncovered point can be computed by $u = x^*/\|x^*\|$.

Algorithm

Algorithm (Cover - QP)

Input: Caps K_i, $i = 1, \ldots, n$ defined by $t_i \in S_d(1)$ and $\theta_i \in (-1, 1)$.

1. *Solve the QP problem* (2).
2. *If the problem is not feasible, or unbounded or an optimal value x^* satisfies $M = \|x\|^2 > 1$, return* `True`*. Otherwise return* `False`*.*

Since the QP problems can be solved in the polynomial time, Algorithm **Cover - QP** is polynomial.

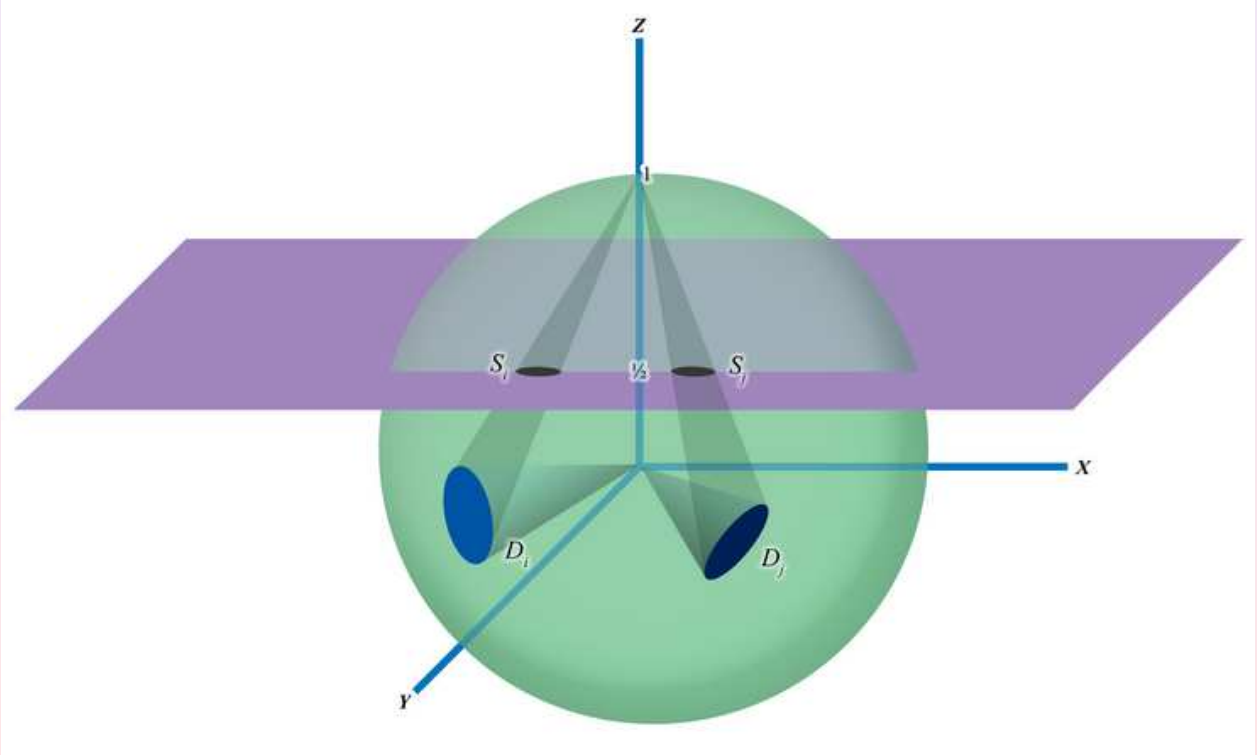
On the other side, problem (2) is concave (since it is maximization problem) and QP solvers should have a problem with local maximum trapping.

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Description of the method

- We describe recursive algorithm for solving the Problem **SphCoverVer**.
- The main idea is to reduce the starting d-dimensional problem on several $d-1$-dimensional problems.
- Such reduction is obtained by applying an inversion.



Description of the method

Definition

Let $c \in \mathbb{R}^D$ be a given point, and let R be a positive real number. Inversion $\psi_{c,R}(x)$ is a function $\psi_{c,R} : \mathbb{R}^D \to \mathbb{R}^D$ that maps every point $x \in \mathbb{R}^D$ to a point y such that:

$$y = c + \frac{R^2}{\|x - c\|^2}(x - c). \tag{3}$$

The point c is called the center of inversion $\psi_{c,R}$ and R is the radius of inversion.

The image of the unit hypersphere $S_d(1)$ is the hyperplane $x_1 = 1/2$.

Images of the caps K_i, $i = 1, \ldots, n$ consists of $d - 1$-dimensional hyperspheres, belonging to the hyperplane $x_1 = 1/2$ and their exteriors or interiors.

Lemma

Let $c = (1, 0, \ldots, 0) \in \mathbb{R}^d$ and assume that $c \notin D_i$ for every $i = 1, \ldots, n$. Image of D_i, by an inversion $\psi_{c,1}$ is a $d-1$-dimensional hypersphere S_i with center $\beta_i = (\beta_{i1}, \ldots, \beta_{id})$ and radius r_i. Moreover, the image of K_i is $R_i = S_i \cup \mathrm{ext} S_i$, if $c \in K_i$ and $R_i = S_i \cup \mathrm{int} S_i$, if $c \notin K_i$. Values β_i and r_i are given by the following expressions:

$$\beta_{i1} = \frac{1}{2}, \quad \beta_{ij} = \frac{t_{ij}}{2(\theta_i - t_{i1})}, \quad r_i = \frac{\sqrt{1 - \theta_i^2}}{2(\theta_i - t_{i1})}. \tag{4}$$

Due to the previous lemma, Problem **SphCoverVer** reduces to the following:

Problem SpcCoverVer

For given $d-1$-dimensional hyperspheres S_i, with center β_i and radius r_i contained in the hyperplane $x_1 = 1/2$, and sets R_i such that $R_i = S_i \cup \mathrm{int} S_i$ or $R_i = S_i \cup \mathrm{ext} S_i$ ($i = 1, \ldots, n$), check if sets R_i covers the whole hyperplane.

If $R_i = S_i \cup \mathrm{int} S_i$ we call set R_i *internal* and otherwise we call it *external*.

If all the sets R_i are internal, then we can immediately conclude that the answer to the Problem **SphCoverVer** is `False`.

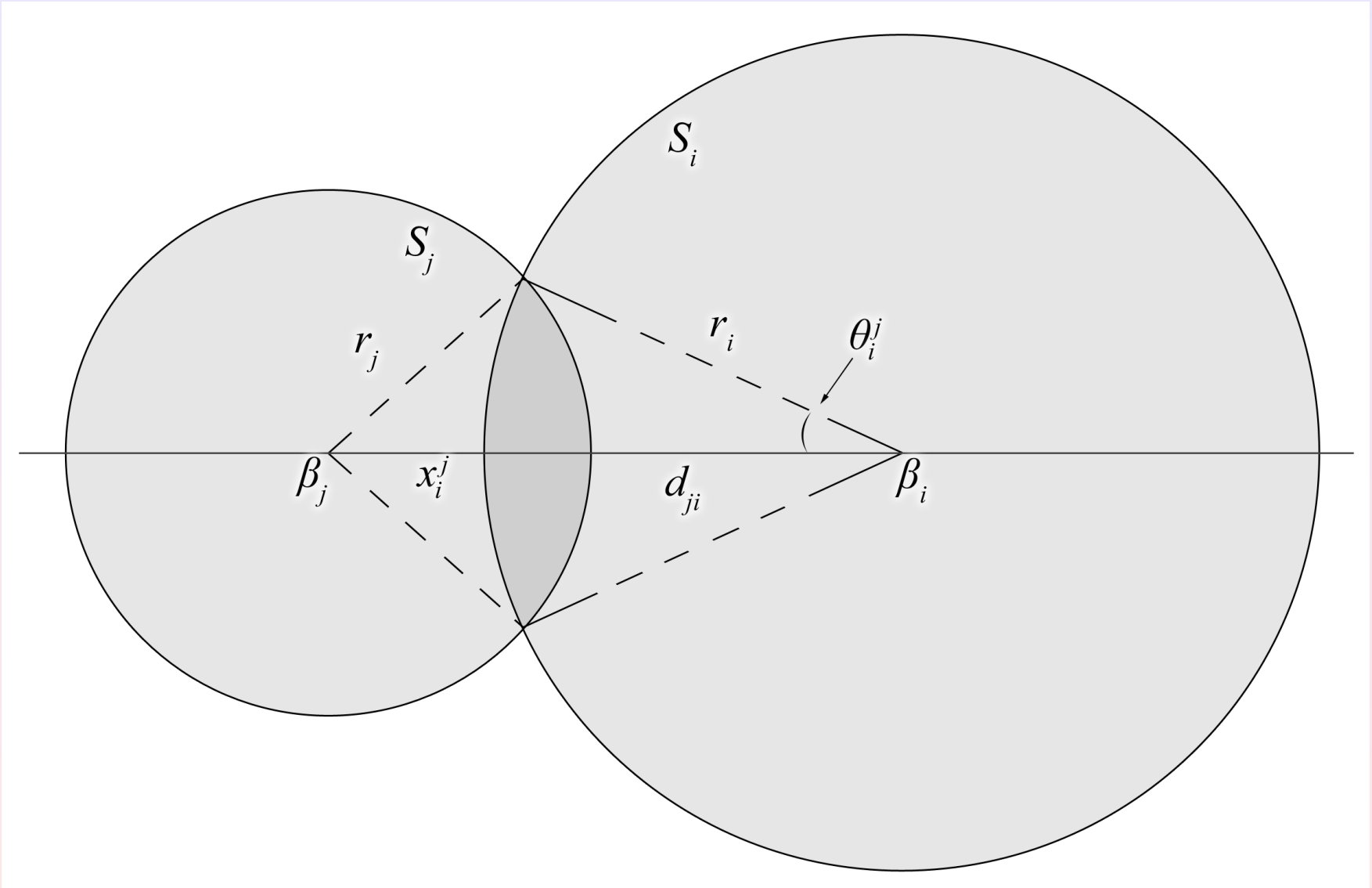
Theorem

Sets $R_1, \ldots, R_n$ cover the whole hyperplane $x_1 = 1/2$ if and only if every hypersphere $S_i = \partial R_i$ is covered by other sets R_j, $j \neq i$, i.e.

$$S_i = \partial R_i \subset \bigcup_{j \neq i} R_j.$$

Recall that by ∂X we denote the boundary of a given set X.

According to the last theorem we need to check whether all hyperspheres S_i are covered by sets R_j, $j \neq i$.


S_i
S_j
r_j
r_i
θ_i^j
β_j
x_i^j
d_{ji}
β_i

Case 1. Hyperspheres S_i and S_j have a nonempty intersection. In such case $S_i \cap R_j$ is a hypercap K_j^i defined as:

$$K_j^i = \left\{ x \in S_i \mid \frac{(x - \beta_i, x_j^i)}{\|x - \beta_i\| \|x_j^i\|} \geq \theta_j^i \right\}, \quad i \neq j,$$

where we define:

$$\begin{aligned} \theta_j^i &\triangleq \begin{cases} \frac{r_i^2 + d_{ij}^2 - r_j^2}{2 r_i d_{ij}}, & R_j \text{ is internal} \\ -\frac{r_i^2 + d_{ij}^2 - r_j^2}{2 r_i d_{ij}}, & R_j \text{ is external} \end{cases} \\ x_j^i &\triangleq \begin{cases} d_{ij}^{-1}(\beta_j - \beta_i), & R_j \text{ is internal} \\ -d_{ij}^{-1}(\beta_j - \beta_i), & R_j \text{ is external} \end{cases}, \quad d_{ij} \triangleq \|\beta_i - \beta_j\|, \quad i \neq j. \end{aligned} \tag{5}$$

Case 2. Hyperspheres S_i and S_j are disjoint. Equivalent condition is $\theta_j^i \notin (-1, 1)$, where θ_j^i is defined by (5). In such case holds either $S_i \subset R_j$ or $S_i \cap R_j = \emptyset$.

According to the above discussion, we need to check if caps K_j^i cover the hypersphere S_i. This is an instance of Problem **SphCoverVer**, for dimension $d - 1$.

Hence, we reduce original Problem **SphCoverVer** to at most $n - 1$ equivalent problems of dimension $d - 1$.

Base case

As the base case, we choose the case $d = 2$ of Problem **SpcCoverVer**.

We assume that there are given sets R_i such that $R_i = [c_i, d_i]$ (R_i is internal) or $R_i = \mathbb{R} \setminus (c_i, d_i)$ (R_i is external) for some real numbers $c_i < d_i$ ($c_i = \beta_{i2} - r_i$ and $d_i = \beta_{i2} + r_i$).

The problem is to check if sets R_i covers the whole real line $\mathbb{R}$, i.e. $\bigcup_{i=1}^{n} R_i = \mathbb{R}$!

Assume that $R_1, \ldots, R_s$ be external and $R_{s+1}, \ldots, R_n$ internal. Then

$$\mathbb{R} \setminus \bigcup_{i=1}^{s} R_i = (c', d'), \qquad c' = \max_{i=1,\ldots,s} c_i, \quad d' = \min_{i=1,\ldots,s} d_i.$$

We need to check if an interval (c', d') is covered by $[c_{s+1}, d_{s+1}], \ldots, [c_n, d_n]$.

We can do this by sorting last segments with respect to c_i and sequentially shortening target interval (c', d').

Algorithm (Cover2-2D)

Input: Values β_{i2}, r_i ($r_i \geq 0$) and out_i for $i = 1, \ldots, n$. We assume that $\mathrm{out}_i = \texttt{True}$ if R_i is external and otherwise $\mathrm{out}_i = \texttt{False}$.

1. *Let $c_i = \beta_{i2} - r_i$ and $d_i = \beta_{i2} + r_i$*
2. *Reorder the sets R_i such that $\mathrm{out}_i = \texttt{True}$ for $i = 1, \ldots, s$ and $\mathrm{out}_i = \texttt{False}$ for $i = s + 1, \ldots, n$.*
3. *Sort the intervals (c_i, d_i) for $i = s + 1, \ldots, n$ in ascending order with respect to the value c_i.*
3. *Let $c' := \max_{i=1,\ldots,s} c_i$ and $d' := \max_{i=1,\ldots,s} d_i$*
4. *For every $i = s + 1, \ldots, n$ do the following:*
   - 4.1. *If $c' \geq d'$ return $\texttt{True}$. Otherwise continue.*
   - 4.2. *If $c_i > c'$ then return $\texttt{False}$. Otherwise set $c' := \max\{c', d_i\}$ and continue.*
5. *If $c' \geq d'$ return $\texttt{True}$, otherwise return $\texttt{False}$.*

Note that the complexity of the Algorithm **Cover2-2D** is $\mathcal{O}(n \log n)$ if quicksort (or any other $\mathcal{O}(n \log n)$ sorting method) is used for intervals sorting.

Algorithm (Cover)

Input: Caps K_i, $i = 1, \ldots, n$ defined by $t_i \in S_d(1)$ and $\theta_i \in (-1, 1)$.

1. *If $n = 1$ then return* `False`.
2. *Let $c = (1, 0, \ldots, 0) \in \mathbb{R}^d$. Check if $c \in \partial K_i$ for some i. In such case, rotate the whole hypersphere by some small angle δ in x_1x_2 plane such that $c \in \delta K_i$, $i = 1, \ldots, n$ (condition of Lemma 1).*
3. *Compute the vectors β_i and values r_i using eq. (4). If $t_{i1} > \theta_i$ ($c \in K_i$) set $\mathrm{out}_i =$* `True`*, otherwise set $\mathrm{out}_i =$* `False`*. If there holds $\mathrm{out}_i =$* `False` *for every $i = 1, \ldots, n$ then return* `False`.
4. *If $d = 2$ apply Algorithm* **Cover2-2D** *for β_{i2}, r_i and out_i, $i = 1, \ldots, n$ and return obtained value. Otherwise continue.*
5. *For every $i = 1, \ldots, n$ do the following:*
   - 5.1 *Determine x_j^i and θ_j^i (relations (5)), for every $j \neq i$, $j = 1, \ldots, n$.*
     If $\theta_j^i \notin (-1, 1)$, let $A = x_j^i + (r_i, 0, \ldots, 0)$. If point A belongs to R_j, set $i = i + 1$ and go to step 5. Otherwise continue.
     If $\theta_j^i \in (-1, 1)$, form the cap K_j^i. If $\mathrm{out}_j =$ `True` *then set $x_j^i = -x_j^i$ and $\theta_j^i = -\theta_j^i$.*
   - 5.2 *Apply Algorithm* **Cover** *on the set of all formed caps K_j^i.*

Localization of uncovered point

Algorithm **Cover** can be adopted to compute (l, d)-boundary point for uncovering constellation.

Point $u \in S_d(1)$ is (l, d)-boundary point if it belongs to an intersection of l boundaries $D_i = \partial K_i$, but $u \notin \mathrm{int} K_i$ for every $i = 1, \ldots, n$.

Let $C_i^\alpha = C(t_i; \theta_i - \alpha)$, for $i = 1, \ldots, n$. In other words, let the cones C_i be slightly widened by a small value α. If the enlarged constellation is also uncovering, (l, d)-boundary point of enlarged constellation is uncovered point of original.

Algorithm (FindUncoveredPoint)

Input: Caps K_i, $i = 1, \ldots, n$ defined by $t_i \in S_d(1)$ and $\theta_i \in [-1, 1]$.

1. *Apply an Algorithm* **Cover** *on values t_i and θ_i. If the result is* `True`*, return* `True`*. Otherwise continue.*
2. *Set $\alpha = 0.01$.*
3. *Set $\theta_i^\alpha = \theta_i - \alpha$. Apply an Algorithm* `Cover` *on values t_i^α and θ_i and compute (l, d)-boundary point u^d.*
4. *If the result is* `True`*, set $\alpha = 0.9\alpha$ and go to step 2. Otherwise return* `False` *and point u^d.*

Outline of the talk

1. Introduction and preliminaries
   - Motivation
   - Formulation of the problem
   - Minimal covering constellation for $\theta_i = \cos(\pi/6)$
2. Quadratic programming approach
   - Reduction to quadratic programming
   - Algorithm
3. Recursive algorithm
   - Description of the method
   - Base case
   - Complete algorithm
   - Localization of uncovered point
4. Numerical examples

Algorithm **Cover** is implemented in `C`. Algorithm **Cover-QP** is implemented in `Mathematica`, `Matlab` and it is written AMPL model.

Results of testing Algorithm **Cover-QP** on constellation `four_d_85` ($n = 85$, $d = 4$):

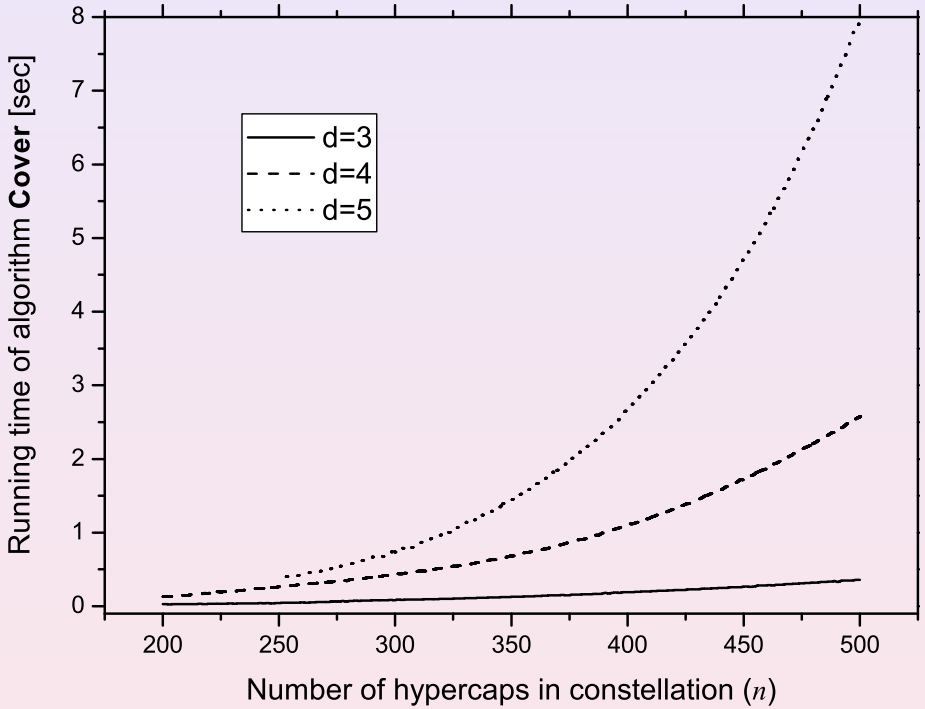
	Objective function value	Optimal point			
		x_1	x_2	x_3	x_4
`Matlab`	0.9756	0.9477	-0.0063	-0.0004	0.2314
`Mathematica`	0.935851	0.172452	0.898297	-0.158036	-0.272394
`Mathematica`: 20 digits	0.891469	-0.390757	0.136777	0.77625	0.342789
`Mathematica`: 50 digits	0.946319	0.652345	-0.476475	0.236943	0.487436
`Mathematica`: 100 digits	0.958959	-0.901428	0.034657	-0.231813	0.302405
`MINOS`	0.953615	0.628285	0.469643	0.426044	-0.39597
`FortMP`	0.947382	0.037172	0.869515	0.0528789	0.224396

This example clearly shows the instability of the Algorithm **Cover - QP**.

On the same example, Algorithm **Cover** returns `False` and following uncovered point:

$$x^{\text{uncovered}} = (0.134309, 0.457496, -0.791181, -0.383002).$$

We test the working time of the Algorithm **Cover**. Implementation is compiled by GNU C compiler and run on the machine with AMD Phenom II CPU on 3.0 GHz and CentOS 5.2 (Linux) operating system.



All testing constellations had coverage (output of the Algorithm **Cover** was `True`). As it can be seen from the graph, Algorithm **Cover** is practically applicable and does not reach theoretically obtained complexity $\mathcal{O}\left(n^{2d-1}\log n\right)$.

Upper bound for $\mathbf{M}(d)$

Algorithm **Cover** is used to determine the upper bound $\mathbf{M}_u(d)$ of the the minimal number $\mathbf{M}(d)$ of hypercaps with $\theta = \cos(\pi/6)$, covering the unit hypersphere.

The following table shows the values of $\mathbf{M}_u(d)$ for $d = 3, 4, 5, 6$:

d	3	4	5	6
$\mathbf{M}_u(d)$	22	81	234	715

However, the closed-form expression for some upper bound $\mathbf{M}_u(d)$ is still unknown and its existence is left as the open question.

Future work

1. Design of synchronous satellite systems: Determine how many satellites with directional antennae are required to cover the Earth.
2. Find analytical expression for $\mathbf{M}_u(d)$ or even $\mathbf{M}(d)$.
3. Parallel implementation of Algorithm **Cover**.

Publications and acknowledgement

- D. Pokrajac, M.D. Petković, L.J. Latecki, A. Lazarević, N. Reljin, J. Milutinović, *Computational Geometry Issues of Reverse Nearest Neighbor Algorithm*, Hawaii International Conference on Statistics, Mathematics and Related Fields, Honolulu, HI, January 2008.
- M.D. Petković, D. Pokrajac, L.J. Latecki, *Algorithms for the spherical coverage verification*, submitted for publication.

Authors wish to thank to Samantha McDaniel for help on drawing the figures.

Thanks for attention!