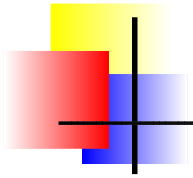
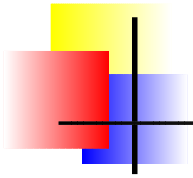


Computing Core-Sets and Approximate Smallest Enclosing HyperSpheres in High Dimensions

Piyush Kumar & Joseph S.B. Mitchell & Alper Yıldırım
{piyush,jsbm,yildirim}@ams.sunysb.edu

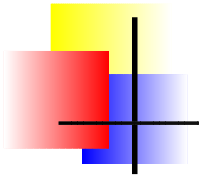
http://www.compgeom.com/meb/

Department of AMS, SUNY Stony Brook

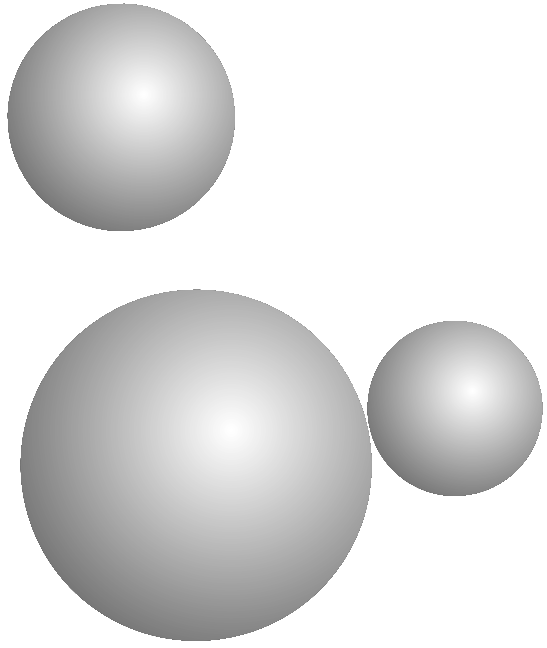


Talk Outline

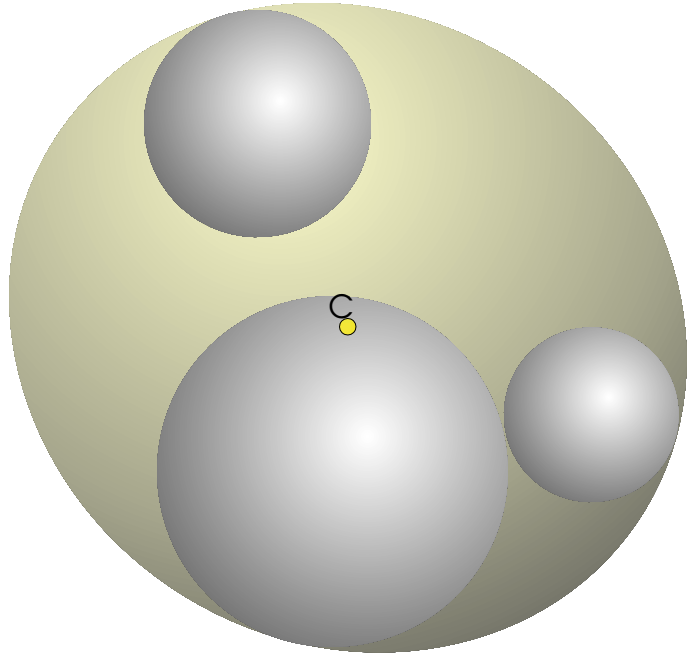
- Introduction
- SOCP Formulation
- Using Core-Sets for Approximating the MEB
- Implementation and Experiments
- Open Problems

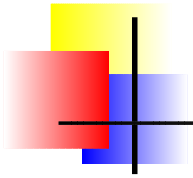


Introduction



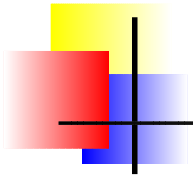
Introduction





Motivation

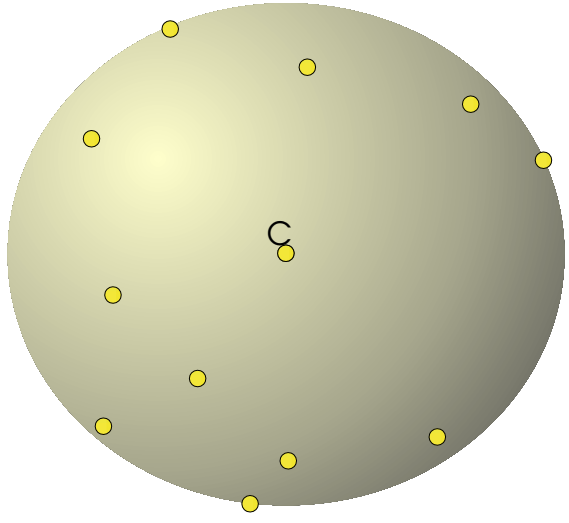
- Gap tolerant classifiers [B98[8]]
- Tuning Support Vector Machines [CVBM02[10]]
- Support Vector Clustering [CVBM02[5],BJKS03[3]]
- Fast farthest neighbor query approximation [GIV01[17]]



Motivation

- k-center clustering [BHI02[5]]
- Testing of radius clustering for $k = 1$ [ADPR00[2]]
- Approximate 1-cylinder problem [BHI02[5]]
- Sphere trees [H96[19]]
- Other applications [EH72[13]]

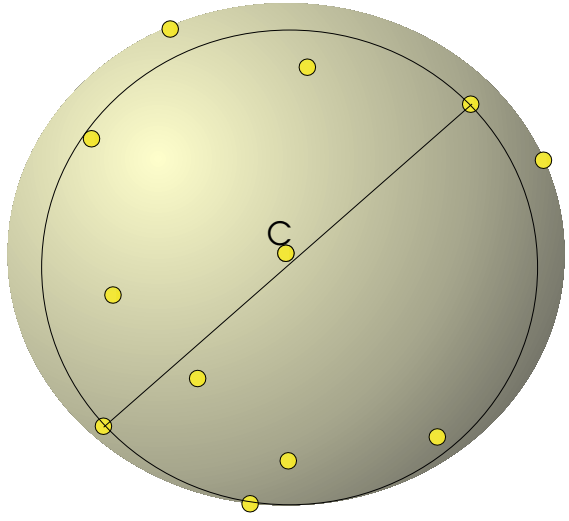
Core Sets



X is a core set for $S = \{p_1, p_2, ...p_n\}$ if

- $X \subseteq S$

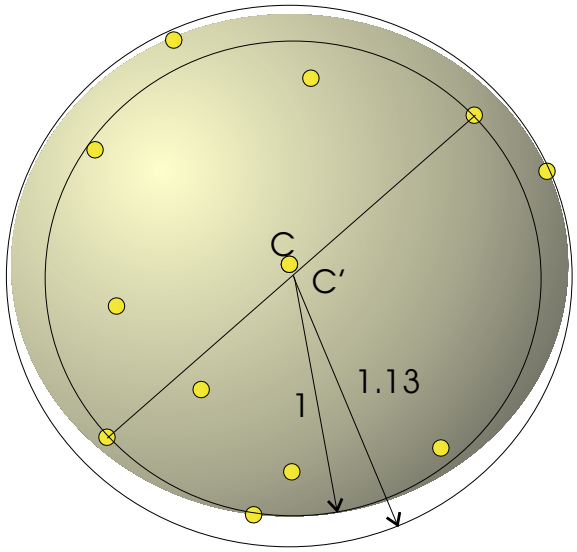
Core Sets



X is a core set for $S = \{p_1, p_2, ...p_n\}$ if

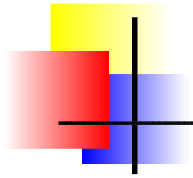
- $X \subseteq S$
- $B_{c',r} = \mathsf{MEB}(X)$

Core Sets



X is a core set for $S = \{p_1, p_2, ...p_n\}$ if

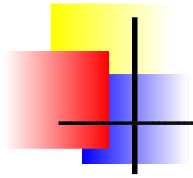
- $X \subseteq S$
- $B_{c',r} = \mathsf{MEB}(X)$
- $B_{c',(1+\epsilon)r} \supset S$ for $\epsilon > 0$



Related Work

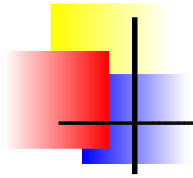
- LP-type problem, $\mathcal{O}(c^{f(d)}n)$ solution [MSW92[22], Gärtner[15]; CGAL[a]]
- $\mathcal{O}(d^3 n \log \frac{1}{\epsilon})$ solution, [GLS88[18]]
- Fast Implementations in high dimensions :
  - Simplex based [Gärtner and Schönherr[16]]
  - SOCP based [ZST02[34]]

[a] http://www.cgal.org



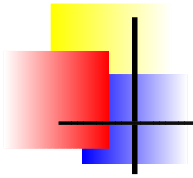
Related Work

- Core Set Sizes :
  - $\mathcal{O}(\frac{1}{\epsilon^2})$ [BHI02[5]]
  - $\mathcal{O}(\frac{1}{\epsilon})$ [Bădoiu and Clarkson[6], KMY03]
- Quadratic Programming for MEBs :
  - $\mathcal{O}(d^3 n \log \frac{1}{\epsilon})$ solution, [GLS88[18]]
  - $\mathcal{O}(\sqrt{n} d^2 (n+d) \log(1/\epsilon))$ [KMY03]



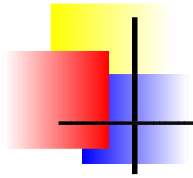
Results

- Worst Case Run Times:
  - $\mathcal{O}\left(\frac{dn}{\epsilon^2} + \frac{1}{\epsilon^{10}} \log \frac{1}{\epsilon}\right)$ [BHI02[5]]
  - $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^5}\right)$ [BC03[6]]
  - $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^{4.5}} \log \frac{1}{\epsilon}\right)$ [KMY03]
  - $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^4} \log^2 \frac{1}{\epsilon}\right)$ [S03[28],KMY03]
- k-center clustering, ($2^{\mathcal{O}(\frac{k \log k}{\epsilon})} dn$) [BC03[6], KMY03]



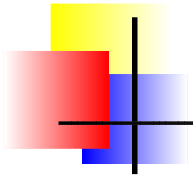
Results

- In Practice :
  - Core Set Sizes:
    - Dependent on dimension!
    - Very Weak dependence on ϵ!
    - $\leq \min\{d+1, \frac{1}{\epsilon}\}$!



Results

- In Practice :
  - Core Set Sizes:
    - Dependent on dimension!
    - Very Weak dependence on ϵ!
    - $\leq \min\{d+1, \frac{1}{\epsilon}\}$!
  - Run Times:
    - Much smaller than Worst Case.
    - Weakly dependent on epsilon.

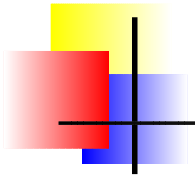


SOCP Formulation

Second Order Cone Program is of the form

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & ||A_i x + b_i||_2 \leq c_i^T + d_i, \quad i = 1..n \\ & Fx = g \end{aligned}$$

- $x \in \mathbb{R}^d$
- LP is a special case
- new IP methods can solve (almost) as fast as LPs



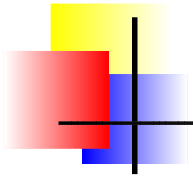
SOCP Formulation

MEB as SOCP

$$\min_{c,r} \; r, \quad \text{s.t.} \quad \|c - p_i\| \leq r$$

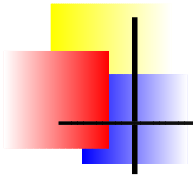
$$i = 1, \ldots, n$$

- Number of iterations = $\mathcal{O}(\sqrt{n}\log(1/\epsilon))$, In Practice ≤ 20, very weak dependence on n.
- IP solves it in $\mathcal{O}(\sqrt{n}d^2(n+d)\log(1/\epsilon))$



Why Core Sets?

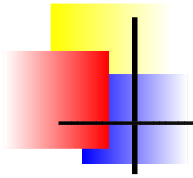
- IP solves it in $\mathcal{O}(\sqrt{n}d^2(n+d)\log(1/\epsilon))$.
- To make a practical algorithm, we need a way to reduce either d or n.
- We reduce n to $\mathcal{O}(\frac{1}{\epsilon})$ using core sets.
- $n = \mathcal{O}(\frac{1}{\epsilon}) \Rightarrow d = \mathcal{O}(\frac{1}{\epsilon})$.



The Core Set Algorithm: $\mathcal{O}(\frac{1}{\epsilon^2})$

Require: Input: $S \in \mathbb{R}^d$, $\epsilon > 0$, $X_0 \subset S$

1: $X \leftarrow X_0$
2: **loop**
3: Compute $B_{c,r} = \text{MEB}(X)$ using SOCP
4: **if** $S \subset B_{c,(1+\epsilon)r}$ **then**
5: Return $B_{c,r}$, X
6: **else**
7: $p \leftarrow$ point $q \in S$ maximizing $||cq||$
8: **end if**
9: $X \leftarrow X \cup \{p\}$
10: **end loop**



The Core Set Algorithm

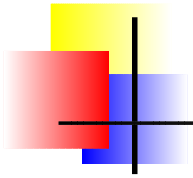
- Use SDPT3[a] to solve SOCP. [TTT99[30]]
- Implementation Uses random sampling in Step 7.
- I/O Efficient under mild assumptions.
- Works for Balls,Points

[a] http://www.math.nus.edu.sg/~mattohkc/sdpt3.html

Better Core Set Algorithm: $\mathcal{O}(\frac{1}{\epsilon})$

Require: Input: $S \in \mathbb{R}^d$, $\epsilon = 2^{-m}$, $X_0 \subset S$

1: **for** $i = 1$ to m **do**
2: Call Algorithm 1 with input S, $\epsilon = 2^{-i}$, X_{i-1}
3: $X_i \leftarrow$ the output core-set
4: **end for**
5: Return MEB$(X_m), X_m$

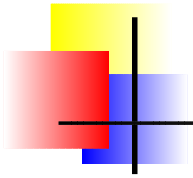


Better Core Set Algorithm: $\mathcal{O}(\frac{1}{\epsilon})$

Lemma: The number of points added to X in round $i+1$ is at most 2^{i+6}.

Theorem: The core-set output by Algorithm 2 has size $\mathcal{O}(1/\epsilon)$.

Proof: $|X_m| = \sum_{i=1}^{m} 2^{i+6} = \mathcal{O}(2^m) = \mathcal{O}(1/\epsilon)$.



Better Core Set Algorithm: $\mathcal{O}(\frac{1}{\epsilon})$

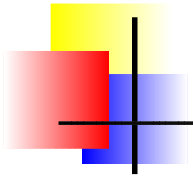
Theorem: A $(1+\epsilon)$-approximation to the MEB of a set of n balls in d dimensions can be computed in time $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^{4.5}} \log \frac{1}{\epsilon}\right)$.

Proof: SOCP $\Rightarrow \mathcal{O}\left(\frac{d^2}{\sqrt{\epsilon}}\left(\frac{1}{\epsilon}+d\right)\log\frac{1}{\epsilon}\right)$

We parse thru the input $\mathcal{O}(\frac{1}{\epsilon})$ times

$\Rightarrow \mathcal{O}\left(\frac{nd}{\epsilon} + \frac{d^2}{\epsilon^{3/2}}\left(\frac{1}{\epsilon}+d\right)\log\frac{1}{\epsilon}\right)$.

Now put $d = \mathcal{O}(1/\epsilon)$ to get a total bound of $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^{4.5}} \log \frac{1}{\epsilon}\right)$.

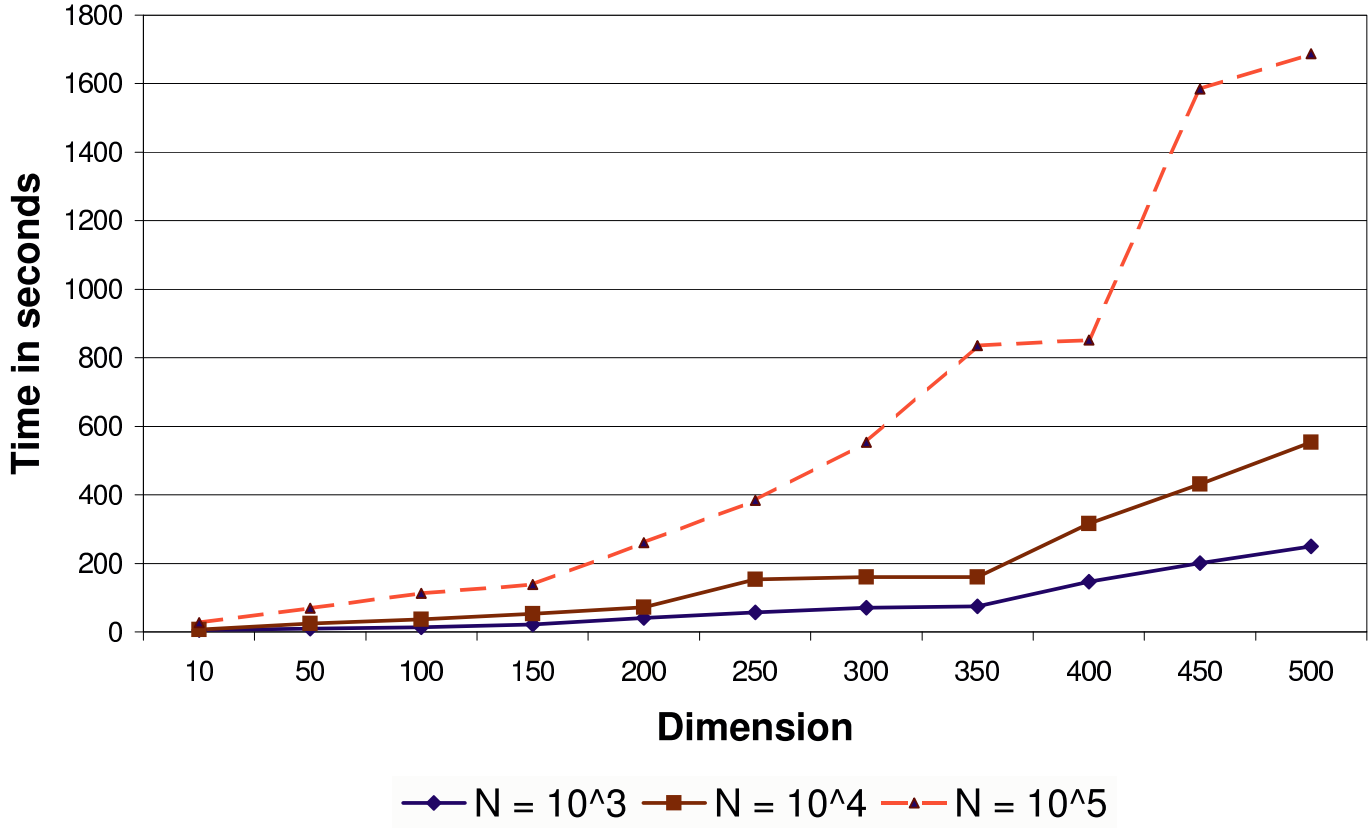


$\mathcal{O}(\frac{dn}{\epsilon^2})$ Algorithm [BC03[6]]

Require: A point set $S = \{p[1], p[2], ..., p[n]\} \in \mathbb{R}^d$

1: $i \leftarrow \text{random}(1, n)$
2: Choose $p[j] \in S$ farthest from $p[i]$
3: Choose $p[k] \in S$ farthest from $p[j]$
4: $c_3 = \frac{1}{2}(p[j] + p[k])$
5: **for** $i = 3..iter$ **do**
6: Find farthest point $p \in S$ from c_i
7: $c_{i+1} \leftarrow (1 - \frac{1}{i+2})c_i + \frac{1}{i+2}p$
8: **end for**
9: Return c_{iter+1}

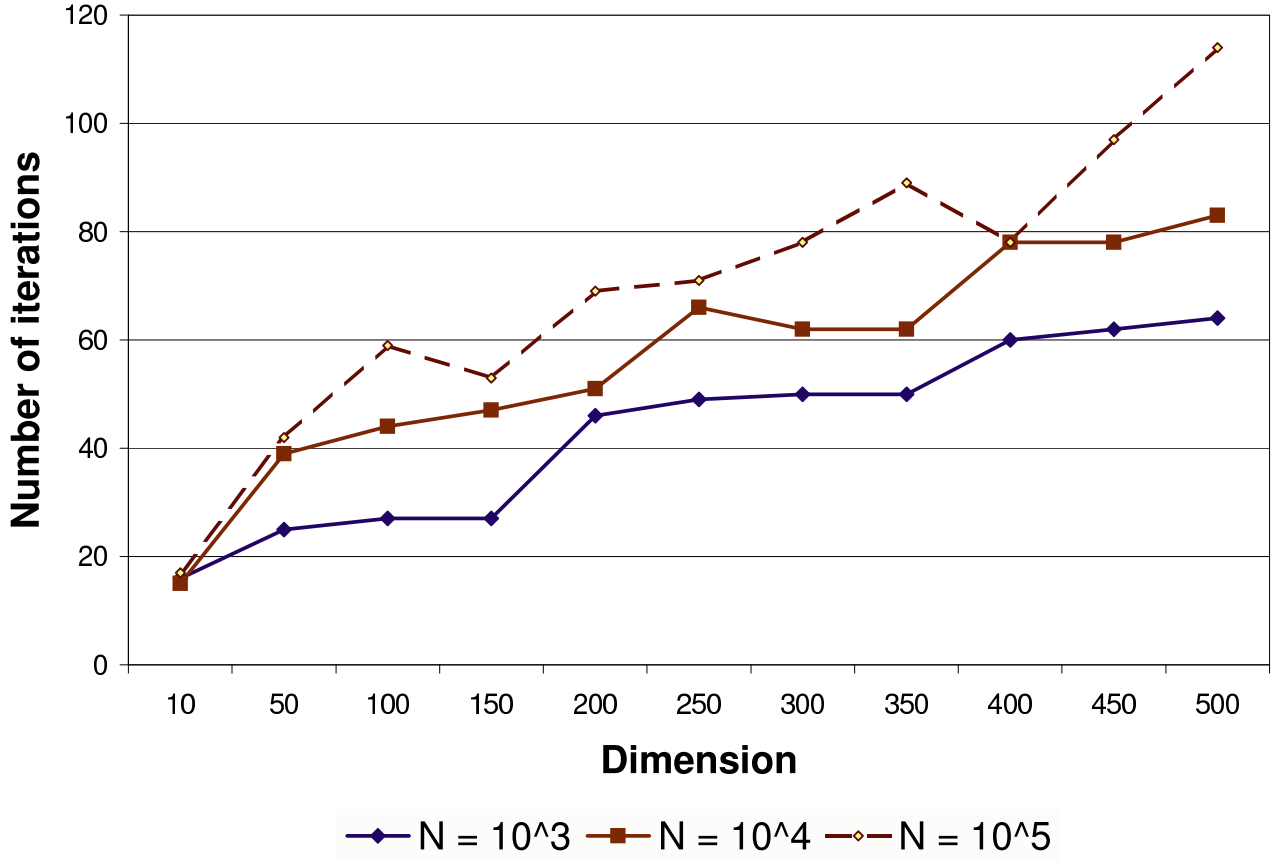
Implementation and Experiments



Running time of algorithm 1

$\epsilon = 0.001, \mu = 0, \sigma = 1.$

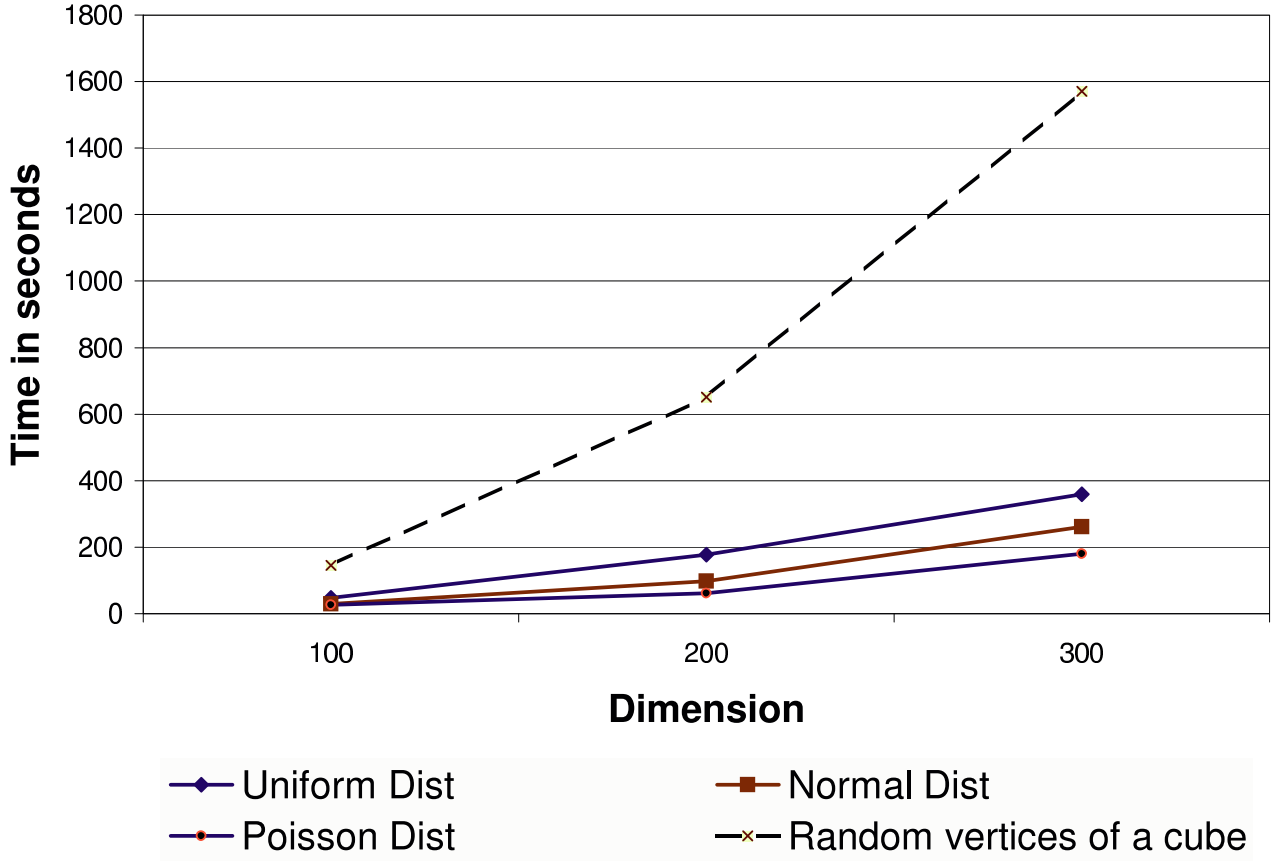
Implementation and Experiments



Core Set Sizes

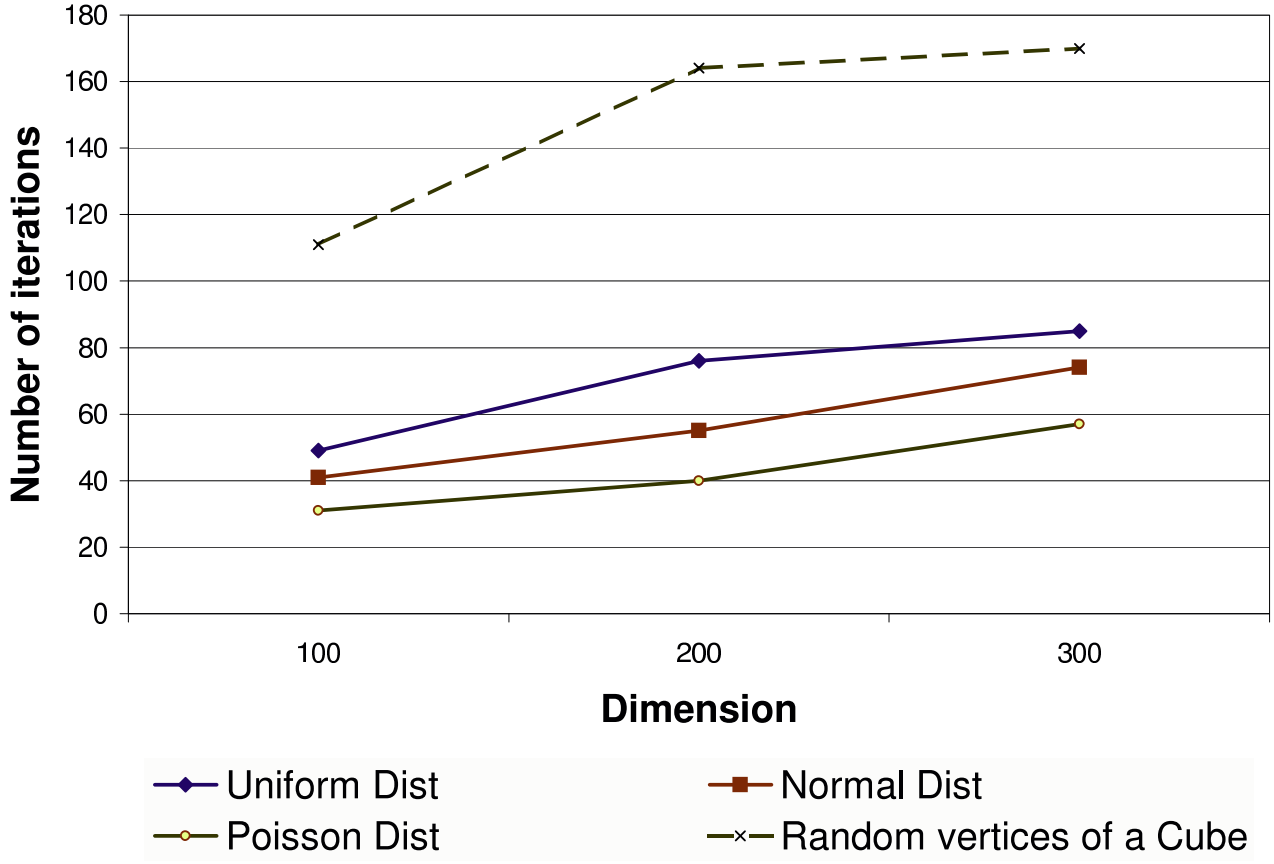
$\epsilon = 0.001, \mu = 0, \sigma = 1$

Implementation and Experiments



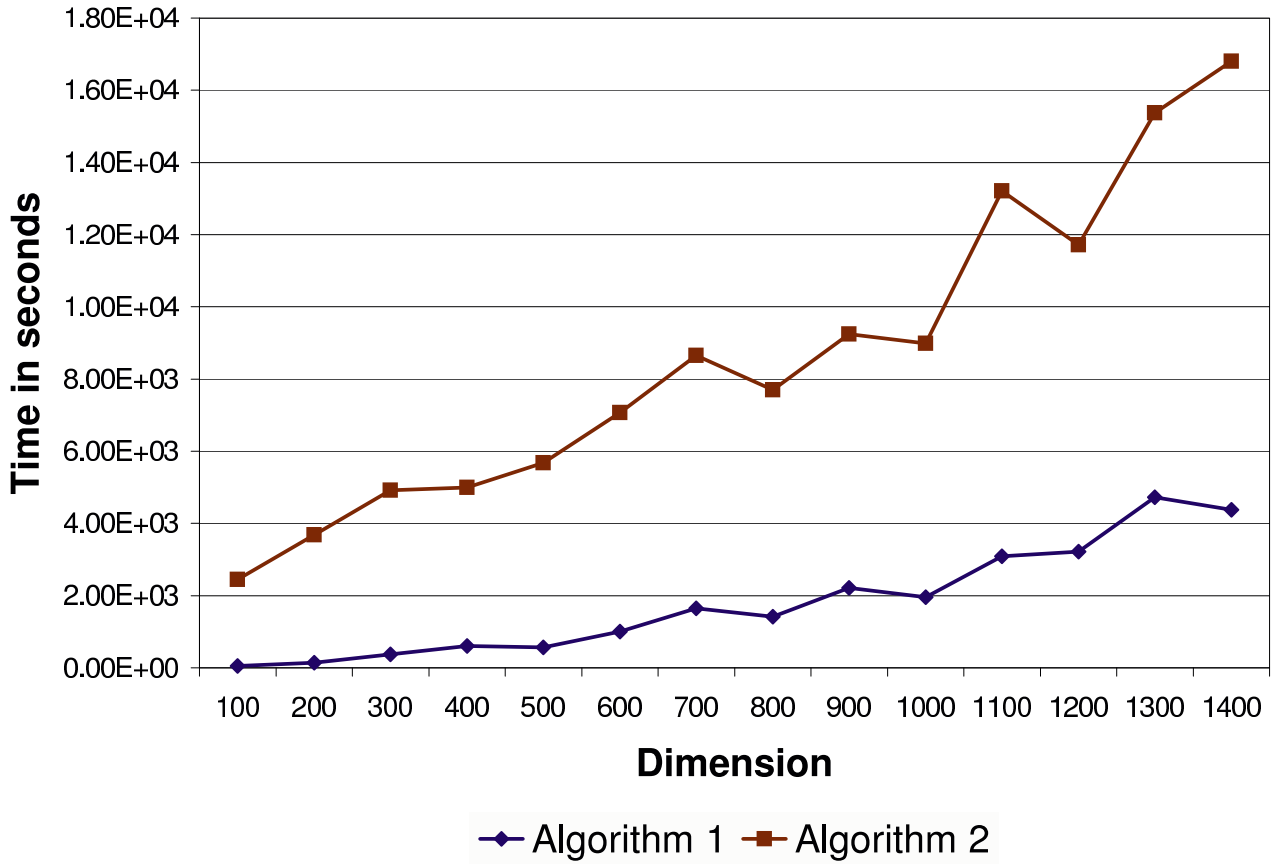
Different Distributions $n = 10000$

Implementation and Experiments



Different Distributions $n = 10000$

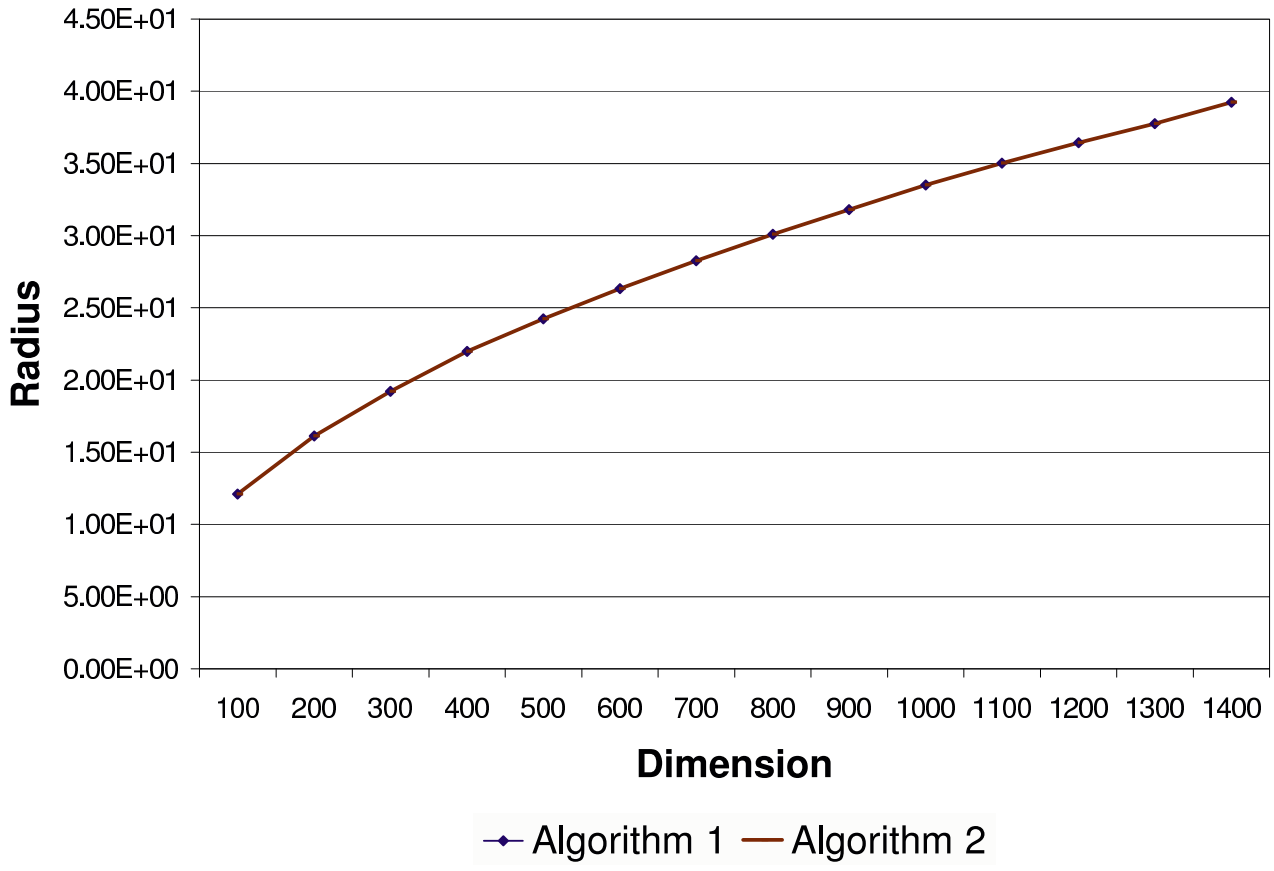
Implementation and Experiments



Timing Comparison (Algorithm 1,2)

$n = 1000, \epsilon = 2^{-10}, \mu = 0, \sigma = 1$

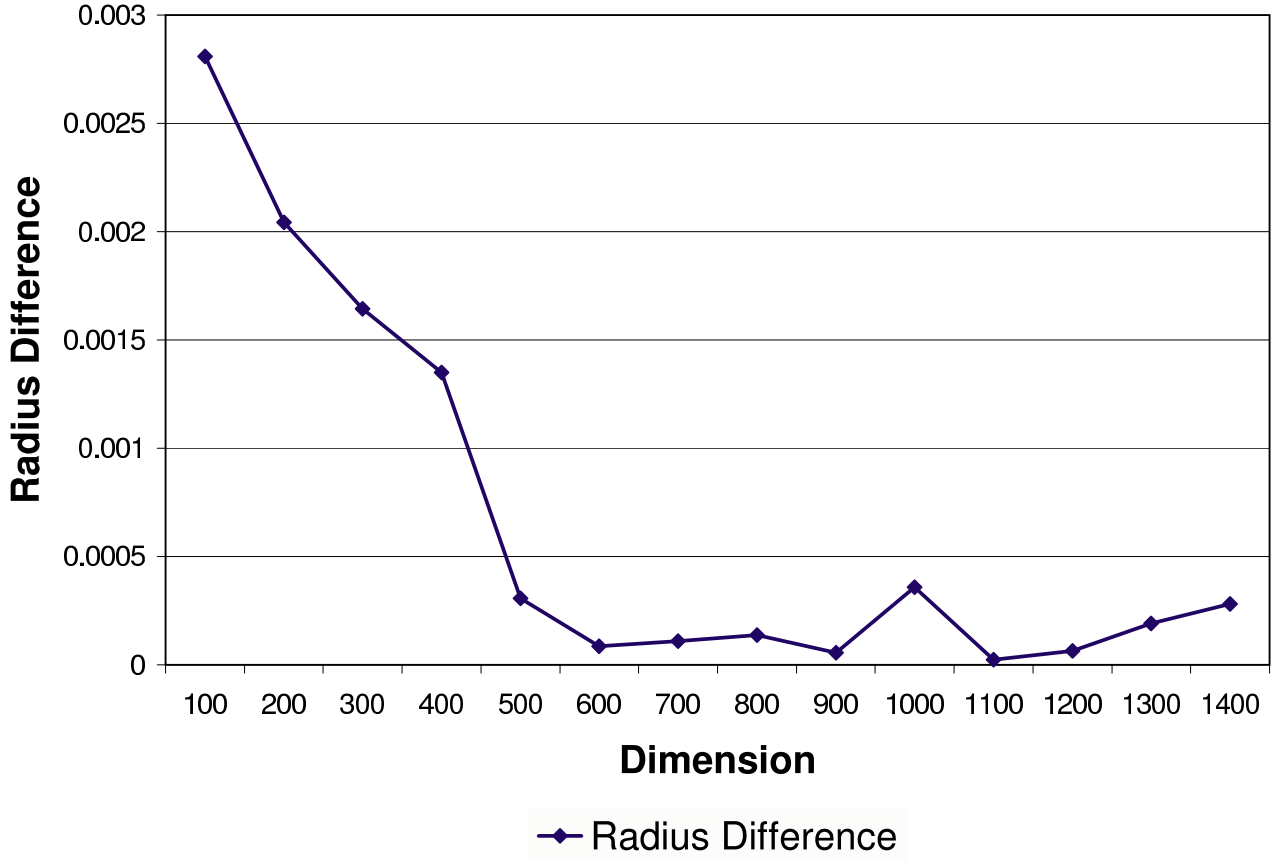
Implementation and Experiments



Radius Comparison

$$n = 1000, \epsilon = 2^{-10}, \mu = 0, \sigma = 1$$

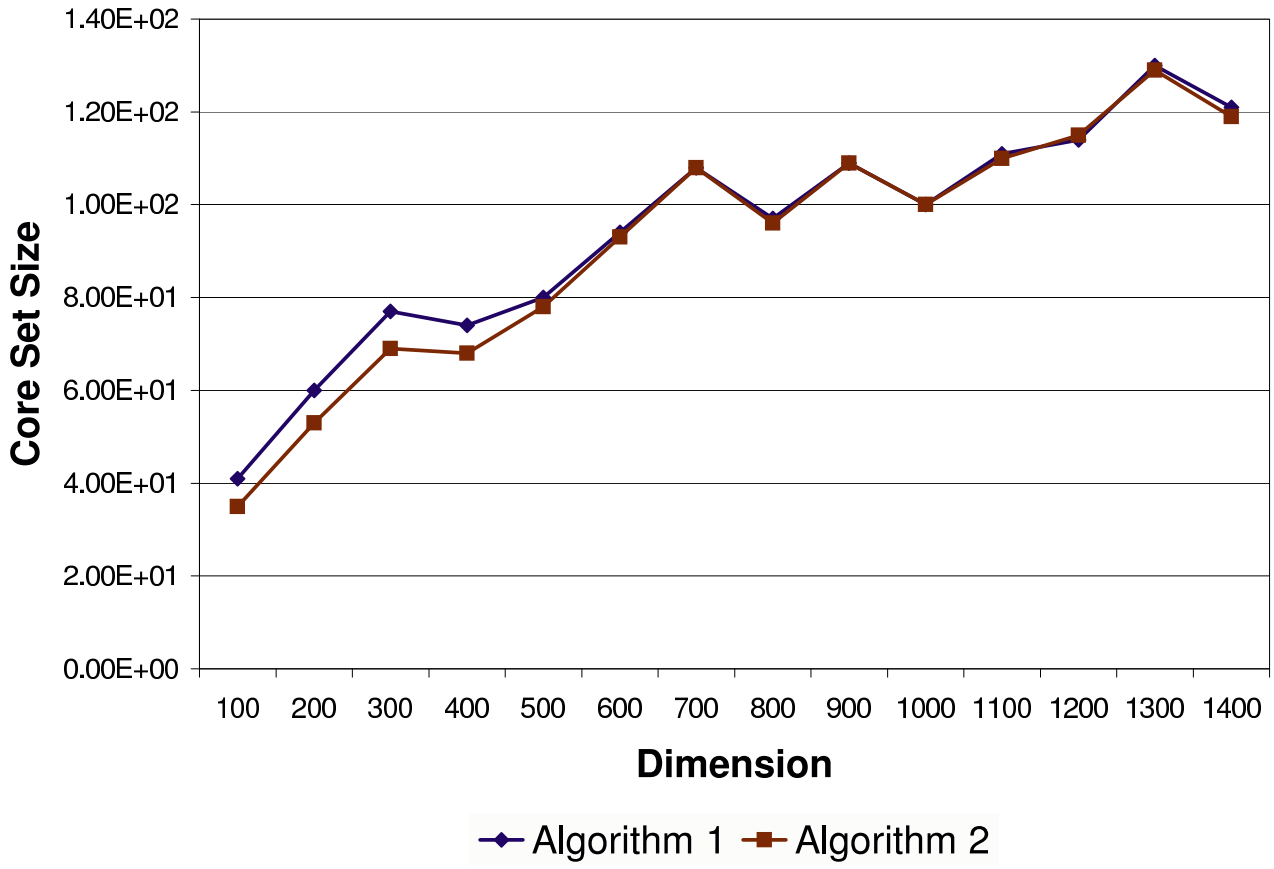
Implementation and Experiments



Radius Difference

$$n = 1000, \epsilon = 2^{-10}, \mu = 0, \sigma = 1$$

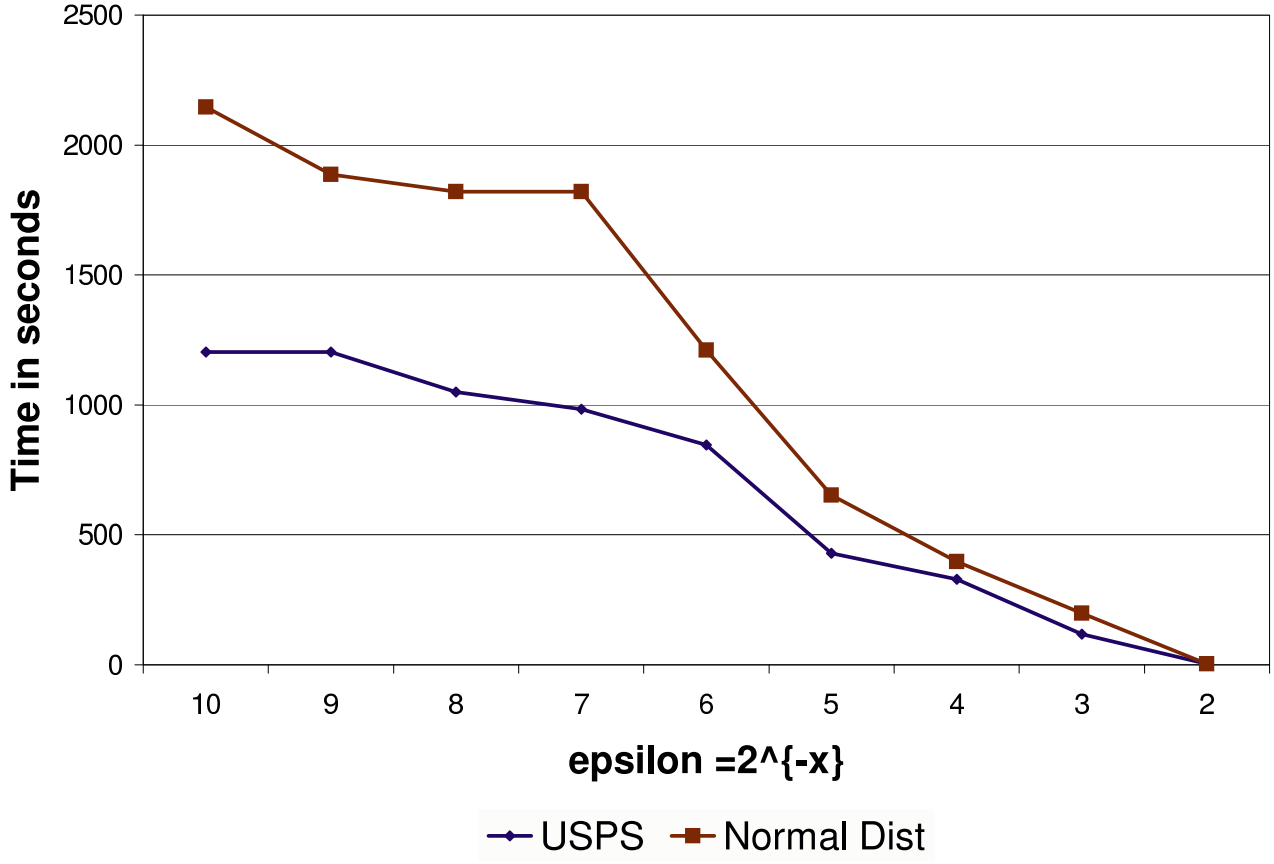
Implementation and Experiments



Core Set Size Comparison

$n = 1000, \epsilon = 2^{-10}, \mu = 0, \sigma = 1$

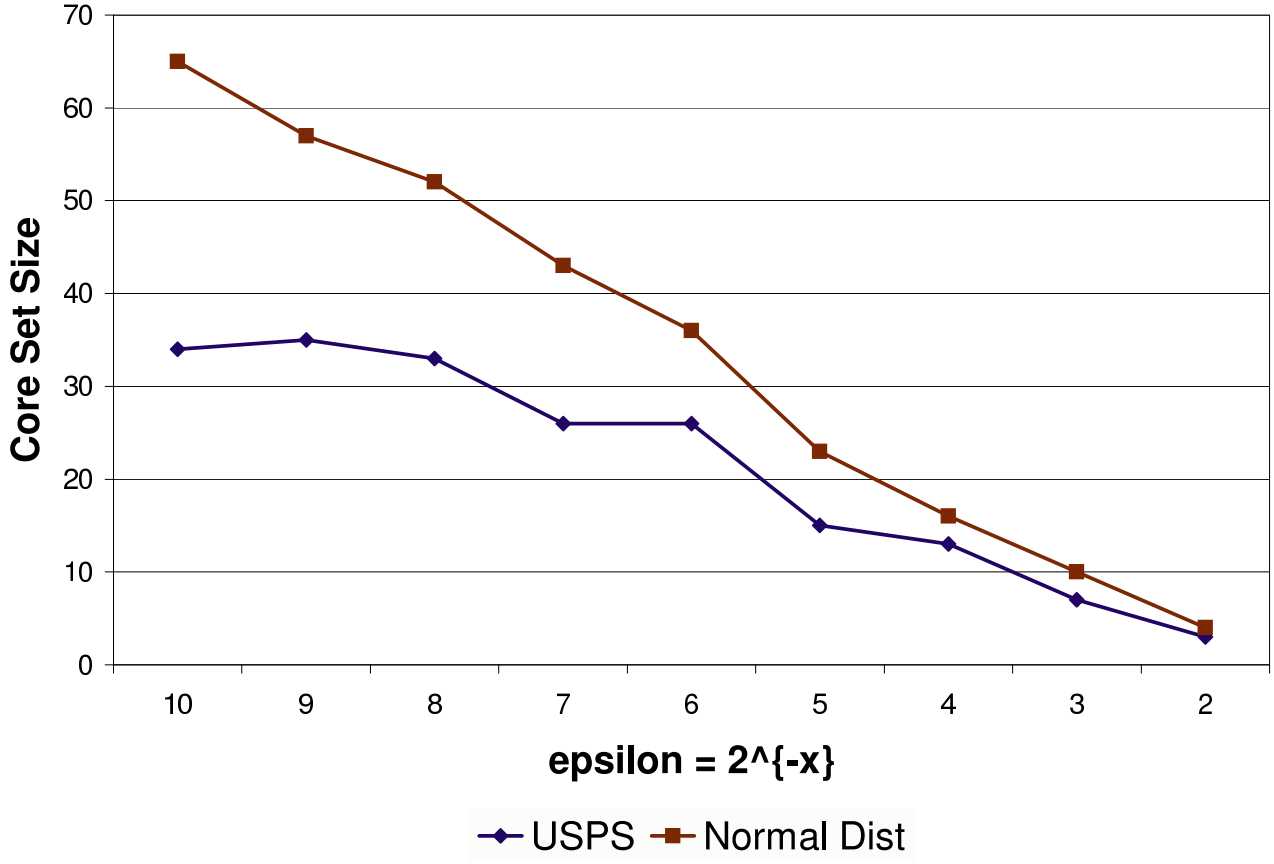
Implementation and Experiments



USPS vs. Normal Data

$n = 7291, \mu = 0, \sigma = 1, d = 256$

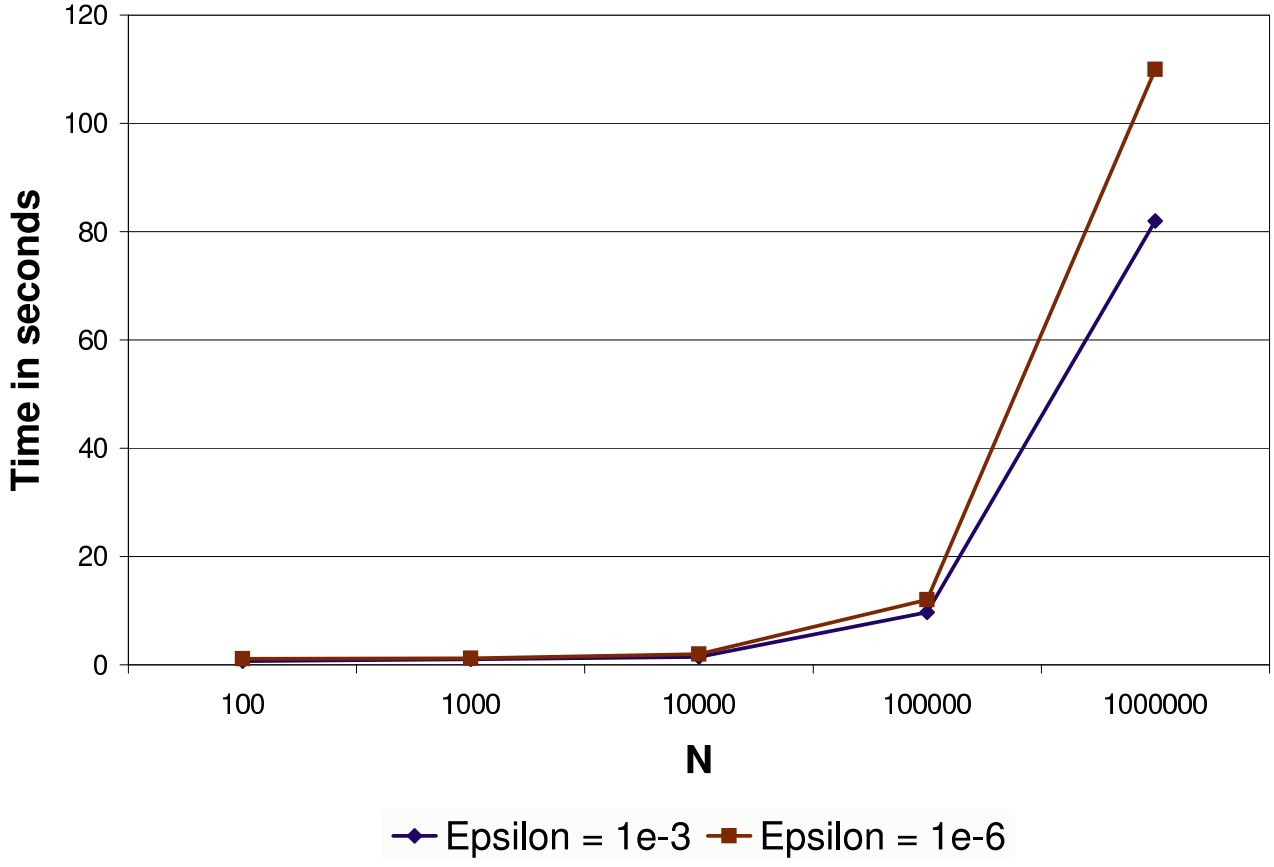
Implementation and Experiments



USPS vs. Normal Data

$n = 7291, \mu = 0, \sigma = 1, d = 256$

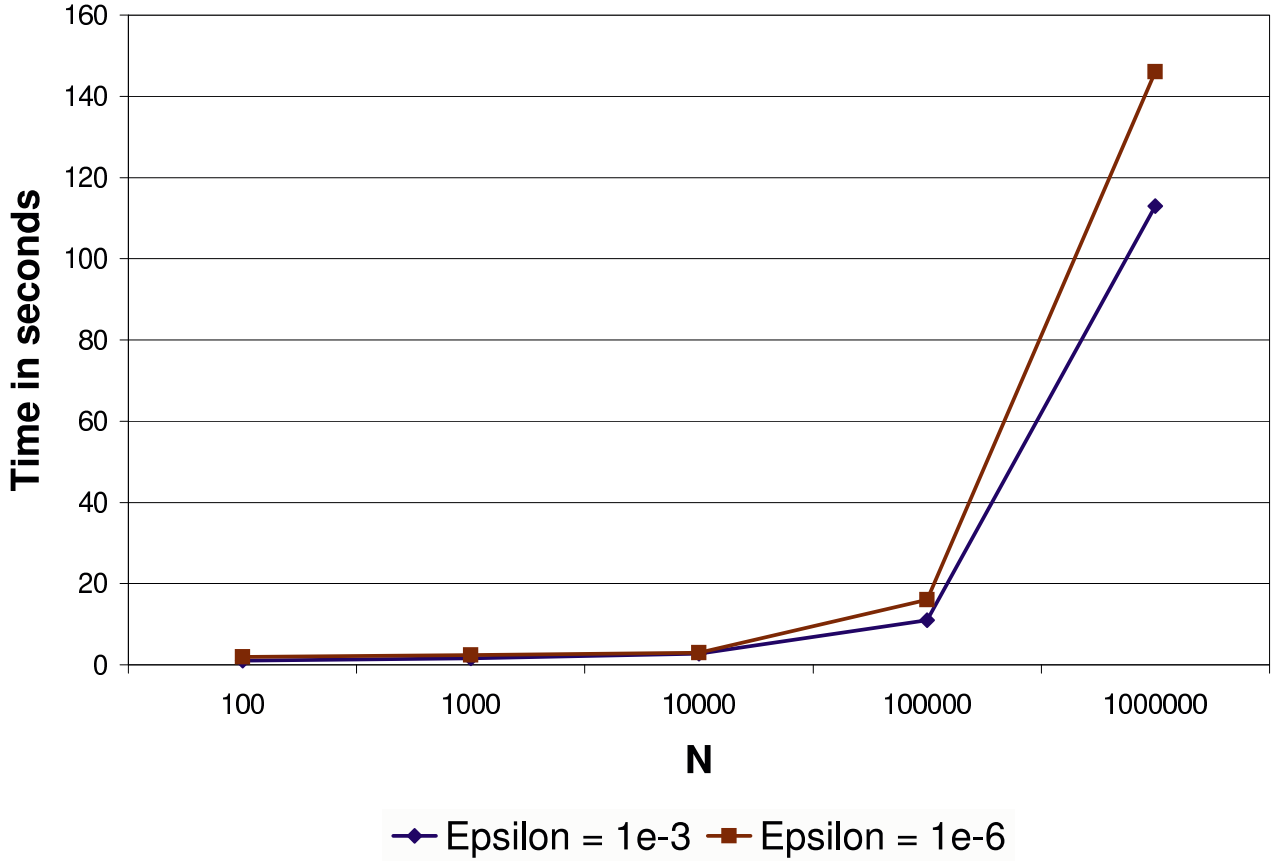
Implementation and Experiments



Experiments in $\mathbb{R}^2$

$\mu = 0, \sigma = 1$

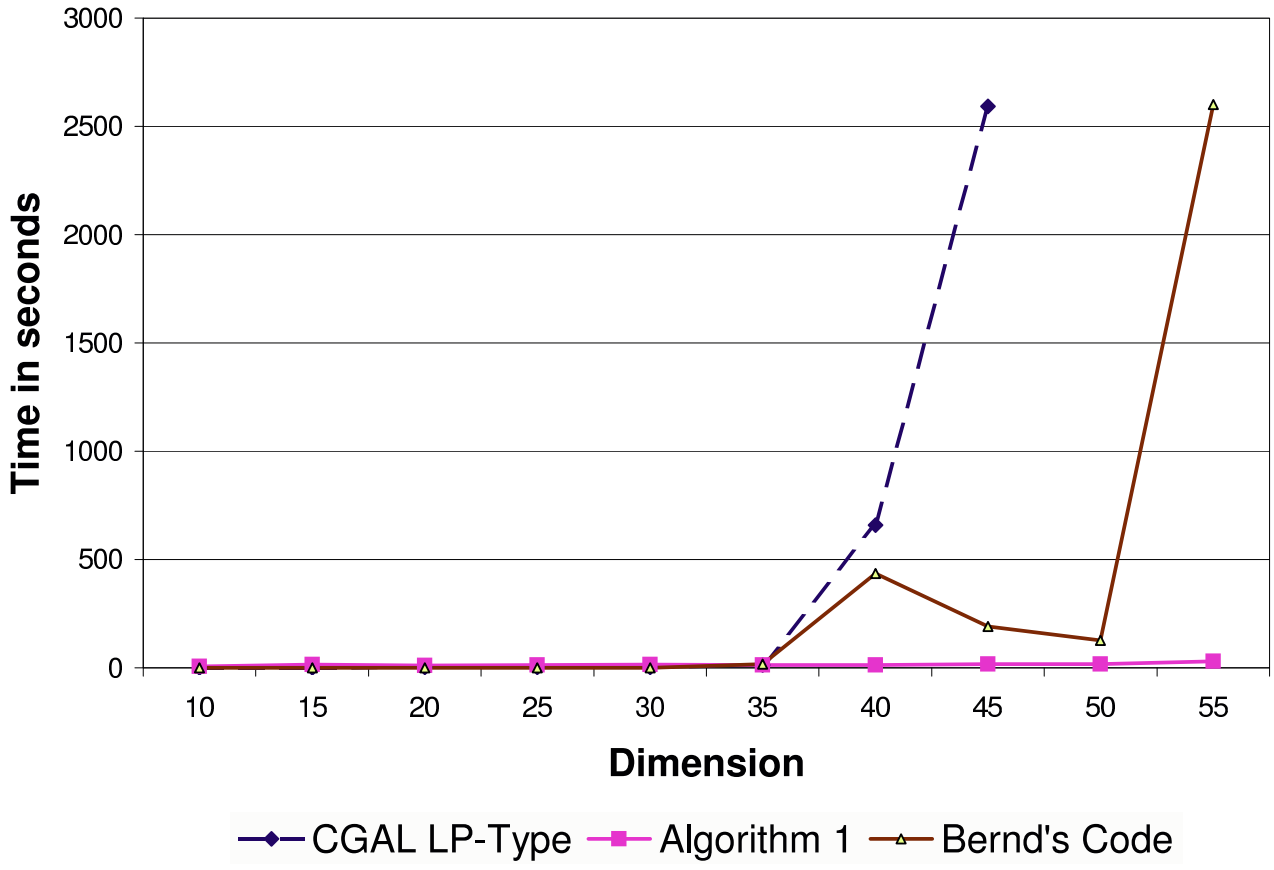
Implementation and Experiments



Experiments in $\mathbb{R}^3$

$\mu = 0, \sigma = 1$

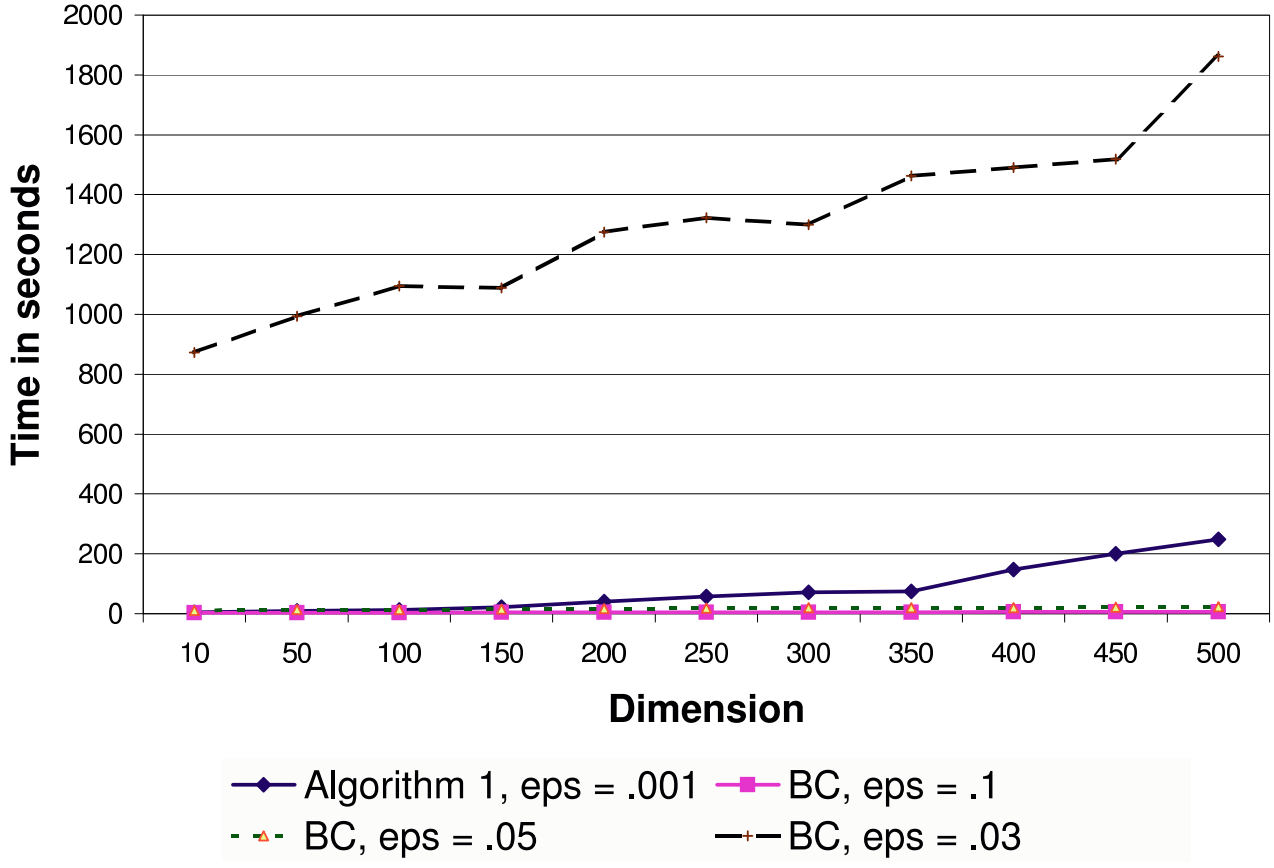
Shameless Promotion ☺



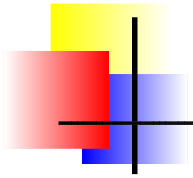
Timing Comparison

$$n = 1000, \epsilon = 10^{-6}, \mu = 0, \sigma = 1$$

Shameless Promotion ☺

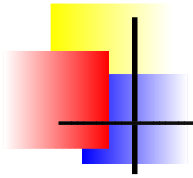


Algorithm Comparison $\mu = 0, \sigma = 1, n = 1000$



Open Problems

- In Practice :
  - Outliers?
  - 1-cylinder? k-center?
  - Minimum Volume Ellipsoids?
  - Warm Start?
  - $\mathcal{O}\left(\frac{nd}{\epsilon} + \frac{1}{\epsilon^4}\log^2\frac{1}{\epsilon}\right)$ Algorithm?



Open Problems

- In Theory :
  - Optimal Core Set Size?
  - Dimension independent core sets for other LP-Type problems?
  - Tight Core Sets for various Distributions?
  - MVEs: Core Sets smaller than $\Theta(d^2)$?

References

[1] F. Alizadeh and D. Goldfarb. Second-order Cone Programming. *Technical Report* RRR 51, Rutgers University, Piscataway, NJ 08854. 2001

[2] N. Alon, S. Dar, M. Parnas and D. Ron. Testing of clustering. In *Proc. 41st Annual Symposium on Foundations of Computer Science*, pages 240–250. IEEE Computer Society Press, Los Alamitos, CA, 2000.

[3] Y. Bulatov, S. Jambawalikar, P. Kumar and S. Sethia. Hand recognition using geometric classifiers. Manuscript.

[4] A. Ben-Hur, D. Horn, H. T. Siegelmann and V. Vapnik. Support vector clustering. In *Journal of Machine Learning. revised version Jan 2002*, 2002.

[5] M. Bădoiu, S. Har-Peled and P. Indyk. Approximate clustering via core-sets. *Proceedings of 34th Annual ACM Symposium on Theory of Computing*, pages 250–257, 2002.

[6] M. Bădoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of 14th ACM-SIAM Symposium on Discrete Algorithms*, to appear, 2003.

[7] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. Manuscript.

[8] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[9] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. In *Proceedings of 16th Annual ACM Symposium on Computational Geometry*, pages 300–309, 2000.

[10] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1/3):131, 2002.

[11] O. Eğecioğlu and B. Kalantari. Approximating the diameter of a set of points in the euclidean space. *Information Processing Letters, 32:205-211*, 1989.

[12] M. Frigo, C. E. Lieserson, H. Prokop and S. Ramachandran. Cache Oblivious Algorithms. *Proceedings of 40th Annual Symposium on Foundations of Computer Science*, 1999.

[13] D. J. Elzinga and D. W. Hearn. The minimum covering sphere problem. *Magangement Science*, 19(1):96–104, Sept. 1972.

[14] K. Fischer. Smallest enclosing ball of balls. Diploma thesis, Institute of Theoretical Computer Science, ETH Zurich, 2001.

[15] B. Gärtner. Fast and robust smallest enclosing balls[1]. In *Proceedings of 7th Annual European Symposium on Algorithms (ESA)*. Springer-Verlag, 1999.

[16] B. Gärtner and S. Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Proceedings of 16th Annual ACM Symposium on Computational Geometry*, pages 110–118, 2000.

[17] A. Goel, P. Indyk and K. R. Varadarajan. Reductions among high dimensional proximity problems. In *Poceedings of 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 769–778, 2001.

[18] M. Grötschel and L. Lovász and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Algorithms and Combinatorics, Springer-Verlag, Vol 2, 1988.

[19] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, July 1996.

[20] W. Johnson and J. Lindenstrauss Extensions of Lipschitz maps into a Hilbert space. *Contemp. Math.* 26, pages 189–206, 1984.

[1] `http://www.inf.ethz.ch/personal/gaertner`

[21] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 248:193–228, 1998.

[22] J. Matoušek, Micha Sharir and Emo Welzl. A subexponential bound for linear programming. In *Proceedings of 8th Annual ACM Symposium on Computational Geometry*, pages 1–8, 1992.

[23] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming*. SIAM Publications, Philadelphia, 1994.

[24] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[25] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8:324–362, 1998.

[26] M. Pellegrini. Randomized combinatorial algorithms for linear programming when the dimension is moderately high. In *Proceedings of 13th ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[27] J. Renegar. A Mathematical View of Interior-Point Methods in Convex Optimization. *MPS/SIAM Series on Optimization 3*. SIAM Publications, Philadelphia, 2001.

[28] Sariel Har-Peled. Personal Communications.

[29] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11/12:625–653, 1999.

[30] K. C. Toh, M. J. Todd and R. H. Tütüncü. SDPT3 — a Matlab software package[2] for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[31] R. H. Tütüncü, K. C. Toh and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. Technical report, Cornell University, 2001. To appear in *Mathematical Programming*.

[32] S. Xu, R. Freund and J. Sun. Solution methodologies for the smallest enclosing circle problem. Technical report, Singapore-MIT Alliance, National University of Singapore, Singapore, 2001.

[33] E. A. Yıldırım and S. J. Wright Warm-Start Strategies in Interior-Point Methods for Linear Programming. *SIAM Journal on Optimization* 12/3, pages 782–810.

[34] G. Zhou, J. Sun and K.-C. Toh. Efficient algorithms for the smallest enclosing ball problem in high dimensional space. Technical report, 2002. To appear in Procedings of Fields Institute of Mathematics.

[2] `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`