Research Statement

Donald Sheehy

April 20, 2010

Overview

Many "well-behaved" mathematical objects are not so well-behaved in a computer system. A smooth function on a space may have nice mathematical structure but that does not mean we can represent it as bits. Many problems in robotics, machine learning, and data analysis are similar in that the underlying object may be complex. Moreover, the inputs that stand as proxy for these objects are clouds of points in a geometric space. I work on algorithms and data structures for geometric problems on point clouds. My work strives to discover, describe, represent, and search this implicit structure.

I mainly focus on methods to construct low-complexity simplicial complexes that accurately encode information about the input point set, underlying sample distribution, the ambient space, or some unknown function on the ambient space. My goal is to make sense of the complex, competing tradeoffs between the geometric, combinatorial, and topological information in important computational problems to produce useful, new algorithms.

Low-Complexity Complexes

Throughout this research statement, let P be a set of n points in d-dimensional Euclidean space. A funny thing happens when we build a simplicial complex P. The number of simplices can vary quite drastically with n, ranging from $O(n)$ to $O(n^{d/2})$.[1] The size of the complex can dominate the number of points as the main parameter for describing the time and space complexity of algorithms on the point set. This is true even for "nice" complexes like the Delaunay triangulation. A problem instance with more points can have significantly reduced complexity. The dimension does not have to get very large before this is the difference between tractable and intractable.

In addition to the combinatorial blowup in the number of simplices, one must also contend with the geometric blowup in the output size of complexes that faithfully represent distances. In such complexes, the Voronoi cells of the vertices have bounded aspect ratio and the dual Delaunay complex is useful for finite element simulation. Tight upper and lower

[1]The big-oh suppresses constants that depend only on d.

bounds indicate that the number of points required to give P this property is $O(n \log \Delta)$, where Δ is the spread of P, i.e. the ratio between the largest and smallest interpoint distances. In general $\log \Delta$ could be unbounded by any fixed function of n.

These two problems may be referred to as the curse of dimensionality and the curse of the spread. Often one can tradeoff between geometric and combinatorial and geometric blowup, playing one curse off another. This was the case in my work on size-competitive no-large-angle triangulations [6]. Later, we were able to show a very general condition for which standard Delaunay-refinement meshing algorithms will produce meshes of linear size [7]. This was then extended to show that n well-spaced points on a surface can be extended to volume mesh with only $O(n)$ simplices [5], settling a long-standing conjecture in the meshing literature.

I have also worked on problems related to cartographic morphing which similarly balanced geometric and combinatorial concerns [2, 1].

The Distance Function and Persistent Homology

A natural way to understand a point cloud is to look at the distance function it induces on the space:

$$d_P(x) = \min_{p \in P} |x - p|.$$

The nested family of sets $\{d_P^{-1}[0, \alpha]\}_\alpha$ is known as the offsets filtration of the point set. Recently, I have been working on applying ideas from mesh generation to the problem of approximating the distance function. My coauthors and I were able to show that the persistent homology of the offsets filtration of any point set can be approximated to an arbitrarily small error using a complex of linear size [4]. This improves the worst case complexity to compute the full filtration for geometric persistent homology from $n^{O(d)}$ to $O(n \log \Delta)$.

Geometric Search

Closely related to the distance function is the problem of proximity search. Many applications on point sets require data structures that allow for fast searching. In fact, point location costs can dominate efficient algorithms for building good simplicial complexes.

Simple data structures are known that return constant-factor approximate nearest neighbors in $O(\log n)$ time. In past work, my coauthors and I developed a method for augmenting these structures to achieve efficient finger-search query times [3]. This means that the running time of a query will only depend logarithmically on the number of points near consecutive queries, which could be significantly smaller than n.

Recently, my colleagues at CMU and I have been looking at methods to build fast point location on top of low-complexity complexes. The aim here is to build a complex which gives a good approximation to the Voronoi diagram but does not incur the worst case $n^{O(d)}$ complexity blowup. Other methods have been proposed to approximate Voronoi diagrams with quadtrees. Our approach is unique in that it uses Voronoi diagrams of a superset. This means that in many cases where the input would not exhibit the worst case behavior, we will return the true Voronoi diagram and thus give *exact* nearest neighbors [9].

Data Depth and Centerpoints

Data depth gives an alternative to distance functions as a way of describing the underlying shape of a point sample. Several notions of depth are studied but the most popular is the Tukey depth, defined as

$$D_\pi(x) = \min_{h \in H : x \in h} |P \cap h|,$$

where H is the set of all halfspaces. The Centerpoint Theorem states that there exists a point of depth at least $\frac{n}{d+1}$. Such a point is called a *centerpoint* and acts as a geometric median and is used as a robust statistic for describing P.

The fastest known deterministic algorithm for computing a centerpoint requires $O(n^{d-1})$ time. I was able to show that if one is willing to accept a point of depth $\frac{n}{4d^2}$, the running time can be reduced to $n^{O(\log d)}$ by combining several tools from classical convexity theory [10].

Wedge depth is a natural generalization of Tukey depth which replaces halfspaces with *wedges*, where an α-wedge $W(x,r)$ is defined as

$$W(x,r) = \left\{ y : \angle rxy \leq \frac{\alpha}{2} \right\},$$

for any two distinct points x and r. Let W_x be the set of all α-wedges of the form $W(x,r)$. The α-wedge depth can now be defined as

$$D_\alpha(x) = \min_{w \in W_x : x \in w} |P \cap w|.$$

It is not hard to see from the two definitions that Tukey depth is a special case of wedge depth for $\alpha = \pi$ (thus explaining the notation).

My work on the Centervertex Theorem showed that choosing $\alpha = \frac{3\pi}{2}$ guarantees the existence of a point x with $D_\alpha(x) \geq \frac{n}{d+1}$ *among the points of* P [8]. That is, we can find a deep point among the input set. This is closer to our notion of a median in 1D, where we return a point among the inputs. This was not the case for centerpoints, which generally do not reside inside the set P.

Ongoing and Future work

The main focus of my thesis work is on extending the mesh-based approaches to problems that approximate the distance function on a space, particularly geometric persistent homology. Some problems in this area that I am working on include increasing noise tolerance, combining persistence with statistical approaches, using the point location in the mesh generator to bootstrap the persistence algorithm, and other methods to reduce the complexity of the filtrations used.

References

[1] J. Danciger, S. Devadoss, J. Mugno, D. Sheehy, and R. Ward. Shape deformation in continuous map generalization. *GeoInformatica*, 13(2):203–221, 2009.

[2] J. Danciger, S. Devadoss, and D. Sheehy. Compatible triangulations and point partitions by series triangular graphs. *Computational Geometry: Theory and Applications*, 34:195–202, 2006.

[3] J. Derryberry, D. D. Sleator, D. Sheehy, and M. Woo. Achieving spatial adaptivity while finding approximate nearest neighbors. In *CCCG: Canadian Conference in Computational Geometry*, 2008.

[4] B. Hudson, G. L. Miller, S. Y. Oudot, and D. R. Sheehy. Topological inference via meshing. In *SOCG: Proceedings of the 26th ACM Symposium on Computational Geometry*, 2010.

[5] B. Hudson, G. L. Miller, T. Phillips, and D. Sheehy. Size complexity of volume meshes vs. surface meshes. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[6] G. L. Miller, T. Phillips, and D. Sheehy. Size competitive meshing without large angles. In *34th International Colloquium on Automata, Languages and Programming*, 2007.

[7] G. L. Miller, T. Phillips, and D. Sheehy. Linear-size meshes. In *CCCG: Canadian Conference in Computational Geometry*, 2008.

[8] G. L. Miller, T. Phillips, and D. Sheehy. The centervertex theorem for wedge depth. In *CCCG: Canadian Conference in Computational Geometry*, 2009.

[9] G. L. Miller, T. Phillips, and D. R. Sheehy. Approximating voronoi diagrams with voronoi diagrams. In *The Fall Workshop in Computational Geometry*, 2009.

[10] G. L. Miller and D. Sheehy. Approximate center points with proofs. In *SOCG: Proceedings of the 25th ACM Symposium on Computational Geometry*, 2009.