Abstract

We present a data structure for *spherical range reporting* on a point set S, i.e., reporting all points in S that lie within radius r of a given query point q (with a small probability of error). Our solution builds upon the Locality-Sensitive Hashing (LSH) framework of Indyk and Motwani, which represents the asymptotically best solutions to near neighbor problems in high dimensions. While traditional LSH data structures have several parameters whose optimal values depend on the distance distribution from q to the points of S (and in particular on the number of points to report), our data structure is essentially parameter-free and only takes as parameter the space the user is willing to allocate. Nevertheless, its expected query time basically matches that of an LSH data structure whose parameters have been *optimally chosen for the data and query* in question under the given space constraints. In particular, our data structure provides a smooth trade-off between hard queries (typically addressed by standard LSH parameter settings) and easy queries such as those where the number of points to report is a constant fraction of S, or where almost all points in S are far away from the query point. In contrast, known data structures fix LSH parameters based on certain parameters of the input alone. The algorithm has expected query time bounded by $O(t(n/t)^\rho)$, where t is the number of points to report and $\rho \in (0,1)$ depends on the data distribution and the strength of the LSH family used. The previously best running time in high dimensions was $\Omega(tn^\rho)$, achieved by traditional LSH-based data structures where parameters are tuned for outputting a single point within distance r. Further, for many data distributions where the intrinsic dimensionality of the point set close to q is low, we can give improved upper bounds on the expected query time. We finally present a parameter-free way of using multi-probing, for LSH families that support it, and show that for many such families this approach allows us to get expected query time close to $O(n^\rho + t)$, which is the best we can hope to achieve using LSH.