CSE 347 Analysis of Algorithms April 22, 2019

Lecture 24 Summary

Instructor: Brendan Juba

1. *Clustering* is a task of grouping points together so that points in the same group are relatively close to each other, e.g., in Euclidean distance. *Lloyd's algorithm* is an algorithm that is widely used for this task in practice.

   (a) There are many objective functions used for clustering. A common objective is *k-means*: minimize the sum of squared Euclidean distances of each point from the center of mass (average) of the cluster it is assigned to.

   (b) Lloyd's algorithm alternates between updating the centers of mass by taking the average of the current cluster assignments and reassigning points to the (new) nearest center, until the assignment stabilizes.

   (c) The k-means objective decreases on every iteration, so Lloyd's algorithm must terminate. Actually, this difference can be quantified by how much the cluster center moves (squared), scaled by the size of the cluster.

   (d) Note: Lloyd's algorithm is *not* guaranteed to minimize the k-means objective. It may "get stuck" in a bad local minimum.

   (e) Also note: although Lloyd's algorithm is widely used in practice due to its speed, there are examples on which it requires an exponential number of iterations to terminate.

2. *Smoothed analysis* measures the running time of *noisy* worst-case inputs.

   (a) The *smoothed running time* of an algorithm with *noise parameter* σ is a random variable given by

      i. Taking a worst-case input matrix X of a given size

      ii. Taking a matrix of mean-0 Gaussian random variables with variance σ^2 G of the same dimension as X

      iii. Computing the running time of the algorithm on $X + G$.

   (b) We are often interested in the expected value of this running time (or we may be interested in high-probability bounds)

   (c) Measurement error and natural variation both frequently are observed to have a Gaussian distribution in practice, so unlike average-case analysis (which assumed more detailed knowledge of the distribution of inputs than is plausible in most cases), smoothed analysis is a reasonable measure of running time in many natural situations. At the same time, it discounts pathological cases that rely on implausibly delicate input configurations.

   (d) Precisely: in n dimensions, the probability that a smoothed value is withing ϵ of any fixed target vector y is at most $(\epsilon/\sigma)^n$.

3. In high dimensions, Lloyd's algorithm has polynomial smoothed running time. (Actually, this is true for all dimensions, but we didn't show this here.)

(a) A union bound gives us that the centers of all possible distinct subsets of points cannot be too close to one another.

(b) Therefore, when the clusters change, we have a (polynomial) lower bound on how much the k-means objective must decrease by.

(c) The diameter of the point set gives us a coarse polynomial upper bound on the initial k-means objective. But this is enough: dividing the polynomial upper bound by the polynomial lower bound gives us a polynomial upper bound on the number of steps it would take to reduce the difference to (below) zero. Since the objective is always nonnegative, the algorithm must converge within that many iterations.

4. Smoothed analysis has also been used to show that the Simplex algorithm for linear programming runs in smoothed polynomial time. This helps explain why the Simplex algorithm is so successful in practice in spite of its exponential worst-case complexity.