Ph.D. Open

Vincent Cohen-Addad

Provable algorithms for data mining and unsupervised machine learning

Grading: Questions numbered 1 are worth 1 each, 2 are worth 2 points each, 3 and 4 are worth 3 points each.

On the k-Center Problem

We recall the k-center problem. Let (X, d) be a metric space (e.g.: X could be $X \subset \mathbb{R}^2$ and d be the ℓ_2-distance), and k be an integer. The goal is to find a set C of k points in X, called centers, so as to minimize $\max_{p \in X} \min_{c \in C} d(p, c)$.

Exercise 1

1. Provide a polynomial time algorithm that solves k-center exactly in $X \subset \mathbb{R}$, where d is the ℓ_1 distance. Namely an algorithm whose running time is polynomial in n, k.

2. Provide an algorithm with running time $O(n \log(\Delta) \log n$ that solves k-center exactly in $X \subset \mathbb{R}$, where d is the ℓ_1 distance, where Δ is the ratio of the maximum distance between input point to the minimum distance between input points.

We recall the notion of ε-coreset for k-center. An ε-coreset of an instance $(X, d), k$ of k-center is a subset of points X' such that the value of the optimum k-center solution on instance $(X', d), k$ is wipthin a $(1+\varepsilon)$ factor of the value of the optimum k-center solution on instance $(X, d), k$.

Exercise 2

1. Show that if the input instance $(X, d), k$ is arbitrary, namely that (X, d) is an arbitrary finite metric space, then there is no ε-coreset of size $o(n)$ for $\varepsilon < 1$.

2. Provide an ε-coreset of size $O(k/\varepsilon)$ for k-center where $X \subset \mathbb{R}$, where d is the ℓ_1 distance.

3. Provide an ε-coreset of size $O(k/\varepsilon^\delta)$ for k-center where $X \subset \mathbb{R}^\delta$, where d is the ℓ_2 distance.
4. Provide a $(1+\varepsilon)$-approximation algorithm for k-center with running time $O((k/\varepsilon)^{\delta k/\varepsilon^\delta} + nk^2/\varepsilon^\delta)$ where $X \subset \mathbb{R}^\delta$.

On Approximate Nearest Neighbors

We recall the definition of the γ-Approximate Nearest Neighbor problem we saw in class. Given a set $X \subset \mathbb{R}^\delta$, a parameter σ, our goal is to build a data structure that on an input query point q, outputs an element $p \in X$ at distance at most σ from p if there is one; or outputs that there is no element of X at distance less than $\gamma\sigma$ from p if there is none; otherwise the data structure may answer arbitrarily. We work with the ℓ_2 distance. The query time refers to the worst-case time the data structure takes to answer a query. Let Δ be the ratio of the maximum distance between input point to the minimum distance between input points.

We would like to build a γ-ANN data structure for $\mathbb{R}^\delta$.

Exercise 3

1. Provide an exact ($\gamma = 1$) deterministic data structure for $\mathbb{R}$ with query time $O(\log n)$.

2. For any ε, provide a randomized data structure for $\mathbb{R}^\delta$ and $\gamma = (1+\varepsilon)$ with query time $O((\varepsilon^{-1}\log n)^\delta)$, and success probability $1 - 1/n$ **Assume delta = 2**

3. For any ε, provide a randomized data structure for $\mathbb{R}^\delta$ and $\gamma = (1+\varepsilon)$ with query time $O((\varepsilon^{-1}\log n)^\delta)$, and success probability $1 - 1/n$

4. Provide a randomized data structure for $\mathbb{R}^\delta$ and $\gamma = O(\delta)$ with query time $O(\delta \log n \log(1/\rho))$ and success probability $1 - \rho$.