Institut für Informatik
Dr. Melanie Schmidt
Clemens Rösner

Cluster Analysis
Winter 2018/19

# Problem Set 12

Please hand in your solutions for this problem set via email (roesner@cs.uni-bonn.de) or personally at room 2.060 until *Tuesday, 22th of January.*

## Problem 1

Explain how the coreset-construction algorithm from the lecture can be adapted to instances with weighted points.

## Problem 2

Assume that you are given access to an algorithm that, given a set $P \subset \mathbb{R}^d$ and a number $k \in \mathbb{N}^{\geq 1}$, computes a $(k, \epsilon)$-coreset $S$ for $(P, k)$. Let $s$ be the size of the resulting coreset ($s$ may depend on various parameters). Consider the following streaming algorithm.

1. Read $s$ points from the stream into a set $A$.
2. **Repeat**
3. Read $s$ points from the stream into a set $B$. (Or less, at the end of the stream).
4. Compute a $(k, \epsilon)$-coreset for $(A \cup B, k)$ and store it in $A$.
5. **until** end of stream

What is the error of the final coreset?

## Problem 3

Show how to compute the centroid of $n$ points in the streaming model. In the streaming model we assume that we only see each point once and are not allowed to store all the points that we have seen so far. Instead we only have space of size $O(f(n))$ to store data.

- Show how to compute the centroid with space $O(1)$.
- Assume that we have $O(\log(n))$ space, but each computation can induce a numerical error of $\varepsilon$. Show how the final error can be bound by $O(\varepsilon \cdot \log(n))$.

## extra Problem

When the Merge&Reduce technique was explained in the lecture we assumed that we know the number of points that are coming before setting the parameters of the algorithm. How can we get rid of this assumption?