A-001

ℓ-diversity clustering on the line

Toshihiro Akagi* Shin-ichi Nakano*

1 Introduction

Given a set C of n points, a set Q of m colors, the color $col(c) \in Q$ of each $c \in C$, the distance d between each pair of points, and a number ℓ, an ℓ-diversity clustering of C is a partition $\mathcal{P}$ of points into clusters such that each cluster has at least ℓ points all of which have distinct colors. Then we define (the max version of) the cost of an *ℓ-diversity clustering* as $\max_{\mathcal{C}\in\mathcal{P}} \max_{u,v\in\mathcal{C}} d(u,v)$. The min-max version of the ℓ-diversity clustering problem is the problem to find an ℓ-diversity clustering having the minimum cost [3]. Similar but different ℓ-diversity clustering problem is discussed in [2], in which each cluster has points colored by each color at most probability $1/\ell$. Those ℓ-diversity problems have an application for publishing data with protecting the privacy [2, 3]. We can assume $m \geq \ell$, since otherwise it has no solution.

If $|Q| = |C|$ then this is the r-gathering problem[1] with $r = \ell$, and since the r-gathering problem is known to be NP-hard[1], the ℓ-diversity problem is also NP-hard. Li. et al.[3] gave a polynomial-time 2-approximation algorithm for any ℓ. Ghinita. et al.[2] gave an $O(2^m \cdot n^{m+2})$ time algorithm to solve the (similar but different) ℓ-diversity clustering problem when all C are on the line. In this paper we give a faster $O(2^m \cdot (\frac{n}{m}+1)^m)$ time algorithm to solve the ℓ-diversity clustering problem when all C are on the line. When m is a (small) constant, those algorithms runs in polynomial-time.

The remainder of the paper is organized as follows. Section 2 gives an algorithm to solve the ℓ-diversity clustering problem when all C are on the line, based on dynamic programming approach. Finally Section 3 is a conclusion.

2 Our algorithm

In this section we design an algorithm to solve the ℓ-diversity clustering problem when all points are on a horizontal line, based on dynamic programming approach.

Let C^j be the set of points with color j, c_i^j be the i-th point from the left among C^j, and C_i^j be $\{c_1^j, c_2^j, \ldots, c_i^j\}$, which is the leftmost i points in C^j.

The input of the problem is a set C of n points on the horizontal line, a set Q of m colors, the color $col(c) \in Q$ for each $c \in C$, and a number $\ell \leq m$. The distance d between each pair of points is the Euclid distance. Then subproblem $P(i_i, i_2, \ldots, i_m)$ is the problem to find an ℓ-diversity clustering $\mathcal{P}$ of $C_{i_1}^1 \cup C_{i_2}^2 \cup \cdots \cup C_{i_m}^m \subseteq C$ having the minimum cost. Note that the original problem is denoted by $P(|C^1|, |C^2|, \ldots, |C^m|)$. Let $cost(i_1, i_2, \ldots, i_m)$ be the cost of a solution of $P(i_1, i_2, \ldots, i_m)$. Especially if $P(i_1, i_2, \ldots i_m)$ has no solution then we define $cost(i_1, i_2, \ldots, i_m) = \infty$.

We say an ℓ-diversity clustering $\mathcal{P} = (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k)$ of C is *monotone* if $c_1^j, c_2^j, \ldots$ appear in this order in $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ for each $j \in Q$, or more formally, if $c_{i'}^j, c_i^j \in C^j$ with $i' < i$ and, in an ℓ-diversity clustering $\mathcal{P}$, $c_{i'}^j \in \mathcal{C}_{t'}, c_i^j \in \mathcal{C}_t$ and $\mathcal{C}_{t'}, \mathcal{C}_t \in \mathcal{P}$, then $t' \leq t$ holds.

Lemma 1. If $P(i_1, i_2, \ldots, i_m)$ has a solution then it has a monotone solution.

Proof. Assume otherwise for a contradiction. Some $P(i_1, i_2, \ldots, i_m)$ has a solution, but all of them are not monotone. Let $\mathcal{P} = (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m)$ be the solution of $P(i_1, i_2, \ldots, i_m)$ with the minimum number of "reverse points", which is the pair of points $c_{i'}^j$ and c_i^j with $i' < i$ but $t < t'$ hold where $c_{i'}^j \in \mathcal{C}_{t'}, c_i^j \in \mathcal{C}_t$.

Then modify $\mathcal{P}$ by swapping $c_{i'}^j$ and c_i^j, that is remove $c_{i'}^j$ from $\mathcal{C}_{t'}$ and append it to $\mathcal{C}_t$, and remove c_i^j from $\mathcal{C}_t$ and append it to $\mathcal{C}_{t'}$. The resulting partition is again an ℓ-diversity clustering. Also the maximum cost of $\mathcal{C}_{t'}$ and $\mathcal{C}_t$ is decreased, or the number of reverse pair is decreased. A contradiction. □

We have two more lemmas.

Lemma 2. If $P(i_1, i_2, \ldots, i_m)$ has a solution then it has a solution such that each cluster has at most $2\ell - 1$ points.

Proof. Assume some cluster $\mathcal{C}$ in a solution has more than 2ℓ points, then divide the cluster into two clusters. The resulting ℓ-diversity clustering has less cost. A contradiction. □

Lemma 3. (a) If $i_1 + i_2 + \cdots + i_m < \ell$ then $P(i_1, i_2, \ldots, i_m)$ has no solution. (b1) If $\ell \leq i_1 + i_2 + \cdots + i_m \leq 2\ell - 1$ and $0 \leq i_j \leq 1$ for each $j = 1, 2, \ldots, m$ then problem $P(i_1, i_2, \ldots, i_m)$ has a solution consisting of exactly one cluster. (b2) If $\ell \leq i_1 + i_2 + \cdots + i_m \leq 2\ell - 1$ and $2 \leq i_j$ for some j then problem $P(i_1, i_2, \ldots, i_m)$ has no solution. (c) $P(i_1, i_2, \ldots, i_m)$ with $2\ell \leq i_1 + i_2 + \cdots + i_m$ has a solution (consisting of two or more clusters) iff some problem $P(i'_1, i'_2, \ldots, i'_m)$ with $i_1 - 1 \leq i'_1 \leq i_1, i_2 - 1 \leq i'_2 \leq i_2, \cdots, i_m - 1 \leq i'_m \leq i_m$ and $\ell \leq (i_1 - i'_1) + (i_2 - i'_2) + \cdots + (i_m - i'_m) \leq 2\ell - 1$ has a solution.

Proof. (a), (b1) and (b2) are Obvious.
(c:⇒) Assume $P(i_1, i_2, \ldots, i_m)$ has a solution $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$. Then $\mathcal{P}' = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{k-1}\}$ is a ℓ-diversity clustering of some smaller problem $P(i'_1, i'_2, \ldots, i'_m)$ with $i_1 - 1 \leq i'_1 \leq i_1, i_2 - 1 \leq i'_2 \leq i_2, \cdots, i_m - 1 \leq i'_m \leq i_m$ and $\ell \leq (i_1 - i'_1) + (i_2 - i'_2) + \cdots + (i_m - i'_m) \leq 2\ell - 1$.
(c:⇐) By appending an ℓ-diversity clustering of $P(i'_1, i'_2, \ldots, i'_m)$ with one more cluster $\mathcal{C}$ consisting of the set $\{c_i^j \in C \mid i'_j = i_j - 1, j \in [1..m]\}$. We can construct an ℓ-diversity clustering of $P(i_1, i_2, \ldots, i_m)$. □

By Lemma 3, each problem $P(i_1, i_2, \ldots, i_m)$ can be solved by checking all possible at most 2^m of smaller problems $P(i'_1, i'_2, \ldots, i'_m)$ where each i'_j for $j \in Q$ is either i_j or $i_j - 1$, and the $cost(i_1, i_2, \ldots, i_m)$ is the minimum of the

*Department of Computer Science, Gunma University

maximum of $cost(i'_1, i'_2, \ldots, i'_m)$ or the cost of the last cluster $\mathcal{C}$, that is $\max_{u,v\in\mathcal{C}} d(u,v)$ where $\mathcal{C} = \{c_i^j \in C \mid i'_j = i_j - 1, j \in [1..m]\}$, over all $(i'_1, i'_2, \ldots, i'_m)$. See Algorithm **find ℓ-diversity clustering**.

Algorithm 1 find ℓ-diversity clustering (C, Q, ℓ)

```
for i_1 = 0 to |C^1| do
  for i_2 = 0 to |C^2| do
    ⋮
    for i_m = 0 to |C^m| do
        // exactly one cluster case //
      if i_1 ≤ 1, i_2 ≤ 1, ··· , i_m ≤ 1 and ℓ ≤ i_1 + i_2 + ··· + i_m ≤ 2ℓ − 1 then
        set C = {c_1^j ∈ C | i_j = 1, j ∈ [1..m]}
        cost(i_1, i_2, ..., i_m) = max_{u,v∈C} d(u, v)
      else if i_1 + i_2 + ··· + i_m ≥ 2ℓ then
        // two or more clusters case //
        cost(i'_1, i'_2, ..., i'_m) = ∞
        for i'_1 = i_1 − 1 to i_1 do
          for i'_2 = i_2 − 1 to i_2 do
            ⋮
            for i'_m = i_m − 1 to i_m do
              if ℓ ≤ (i_1 − i'_1) + (i_2 − i'_2) + ··· + (i_m − i'_m) ≤ 2ℓ − 1 and P(i'_1, i'_2, ··· , i'_m) has a solution then
                set C = {c_i^j ∈ C | i'_j = i_j − 1, j ∈ [1..m]}
                if cost(i'_1, i'_2, ..., i'_m) > max{cost(i'_1, i'_2, ..., i'_m), max_{u,v∈C} d(u, v)} then
                  // update to the smaller cost //
                  cost(i'_1, i'_2, ..., i'_m) = max{cost(i'_1, i'_2, ..., i'_m), max_{u,v∈C} d(u, v)}
                end if
              end if
            end for
            ⋮
          end for
        end for
      end if
    end for
    ⋮
  end for
end for
return the ℓ-diversity clustering of P(|C^1|, |C^2|, ..., |C^m|)
```

Next we analyze the running time.

The most inner part runs $(|C^1|+1) \times (|C^2|+1) \times \cdots \times (|C^m|+1) \times 2^m$ times. By arithmetic-geometric mean $\left((|C^1|+1) + (|C^2|+1) + \cdots + (|C^m|+1)\right)/m \geq \sqrt[m]{(|C^1|+1) \times (|C^2|+1) \times \cdots \times (|C^m|+1)}$ holds. Since $|C| = |C^1| + |C^2| + \cdots + |C^m| = n$, $\left((|C^1|+1) + (|C^2|+1) + \cdots + (|C^m|+1)\right)/m = (n+m)/m = n/m + 1$, so $(n/m+1)^m \geq (|C^1|+1) \times (|C^2|+1) \times \cdots \times (|C^m|+1)$ holds.

Therefore the running time of the algorithm is as follows.

Theorem One can solve the ℓ-diversity problem in $O(2^m \cdot (\frac{n}{m}+1)^m)$ time when all C are on the line.

3 Conclusion

We designed an algorithm to solve the min-max version of the ℓ-diversity clustering problem. The running time of the algorithm is $O(2^m \cdot (\frac{n}{m}+1)^m)$ time.

We can similarly design an algorithm to solve the min-sum version of the ℓ-diversity clustering problem. The running time of the algorithm is $O(2^m \cdot (\frac{n}{m}+1)^m)$ time.

References

[1] A. Armon, “On min-max r-gatherings”, Theoretical Computer Science, 412, pp.573-582, 2011.

[2] G. Ghinita, P. Karras, P. Kalnis and N. Mamoulis, “Fast data anonymization with low information loss”, Proc. of the 33rd international conference on Very large data bases, VLDB’07, pp.758-769, 2007.

[3] J. Li, K. Yi and Q. Zhang, “Clustering with diversity”, Proc. of ICALP2010, LNCS 6198, pp.188-200, 2010.