Maximize Nearest Neighbor Distance

Hsueh-Lin Huang

April 2016

1 Algorithm

I proposed an algorithm in a stochastic manner. In each iteration, I picked up certain x from N points to update, with the goal to keep current nearest-neighbor relations for all points. The algorithm are divided into a few steps. For the convenience, here we define some variables. For every x, its nearest neighbor is denoted as $x^{(1)}$, and the second-nearest neighbor is denoted as $x^{(2)}$. Also, we assume the nearest neighbor of $z_1, \ldots, z_n$ is x, which means $z_1^{(1)} = \ldots = z_n^{(1)} = x$.

1.1 Determine the Gradient

Determine the update could be divided into two steps, including the update direction g and the step size η. If we assume the relations of nearest neighbors are the same after the update, we should maximize this objective:

$$\max_g \|x + \eta g - x^{(1)}\|^2 + \sum_{i=1}^{n} \|x + \eta g - z_i\|^2.$$

Since the above is a quadratic function, the gradient ascent algorithm ensure the objective increasing. Therefore, we take

$$g = x - x^{(1)} + \sum_{i=1}^{n} x - z_i.$$

1.2 Keep Own Nearest Neighbor

We hope $x^{(1)}$ is still the nearest neighbor of x after the update, so the naive constraint is

$$\|x + \eta g - x^{(1)}\| \leq \|x - x'\|, \forall x' \neq x.$$

We then need to consider N constraints, which may be slow in practice. However, since each step is no larger than 2, so we have

$$\|x + \eta g - x^{(1)}\| \leq \|x - x^{(1)}\| + 2,$$

also we have

$$\|x + \eta g - x'\| \geq \|x - x'\| - 2, \forall x' \neq x,$$

Combining these two we get $\|x + \eta g - x'\| \geq \|x + \eta g - x^{(1)}\|$ if

$$\|x - x'\| \geq \|x - x^{(1)}\| + 4.$$

It suggests we do not need to consider the points whose distance to x is more than $\|x - x^{(1)}\| + 4$.

1.3 Keep Others' Nearest Neighbors

For every point p, we hope $p^{(1)}$ does not change after the update. Since we only move x, the only possibility is x is moving toward certain p and replace the original $p^{(1)}$.

With the same idea above, we have $\|x + \eta g - p\| \geq \|x - p\| - 2$, so x will not replace $p^{(1)}$ if $\|x - p\| - 2 \geq \max_p \|p - p^{(1)}\|$. Then we only need consider the p, where $\|p - x\| \geq \max_p \|p - p^1\| + 2$ with the constraint

$$\|x + \eta g - p\| \geq \|p - p^{(1)}\|.$$

2 Implementation Details

First, we introduce what we should keep for every point x. For every point, we keep its some auxiliary variables, including nearest neighbor $x^{(1)}$ and $z_1, \ldots, z_n$, whose nearest points is x.

In every iteration, we only need to update z and x's auxiliary variables, since x may move away from certain z_i, then z_i's nearest neighbor will change to the original $z_i^{(2)}$. Therefore, we first query the points x', where $\|x' - z_i\| < \|x + \eta g - z_i\|$. If there is no such points, we do not need to do any update. If it is, we just need to update $z_i^{(1)}$ and remove z_i from x's auxiliary variables. For other points, the above math derivation guarantee that we do not need to update their auxiliary variables.

Each our algorithms, we keep doing query points within a ball after given the center and the radius. We make this step efficient by keeping a three dimension `point_hash`, where `point_hash[i][j][k]` is a set keeping points

$$\{(x, y, z) | i \leq \lfloor \frac{x}{10} \rfloor < i + 1, j \leq \lfloor \frac{y}{10} \rfloor < j + 1, k \leq \lfloor \frac{z}{10} \rfloor < k + 1.\}$$

Also, querying a ball is difficult, so we query a cubic which can cover the ball instead. In every iteration, to keep this hash correct, we only need to update two entries for x and $x + \eta g$.

3 Reference

The code of OpenGL part is from my previous homework.