Efficient and Fair Line Construction

DEVTech

Problem Statement

Consider a set of n houses with coordinates given by latitude (x) and longitude (y), defined as a set $P = \{p_1, p_2, \ldots, p_n\}$. The task is to construct a gas pipeline in a straight line that serves all of them, where a straight line $l = \{a, b\}$ is defined by a direction vector a of unit norm and a point b on the line.

Distance Definition: The distance from a point p to a line $l = \{a, b\}$ is given by:

$$\operatorname{dist}(p, l) := \left\|(I - aa^T) \cdot (p - b)\right\|^2$$

Objective 1: Efficient Line

A line is efficient if it minimizes the following cost:

$$\sum_{i=1}^{n} \|\operatorname{dist}(p_i, l)\|$$

1 Trivial Idea:

Expand the L_2 Norm squared term, take gradient with respect to the parameters a and b, solve the equations keeping the constraint on a being a unit norm by applying Lagrange Operator.

2 Key Decisions Made:

Solving the equations was computationally very hard, so went to apply Gradient Descent to get close to optimal value.

3 Proposed Solution

Initialization: We initialize the direction vector a randomly and normalize it to ensure it has a unit norm. The point b on the line is initialized as the centroid of the dataset.

Cost Function: We define a cost function that computes the total distance from each house to the line. This cost function is the sum of squared distances from each house to the line.

Gradient Descent: We apply gradient descent to iteratively update the parameters a and b in the direction that minimizes the cost function. In each iteration, we compute the gradients of the cost function with respect to a and b and update them accordingly.

Termination: We terminate the optimization process when the change in the cost function becomes negligible or after reaching a predefined maximum number of iterations.

Output: Finally, we obtain the optimal direction vector a and point b on the line, along with the minimum cost achieved.

The proposed solution efficiently searches for the line that serves all houses with the minimum total distance. By iteratively updating the parameters using gradient descent, we aim to converge to a locally optimal solution that minimizes the cost function.

4 Output

1. For a partition size of 5000 of the dataset:
    - **Optimal lines:**
        - **Line 1**
            - Direction vector a: $[2.73331076 \times 10^{140}, -1.65294755 \times 10^{140}]$
            - Point b on the line: $[-6.37186689 \times 10^{140}, 2.55546158 \times 10^{140}]$
            - Minimum cost: 19724.781917739136
2. For the complete dataset:
    - **Optimal lines:**
        - **Line 1**
            - Direction vector a: $[-1.24474830 \times 10^{219}, 2.11462956 \times 10^{219}]$
            - Point b on the line: $[-5.51451371 \times 10^{220}, 1.21020160 \times 10^{221}]$
            - Minimum cost: 152352.4433986061

5 Challenges Faced

- Iterative normalization of vector a.
- Changing the learning rate to minimize the cost.

Objective 2: Fair Line Optimization for Gas Pipeline Construction

Introduction

The objective of this project is to optimize the construction of a gas pipeline to serve a set of houses, minimizing the maximum distance of any house from the pipeline. This task involves finding a fair line that efficiently serves all houses.

Challenges Encountered

1. **Optimization Convergence:** Initially, the optimization process did not converge effectively, leading to suboptimal results.
2. **Learning Rate Selection:** Choosing an appropriate learning rate for the optimization algorithm posed a challenge as it directly impacts the convergence behavior.
3. **Parameter Initialization:** The initial values for the direction vector and point on the line required careful consideration for effective optimization.

Key Decisions Made

1. **Use PyTorch:** We used PyTorch because it efficiently calculates gradients of functions like max which is present in the objective function.
2. **Adjustment of Learning Rate:** Experimented with different learning rates to find one that balances convergence speed and stability.
3. **Initialization Strategy:** Initialized the point on the line at the centroid of the house coordinates to provide a reasonable starting point for optimization.
4. **Optimization Algorithm Selection:** Switched to using the Adam optimizer, which adapts the learning rate during optimization and is known to perform well in various scenarios.

Solutions Implemented

1. **Learning Rate Adjustment:** Set the learning rate to 0.1 to facilitate more stable convergence without sacrificing speed.
2. **Parameter Initialization:** Initialized the point on the line at the centroid of the house coordinates to provide a better starting point for optimization.
3. **Optimization Algorithm Switch:** Utilized the Adam optimizer instead of standard gradient descent for improved convergence behavior.

Results

The implemented solutions led to improved convergence behavior and more effective optimization. The pipeline construction process now yields fair lines that efficiently serve all houses, minimizing the maximum distance of any house from the pipeline.

Conclusion

Through careful consideration of optimization parameters and algorithm selection, the project successfully addressed the challenges encountered and achieved the objective of optimizing gas pipeline construction for fair line placement.

Objective 3: Multiple Efficient Lines

For a set of k lines $L = \{l_1, \ldots, l_k\}$, the set is efficient if it minimizes:

$$\sum_{i=1}^{n} \min_{l \in L} \|\text{dist}(p_i, l)\|$$

Design an algorithm that computes k efficient lines or a set of lines which is almost efficient, meaning the above cost is a local minimum.

Challenges Encountered

1. **Optimization Convergence:** The optimization process did not converge effectively initially, leading to suboptimal results.
2. **Learning Rate Selection:** Choosing an appropriate learning rate for the optimization algorithm posed a challenge as it directly impacts convergence behavior.
3. **Parameter Initialization:** Initializing the direction vector and point on the line required careful consideration for effective optimization.

Key Decisions Made

1. **Use of PyTorch:** Utilized PyTorch for efficient gradient calculation.
2. **Learning Rate Adjustment:** Experimented with different learning rates to balance convergence speed and stability.
3. **Initialization Strategy:** Initialized the point on the line at the centroid of the house coordinates for a better starting point.
4. **Optimization Algorithm Selection:** Switched to using the Adam optimizer for improved convergence behavior.

Solutions Implemented

1. **Learning Rate Adjustment:** Set the learning rate to 0.1 for more stable convergence.
2. **Parameter Initialization:** Initialized the point on the line at the centroid of the house coordinates for a better starting point.
3. **Optimization Algorithm Switch:** Utilized the Adam optimizer for improved convergence behavior.

Output

- **Optimal lines:**
  - **Line 1**
    - Direction vector a: $[2.73331076 \times 10^{140}, -1.65294755 \times 10^{140}]$
    - Point b on the line: $[-6.37186689 \times 10^{140}, 2.55546158 \times 10^{140}]$
    - Minimum cost: 19724.781917739136

Challenges Faced

- Iterative normalization of vector a.
- Changing the learning rate to minimize the cost.