The Application of ML to Identifying Critical Constraints

in The Inverse Uncapacitated Facility Location Problem

Zikang LI

COMP 5212

Uncapacitated Facility Location Problem(UFLP)

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}u_{ij} + \sum_{i=1}^{m} f_i v_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} u_{ij} = 1, \quad \forall j \in N & (1) \\
& u_{ij} \leq v_i, \quad \forall i \in M, \forall j \in N & (2) \\
& u_{ij}, v_i \in \{0,1\}, \quad \forall i \in M, \forall j \in N
\end{aligned}
$$

(1) represents each customer $j \in N$ should be severd by one facility.
(2) represents if customer j is served by faility i, then facility i must be open.

IUFLP

$$
\begin{aligned}
\min_{f^0, r^0} \quad & \sum_{i=1}^{m} w_i^f |f_i^0 - f_i| + \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}^r |r_{ij}^0 - r_{ij}| \\
\text{s.t.} \quad & f_i^0 v_i^0 + r_{ij}^0 u_{ij}^0 \leq f_i^0 v_i^{'} + r_{ij}^0 u_{ij}^{'}, \quad \forall i \in M, \forall j \in N && (3) \\
& (v^{'}, u^{'}) \in \Upsilon && (4)
\end{aligned}
$$

Υ is the set of all feasible solutions $x^{'} = (v^{'}, u^{'})$.
The number of solutions satisfying constraints (3) is m^n.
Only a few constraints will be critical to determine the feasible region of this problem. Once we can obtain the set of all critical constraints, we will get the optimal solution.

Dataset Construction

- Set $w^f = 1, w^r = 1$ for simplicity.
- Data point $(f, r, v^0, u^0, v^{'}, u^{'})$.
- Fixed cost: (f, r), the given optimal solution (v^0, u^0), $v^{'}, u^{'}$ is any feasible solution to UFL.
- Use constraint (3) to label the data.
- If $f^0(v^0 - v^{'}) + r^0(u^0 - u^{'}) = 0$, then $(v^{'}, u^{'})$ is a critical constraint, the corresponding data will be labeled 1; otherwise, labeled 0.

Training Procedure

- The cost (f, r) can be very large while the solution always only contains 0,1. Thus, we need to normalize (f, r).
- The number of critical constraints will be way smaller than that of feasible solutions in each instance. Thus, we should avoid generating a feasible solution randomly as the data, otherwise, the dataset will be unbalanced.
- The batch size is 10; the number of epochs is 80; learning rate is 0.001; momentum is 0.9 for SGD-Momentum; the input size is $(m + m * n) * 3$.

Results

(m,n)	Epoch	Logistic(SGD-M)	SVM(SGD-M)	Accuracy
(5,10)	20	0.2495	0.2669	
	40	0.2247	0.2669	
	60	0.2106	0.2670	
	80	0.2018	0.2670	100%, 99%
(10,20)	20	0.3935	0.7971	
	40	0.3799	0.7961	
	60	0.3741	0.7964	
	80	0.3715	0.7961	95%, 93%
(20,40)	20	0.3474	0.5028	
	40	0.3148	0.5028	
	60	0.2978	0.5024	
	80	0.2868	0.5030	99%, 98%

Challenges

- The results depend on the dataset. The different size of datasets and how the datasets are generated will give a very different result.
- Different sizes of m, n need different datasets. Constructing the dataset will be time-consuming with the large size of m, n.
- Finding the same feature behind the different size of this problem will be very crucial to solving the large-scale problem.

Thank You!