Outlier re-insertion via Mixed-Integer Linear Programming

Dimitri J. Papageorgiou
Corporate Strategic Research
ExxonMobil Research and Engineering Company
1545 Route 22 East, Annandale, NJ 08801 USA
dimitri.j.papageorgiou@exxonmobil.com

December 19, 2016

1 Outlier re-insertion

We assume that the distance metric d_{ij} is fixed for the remainder of this section. Let $\mathcal{N}$ be the set of all points, $\mathcal{C}_i$ the set of co-class neighbors of point i, $\bar{\mathcal{C}}_i$ the set of non co-class neighbors of point i, and $\mathcal{O}$ the current set of outliers. Let $\mathcal{IK} = \{(i,k) \in \mathcal{N} \times \mathcal{N} : \nexists\, j \in \mathcal{C}_i \backslash \mathcal{O} : d_{ij} + \epsilon \leq d_{ik}, k \in \bar{\mathcal{C}}_i\}$. Set $M_{ik} = 1 - d_{ik}$. (We are making use of our normalization $d_{ij} \in [0,1]$ here.) For any outlier penalty parameter $\rho > 0$ (I suggest $\rho = 1$), the following mixed-integer linear program attempts to minimize the number of outliers given a fixed distance metric.

$$\min_{\mathbf{y},\mathbf{z}} \quad \rho \sum_{i \in \mathcal{O}} z_i \tag{1a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{C}_i} d_{ij} y_{ij} + \epsilon \leq d_{ik} + M_{ik} z_k \qquad \forall (i,k) \in \mathcal{IK} \tag{1b}$$

$$\sum_{j \in \mathcal{C}_i} y_{ij} = 1 \qquad \forall i \in \mathcal{N} \backslash \mathcal{O}, j \in \mathcal{C}_i \tag{1c}$$

$$\sum_{j \in \mathcal{C}_i} y_{ij} = 1 - z_i \qquad \forall i \in \mathcal{O}, j \in \mathcal{C}_i \tag{1d}$$

$$y_{ij} \leq 1 - z_j \qquad \forall i \in \mathcal{N}, j \in \mathcal{C}_i \cap \mathcal{O} \tag{1e}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in \mathcal{N}, j \in \mathcal{C}_i \tag{1f}$$

$$z_i \in \{0,1\} \qquad \forall i \in \mathcal{O} \tag{1g}$$

$$z_i = 0 \qquad \forall i \in \mathcal{N} \backslash \mathcal{O} \tag{1h}$$

Several observations:

1. Bolun wrote the RHS of constraint (**??**) as $M_i(z_i + z_k)$. This is correct, but the formulation above is tighter, meaning the linear programming relaxation provides a better lower bound, and thus optimizers believe it will be better. Why can we replace $M_i(z_i + z_k)$ with $M_{ik} z_k$? Now that we have constraints (**??**), which effectively set the LHS of (**??**) to ϵ if point i is deemed an outlier ($z_i = 1$), there is no reason to include $M_{ik} z_i$ on the RHS of (**??**) as well.

2. What is the set $\mathcal{IK}$ doing? What is its purpose? Given that the distance metric is fixed, for any pair of points $(i,k) : i \in \mathcal{N}, k \in \bar{\mathcal{C}}_i$, if

$$\min\{d_{ij} : j \in \mathcal{C}_i \backslash \mathcal{O}\} + \epsilon \leq d_{ik}$$

which holds if and only if

$$\exists j \in \mathcal{C}_i \backslash \mathcal{O} : d_{ij} + \epsilon \leq d_{ik} \ , \tag{2}$$

then the corresponding constraint (**??**) is already satisfied (and will continue to be satisfied once outliers are considered) and is therefore redundant.

3. Is it really worth the trouble to create the set $\mathcal{IK}$? Won't the solver take care of this? While solvers are good at eliminating redundant constraints, it is best to help them eliminate constraints a priori especially when we can immediately identify them. This saves the solver time when building the optimization model and in preprocessing. Example: Suppose there are 10 classes, 100 points per class, for a total of 1000 points. For simplicity, suppose there are 3 outliers per class (that is, our algorithm has thus far identified 3 outliers per class so that $|\mathcal{O}| = 30$). There are 900,000 constraints that must be included if constraints (**??**) are written "$\forall i \in \mathcal{N}, k \in \bar{\mathcal{C}}_i$" since, for every point i (of which there are 1,000), there are 900 non-neighbors. However, if we were to replace "$\forall i \in \mathcal{N}, k \in \bar{\mathcal{C}}_i$" with $\forall (i,k) \in \mathcal{IK}$, then at least 846,810 (94%) of the constraints of type (**??**) can be eliminated before doing a single optimization. Why? In this example, there are 970 non-outliers, each of which has 9×97 non-neighbors that are not outliers. These 846,810 pairs of points have already been shown to satisfy constraints (**??**) to arrive at the current distance metric. We write "at least" because there are likely many other pairs of points involving outliers (i.e., $k \in \mathcal{O}$) for which the condition (**??**) also holds.

1.1 AIMMS implementation

In AIMMS, there are several items to be aware of. First, define the set $\mathcal{IK}$ as follows:

$$\mathcal{IK} = \{(i,k) | k \in \bar{\mathcal{C}}_i \ \texttt{and not exists}(j \in \mathcal{C}_i \backslash \mathcal{O} | d_{ij} + \epsilon \leq d_{ik})\}$$

Second, z_i should only be defined for $i \in \mathcal{O}$. You should *not* define it for all $i \in \mathcal{N}$ and then set $z_i = 0$ for all $i \in \mathcal{N} \backslash \mathcal{O}$ as in (**??**). Note that AIMMS understands that if z_i appears in a constraint written for all $i \in \mathcal{N}$, it will only include it when z_i exists.

Third, in summary, the following formulation should be used in AIMMS:

$$\min_{\mathbf{y},\mathbf{z}} \quad \rho \sum_{i \in \mathcal{O}} z_i \tag{3a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{C}_i} d_{ij} y_{ij} + \epsilon \leq d_{ik} + M_{ik} z_k \qquad \forall (i,k) \in \mathcal{IK} \tag{3b}$$

$$\sum_{j \in \mathcal{C}_i} y_{ij} = 1 - z_i \qquad \forall i \in \mathcal{N}, j \in \mathcal{C}_i \tag{3c}$$

$$y_{ij} \leq 1 - z_j \qquad \forall i \in \mathcal{N}, j \in \mathcal{C}_i \cap \mathcal{O} \tag{3d}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in \mathcal{N}, j \in \mathcal{C}_i \tag{3e}$$

$$z_i \in \{0,1\} \qquad \forall i \in \mathcal{O} \tag{3f}$$