Linear Optimization of filtering for AUI

Yihao Du

December 5, 2017

1 Problem definition

In this problem, the goal is to develop an efficient algorithm for optimize filter setting for the retrieval set associated with maximum set-score.

The scenario of global element set E includes m unique elements. Each of the elements in this global set is associated with its score S, which is used for further estimation of set-score. We intend to develop a algorithm (f) of optimize filter setting for the optimal retrieval set (L).

$$\underset{f}{\text{maximize}} \quad \mathit{L(f(E))} \tag{1}$$

It is easy to observe two significant parts in this problem: How to efficiently filter unnecessary elements (f)? How to evaluate different sets (L)?

2 Set Evaluation Metrics

In information retrieval, a Retrieval Set (RS) is usually evaluated using three main metrics: precision, recall and F-score, which are defined as follows:

$$P(RS) = \frac{\sum\limits_{i \in RS} S(i)}{|RS|} = \frac{\sum\limits_{i=0}^{m} S(i) \cdot I(i)}{\sum_{i=0}^{m} I(i)} \tag{2}$$

$$R(RS) = \frac{\sum\limits_{i \in RS} S(i)}{|RE|} = \frac{\sum\limits_{i=0}^{m} S(i) \cdot I(i)}{\sum\limits_{i=0}^{m} S(i)} = \frac{\sum\limits_{i=0}^{m} S(i) \cdot I(i)}{C} \tag{3}$$

$$F(RS) = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times \sum\limits_{i=0}^{m} S(i) \cdot I(i)}{\sum\limits_{i=0}^{m} I(i) + \sum\limits_{i=0}^{m} S(i)} = \frac{2 \times \sum\limits_{i=0}^{m} S(i) \cdot I(i)}{\sum\limits_{i=0}^{m} I(i) + C} \tag{4}$$

where, $S(i)$ is a binary score indication function defined as:

$$S(i) = \begin{cases} 1, & \text{if element } i \text{ is relevant} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Similarly, $I(i)$ is a binary variable that indicates whether an element i is selected in retrieval set, and is formally defined as:

$$I(i) = \begin{cases} 1, & \text{iff } i \in RS \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Finally, $|RE|$ is the size of the relevant element set. The denominator in *Recall* is score sum of global element set E, which only depends on score of data. Therefore, it is a constant C in our problem.

In practice, a probabilistic score is used to approximate $S(i)$ because of unavailability of real label. Therefore, this relaxation of the score $S(i)$ into a continuous value results in *expected* values for the metrics above.

3 Search Algorithm Validation - Metrics Optimization

In order to validate search algorithm, we are able to compare its result with the one obtained by metrics optimization. The objective function in this optimization problem is formulated based on different metrics (Precision, Recall and F-score). The constraints are defined based on different fields (time, keyword and position).

3.1 Objective functions

3.1.1 Precision

The optimization problem to maximize precision is formulated as follows:

$$\begin{aligned} \underset{I}{\text{maximize}} \quad & \frac{\sum_{i=0}^{m} S(i) \cdot I(i)}{\sum_{i=0}^{m} I(i)} \\ s.t \quad & I(i) \in \{0,1\}, \forall i \in RS \end{aligned} \tag{7}$$

Note this is a Mixed-Integer Linear Fractional Programming (MILFP) problem, so Charnes-Cooper transformation is used to transform the problem into a Mixed-Integer Nonlinear Programming (MINLP) as follows:

$$\begin{aligned} \underset{I}{\text{maximize}} \quad & \sum_{i=0}^{m} S(i) \cdot u \cdot I(i) \\ s.t \quad & \sum_{i=0}^{m} u \cdot I(i) = 1 \\ & u > 0, I(i) \in \{0,1\}, \forall i \in RS \end{aligned} \tag{8}$$

where u is defined as follows:

$$u = \frac{1}{\sum_{i=0}^{m} I(i)} \tag{9}$$

Next, Glover's linearization scheme is used to convert the MINLP into an equivalent Mixed-Integer Linear Programming (MILP) problem that can be directly optimized. Hence, a new variables $y(i) = u \cdot I(i)$ is introduced again and using this, an equivalent MILP is given:

$$\begin{array}{ll}\underset{y}{\text{maximize}} & \sum_{i=0}^{m} S(i) \cdot y(i) \\ s.t & \sum_{i=0}^{m} y(i) = 1 \\ & y(i) \leqslant u, \forall i \in RS \\ & y(i) \leqslant M \cdot I(i), \forall i \in RS \\ & y(i) \geqslant u - M \cdot (1 - I(i)), \forall i \in RS \\ & u > 0, y(i) \geq 0, I(i) \in \{0,1\}, \forall i \in RS \end{array} \tag{10}$$

3.1.2 Recall

Maximization of Recall can be formulated as the following optimization problem:

$$\begin{array}{ll}\underset{I}{\text{maximize}} & \dfrac{\sum_{i=0}^{m} S(i) \cdot I(i)}{C} \\ s.t & I(i) \in \{0,1\}, \forall i \in RS \end{array} \tag{11}$$

This problem is reformulated with Glover's linearization as follows:

$$\begin{array}{ll}\underset{y}{\text{maximize}} & \sum_{i=0}^{m} S(i) \cdot y(i) \\ s.t & y(i) \leqslant \frac{1}{C}, \forall i \in RS \\ & y(i) \leqslant M \cdot I(i), \forall i \in RS \\ & y(i) \geqslant u - M \cdot (1 - I(i)), \forall i \in RS \\ & y(i) \geq 0, I(i) \in \{0,1\}, \forall i \in RS \end{array} \tag{12}$$

3.1.3 F-score

Maximization of F-score can be formulated as the following optimization problem:

$$\begin{array}{ll}\underset{I}{\text{maximize}} & \dfrac{\sum_{i=0}^{m} S(i) \cdot I(i)}{\sum_{i=0}^{m} I(i) + C} \\ s.t & I(i) \in \{0,1\}, \forall i \in RS \end{array} \tag{13}$$

—

Note this is a Mixed-Integer Linear Fractional Programming (MILFP) problem, so Charnes-Cooper transformation is used to transform the problem into a Mixed-Integer Nonlinear Programming (MINLP) as follows:

$$\begin{array}{ll}\underset{I}{\text{maximize}} & \sum_{i=0}^{m} S(i) \cdot u \cdot I(i) \\ s.t & \sum_{i=0}^{m} u \cdot I(i) \leq 1 \\ & u > 0, I(i) \in \{0,1\}, \forall i \in RS \end{array} \tag{14}$$

where u is defined as follows:

$$u = \frac{1}{\sum_{i=0}^{m} I(i) + C} \tag{15}$$

—

With Charnes-Cooper transformation and Glover's linearization method, this optimization problem can be transformed into the following problem.

$$\begin{array}{ll}
\underset{y}{\text{maximize}} & \sum_{i=0}^{m} S(i) \cdot y(i) \\
s.t & \sum_{i=0}^{m} y(i) + uC = 1 \\
& y(i) \leqslant u, \forall i \in RS \\
& y(i) \leqslant M \cdot I(i), \forall i \in RS \\
& y(i) \geqslant u - M \cdot (1 - I(i)), \forall i \in RS \\
& u > 0, y(i) \geq 0, I(i) \in \{0,1\}, \forall i \in RS
\end{array} \tag{16}$$

3.2 Constraints

3.2.1 Time

Time filter setting is a two - element tuple (ts, te), which indicates start (ts) and end (te) of time-line. For element i with its time $t(i)$, the time constraint for $I(i)$ can be formulated as follows:

$$I(i) = \begin{cases} 1, & \text{if } (ts \leqslant t(i)) \wedge (t(i) \leqslant te) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

3.2.2 Keyword

Keyword filter setting is a set of unigram (K), which filters the element does not have these words. For element i with its whole unigram set $W(i)$, the keyword constraint for $I(i)$ can be formulated as follows:

$$I(i) = \begin{cases} 1, & \text{if } K \cap W(i) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The implementation method is still unclear in keyword field.

3.2.3 Position

Position filter setting is a four - element tuple (xmin, ymin, xmax, ymax), which filters the element out of a bounding box created based on tuple. For element i with its position $(x(i), y(i))$, the position constraint for $I(i)$ can be formulated as follows:

$$I(i) = \begin{cases} 1, & \text{if } (xmin \leqslant x(i)) \wedge (x(i) \leqslant xmin) \wedge (ymin \leqslant y(i)) \wedge (y(i) \leqslant ymax) \\ 0, & \text{otherwise} \end{cases} \tag{19}$$