# Optimal accuracy threshold using 0-1 ILP formulation

Dr. Pouria Ramazi, Arash Mari Oriyad

**Abstract**

Classification

For a binary classification problem, the accuracy measure which is the portion of the correctly predicted instances with respect to all instances, can be calculated as below,

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where $TP$, $TN$, $FP$, and $FN$ are the number of true-positive, true-negative, false-positive, and false-negative instances, respectively.

While accuracy requires the predicted labels, classification models produce a probability distribution over all available classes for each instance. Hence, one must apply a threshold to the predicted probabilities to obtain deterministic labels. An optimal threshold for a given set of instances is a real value between 0 and 1, providing the highest accuracy.

A trivial approach to determine such a threshold is an exhaustive search that examines all the possible threshold values to obtain the best one. Complete search-based methods are usually not efficient in execution time, especially when the solution space grows. To the best of our knowledge, there is no programming package to find the optimal accuracy threshold without using an exhaustive search.

We treated the optimal accuracy threshold problem as an optimization problem by introducing an integer linear programming (ILP) formulation. Assume a set of $N$ instances with a binary target variable $y$ and suppose $\hat{y}_i$ is the predicted probability for the $i$th instance. Then, $\tilde{y}_i$ is a binary decision variable which is desirable to be calculated as follows:

$$\tilde{y}_i = \begin{cases} 0, & \hat{y}_i < t \\ 1, & \hat{y}_i \geq t \end{cases}$$

where $t$ is the threshold. More precisely, $\tilde{y}_i$ represents the label of $i$th instance after applying the threshold $t$ to the predicted probability $\hat{y}_i$. Hence, the accuracy measure can be calculated as bellow.

$$accuracy = 1 - \frac{1}{N} \sum_{i=1}^{N} |y_i - \tilde{y}_i|$$

Put them all together, the following optimization formulation is achieved.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N} |y_i - \tilde{y}_i| \\
\text{subject to:} \quad & \tilde{y}_i = \begin{cases} 0, & \hat{y}_i < t \\ 1, & \hat{y}_i \geq t \end{cases} \qquad \forall 1 \leq i \leq N \\
& \tilde{y}_i \in \{0,1\} \qquad \forall 1 \leq i \leq N
\end{aligned} \tag{1}
$$

Note that $y_i$ and $\hat{y}_i$ are input parameters and $\tilde{y}_i$ is the decision variable of the problem. Moreover, $\frac{1}{N}$ is ignored in the objective function since $N$ does not affect the final solution.

To transform the proposed constraints into a linear form, for each $i$, we can combine the conditions and replace them with two new constraints. Similarly, to make the objective function linear, a new binary decision variable $z_i = |y_i - \tilde{y}_i|$ is defined and two new constraints are added to the optimization formulation as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N} z_i \\
\text{subject to:} \quad & z_i \geq y_i - \tilde{y}_i & \forall 1 \leq i \leq N \\
& z_i \geq \tilde{y}_i - y_i & \forall 1 \leq i \leq N \\
& \hat{y}_i(1 - \tilde{y}_i) < t & \forall 1 \leq i \leq N \\
& \tilde{y}_i(\hat{y}_i - 1) \geq t - 1 & \forall 1 \leq i \leq N \\
& \tilde{y}_i \in \{0,1\} & \forall 1 \leq i \leq N \\
& 0 \leq t \leq 1
\end{aligned} \tag{2}
$$

Hence, we have designed a 0-1 ILP formulation with $2N$ binary decision variables and $4N$ linear constraints.

The proposed ILP formulation is compared with the exhaustive search method in terms of average execution time (Table 1).

Table 1: Comparing the ILP-based approach with the exhaustive search method for finding the optimal accuracy threshold.

| # Instances | Prediction Decimal | ILP (Python) | Exhaustive Search (Python) | ILP (C++) | Exhaustive Search (C++) | Improved Exhaustive Search (C++) |
|---|---|---|---|---|---|---|
| 1000 | 1 | 0.032 | 0.025 | 0.0068 | 0.0024 | ? |
| 1000 | 2 | 0.150 | 0.199 | 0.067 | 0.0062 | ? |
| 1000 | 3 | 1.133 | 1.243 | 0.802 | 0.028 | ? |
| 1000 | 4 | 1.748 | 1.884 | 1.311 | 0.041 | ? |
| 1000 | 5 | 1.807 | 1.860 | 1.339 | 0.045 | ? |
| 2000 | 1 | 0.033 | 0.026 | 0.010 | 0.0042 | ? |
| 2000 | 2 | 0.160 | 0.205 | 0.076 | 0.013 | ? |
| 2000 | 3 | 2.747 | 1.721 | 2.152 | 0.076 | ? |
| 2000 | 4 | 6.378 | 3.533 | 5.173 | 0.159 | ? |
| 2000 | 5 | ? | ? | ? | 0.172 | ? |
| 5000 | 1 | ? | ? | 0.023 | 0.010 | ? |
| 5000 | 2 | ? | ? | 0.098 | 0.030 | ? |
| 5000 | 3 | 5.998 | 2.075 | 5.060 | 0.234 | ? |
| 5000 | 4 | ? | ? | ? | 0.875 | ? |
| 5000 | 5 | 50.663 | 9.747 | ? | 1.091 | ? |
| $10^4$ | 1 | ? | ? | 0.043 | 0.0216 | ? |
| $10^4$ | 2 | ? | ? | 0.127 | 0.0634 | ? |
| $10^4$ | 3 | ? | ? | 5.527 | 0.397 | ? |
| $2 \times 10^4$ | 1 | ? | ? | 0.086 | 0.0421 | ? |
| $2 \times 10^4$ | 2 | ? | ? | 0.179 | 0.130 | ? |
| $2 \times 10^4$ | 3 | ? | ? | 6.227 | 0.818 | ? |

| | | | | | | |
|---|---|---|---|---|---|---|
| $5 \times 10^4$ | 1 | ? | ? | 0.212 | 0.115 | ? |
| $5 \times 10^4$ | 2 | ? | ? | 0.314 | 0.330 | ? |
| $5 \times 10^4$ | 3 | ? | ? | 7.115 | 2.059 | ? |
| $10^5$ | 1 | ? | ? | 0.429 | 0.218 | ? |
| $10^5$ | 2 | ? | ? | 0.539 | 0.642 | 0.216 |
| $*10^5$ | 3 | ? | ? | 7.351 | 5.014 | 0.275 |
| $2 \times 10^5$ | 1 | ? | ? | 0.866 | 0.439 | ? |
| $2 \times 10^5$ | 2 | ? | ? | 0.966 | 1.370 | 0.425 |
| $*2 \times 10^5$ | 3 | ? | ? | 8.462 | 10.518 | 0.506 |
| $5 \times 10^5$ | 1 | ? | ? | 2.108 | 1.102 | ? |
| $5 \times 10^5$ | 2 | ? | ? | 2.254 | 3.642 | 1.055 |
| $*5 \times 10^5$ | 3 | ? | ? | 10.771 | 29.151 | 1.251 |
| $10^6$ | 1 | ? | ? | 4.131 | 2.229 | ? |
| $10^6$ | 2 | ? | ? | 4.451 | 7.088 | 2.135 |
| $*10^6$ | 3 | ? | ? | 13.010 | 60.555 | 2.685 |
| $2 \times 10^6$ | 1 | ? | ? | 8.198 | 4.703 | ? |
| $2 \times 10^6$ | 2 | ? | ? | 8.692 | 17.500 | 4.308 |
| $*2 \times 10^6$ | 3 | ? | ? | 17.751 | 166.988 | 5.391 |
| $5 \times 10^6$ | 1 | ? | ? | 20.369 | ? | ? |
| $5 \times 10^6$ | 2 | ? | ? | 21.703 | ? | 10.493 |
| $*5 \times 10^6$ | 3 | ? | ? | 32.412 | ? | 12.406 |
| $10^7$ | 1 | ? | ? | ? | ? | ? |
| $10^7$ | 2 | ? | ? | 42.343 | ? | 21.191 |
| $*10^7$ | 3 | ? | ? | 54.738 | ? | 24.885 |
| $2 \times 10^7$ | 1 | ? | ? | ? | ? | ? |
| $2 \times 10^7$ | 2 | ? | ? | 85.005 | ? | 44.151 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $*2 \times 10^{7}$ | 3 | ? | ? | 100.162 | ? | 50.242 |
| $5 \times 10^{7}$ | 1 | ? | ? | ? | ? | ? |
| $5 \times 10^{7}$ | 2 | ? | ? | 212.805 | ? | 104.877 |
| $*5 \times 10^{7}$ | 3 | ? | ? | 242.539 | ? | 130.451 |

Table 2: Uniform vs Weighted Uniform

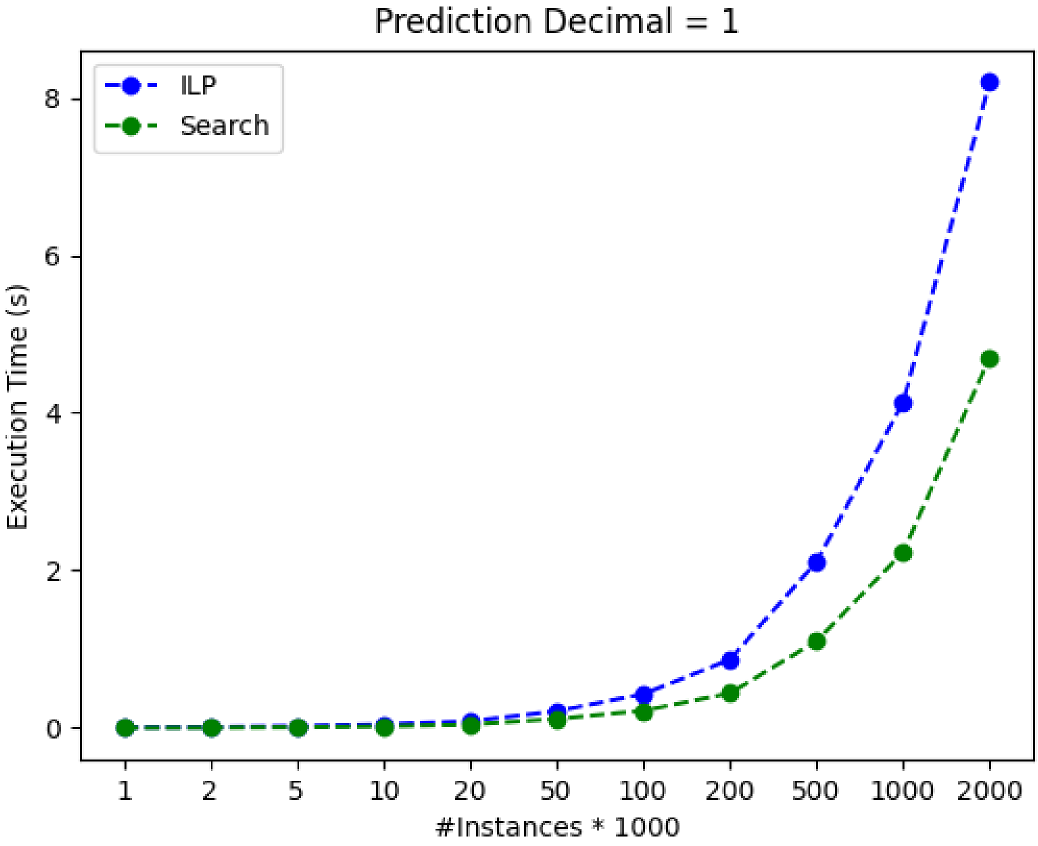
| # Instances | Prediction Decimal | ILP-1 (C++) | ILP-2 (C++) |
|---|---|---|---|
| $10^5$ | 3 | 7.351 | 3.964 |
| $2 \times 10^5$ | 3 | 8.462 | 4.856 |
| $5 \times 10^5$ | 3 | 10.771 | 6.029 |
| $10^6$ | 3 | 13.010 | 8.178 |
| $2 \times 10^6$ | 3 | 17.751 | 13.079 |
| $5 \times 10^6$ | 3 | 32.412 | 25.539 |
| $10^7$ | 3 | 54.738 | 50.199 |
| $2 \times 10^7$ | 3 | 100.162 | 96.225 |

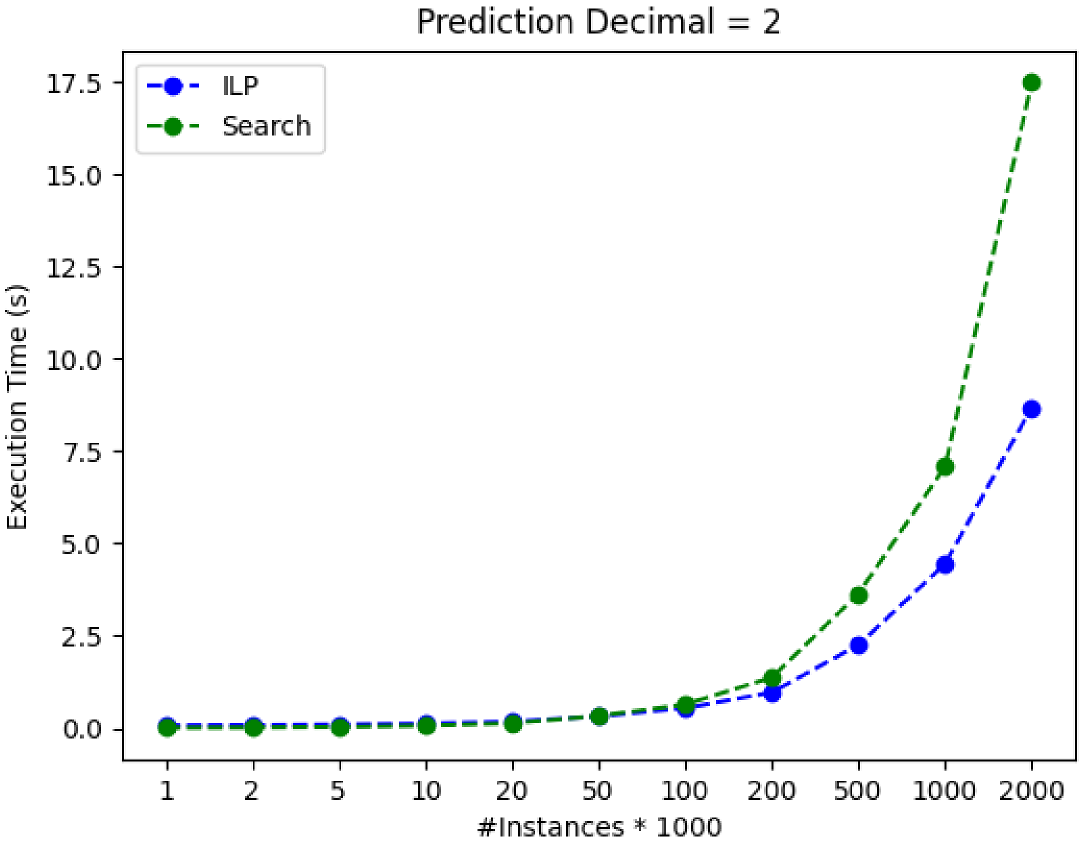
Table 3:

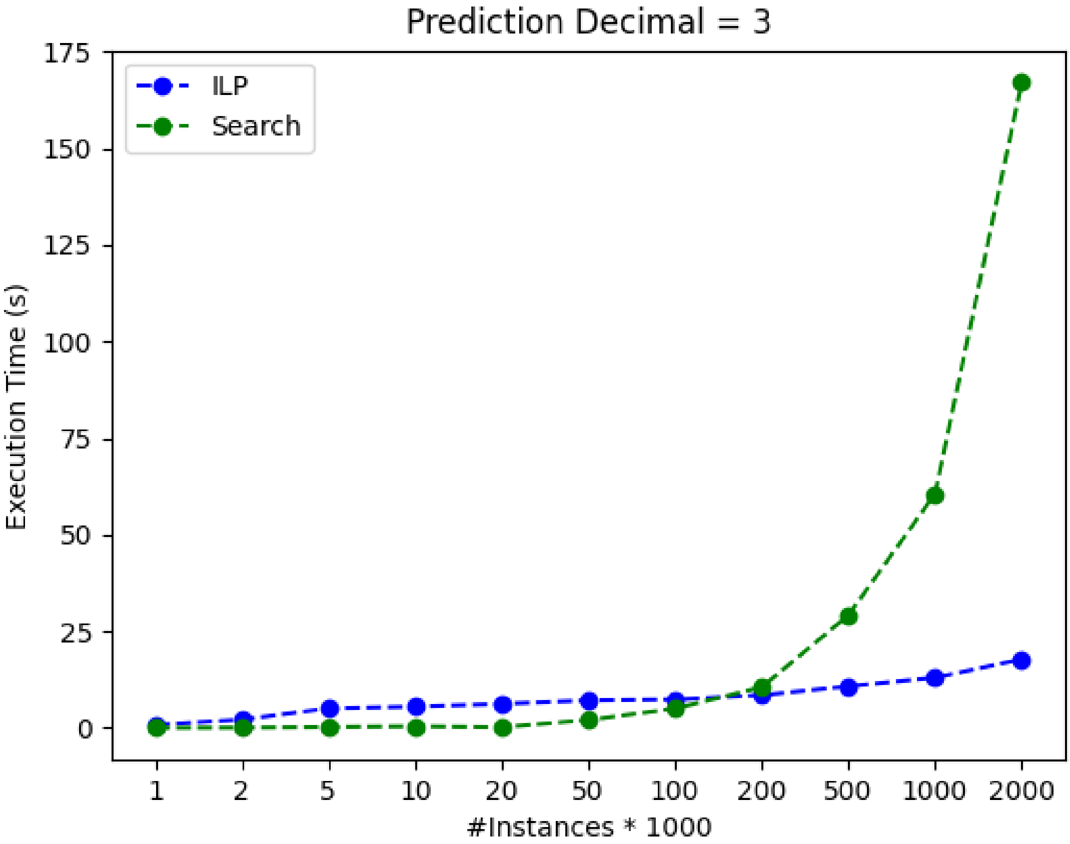
| # Instances | Prediction Decimal | ILP | Search $O(n \log n)$ | Search $O(n)$ | ILP/Search |
|---|---|---|---|---|---|
| $10^5$ | 3 | 7.351 | 0.275 | 0.101 | 72.78 |
| $2 \times 10^5$ | 3 | 8.462 | 0.506 | 0.194 | 43.61 |
| $5 \times 10^5$ | 3 | 10.771 | 1.055 | 0.501 | 21.49 |
| $10^6$ | 3 | 13.010 | 2.685 | 1.052 | 12.36 |
| $2 \times 10^6$ | 3 | 17.751 | 5.391 | 1.916 | 9.26 |
| $5 \times 10^6$ | 3 | 32.412 | 12.406 | 4.893 | 6.62 |
| $10^7$ | 3 | 54.738 | 24.885 | 9.511 | 5.75 |
| $2 \times 10^7$ | 3 | 100.162 | 50.242 | 19.419 | 5.15 |

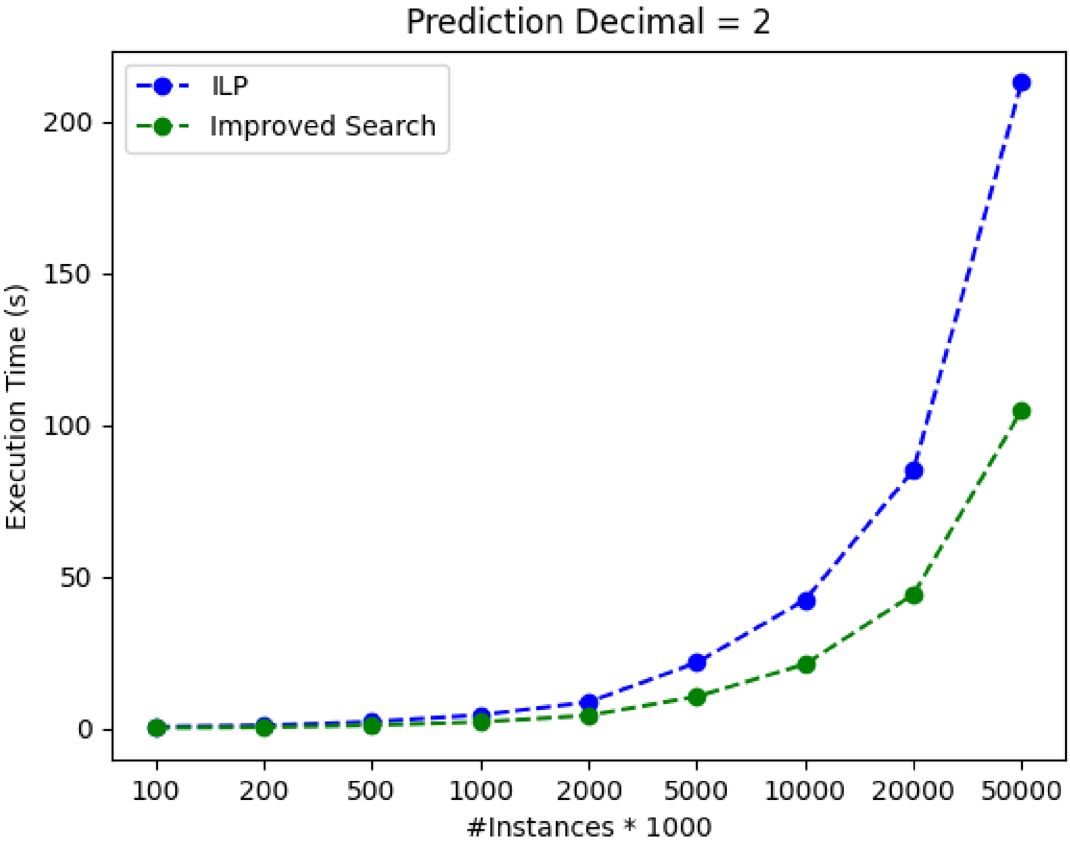
Table 4:

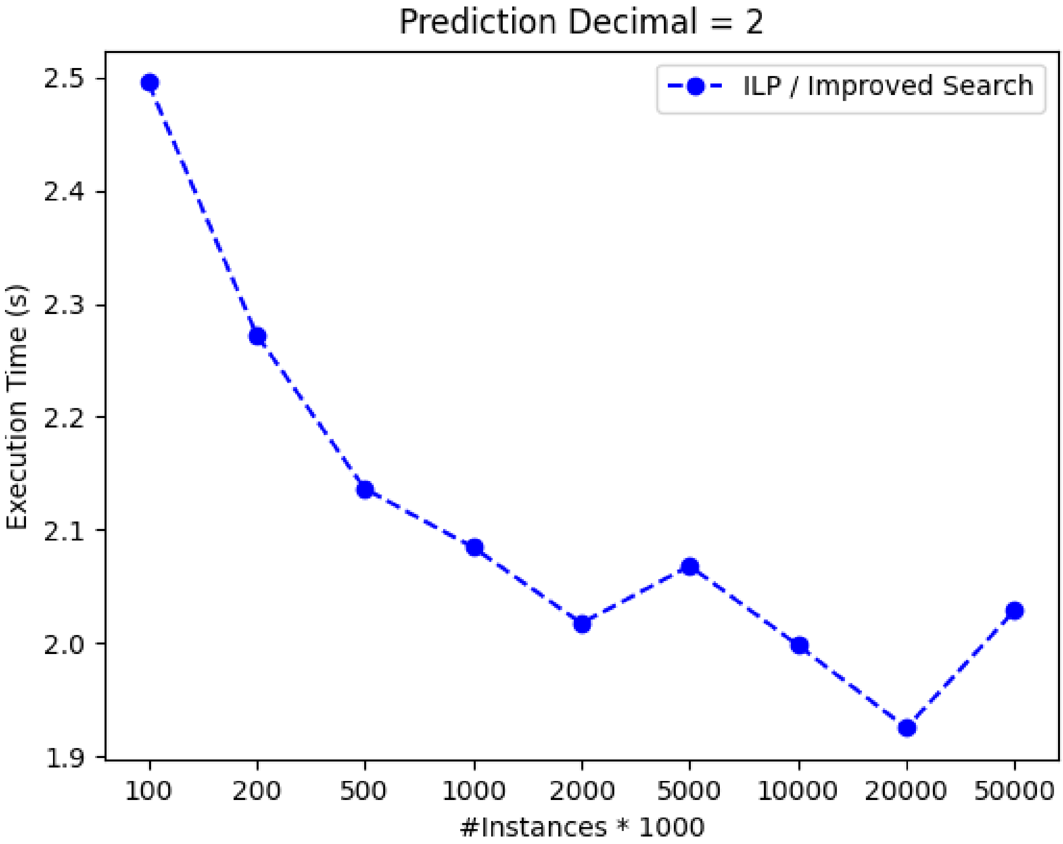
| # Instances | Prediction Decimal | ILP (GLPK) | ILP (Gurobi) |
|---|---|---|---|
| $10^5$ | 3 | 7.351 | 9.374 |
| $2 \times 10^5$ | 3 | 8.462 | 10.207 |
| $5 \times 10^5$ | 3 | 10.771 | 13.163 |
| $10^6$ | 3 | 13.010 | 16.501 |
| $2 \times 10^6$ | 3 | 17.751 | 22.740 |
| $5 \times 10^6$ | 3 | 32.412 | 39.581 |
| $10^7$ | 3 | 54.738 | 63.942 |
| $2 \times 10^7$ | 3 | 100.162 | 114.560 |

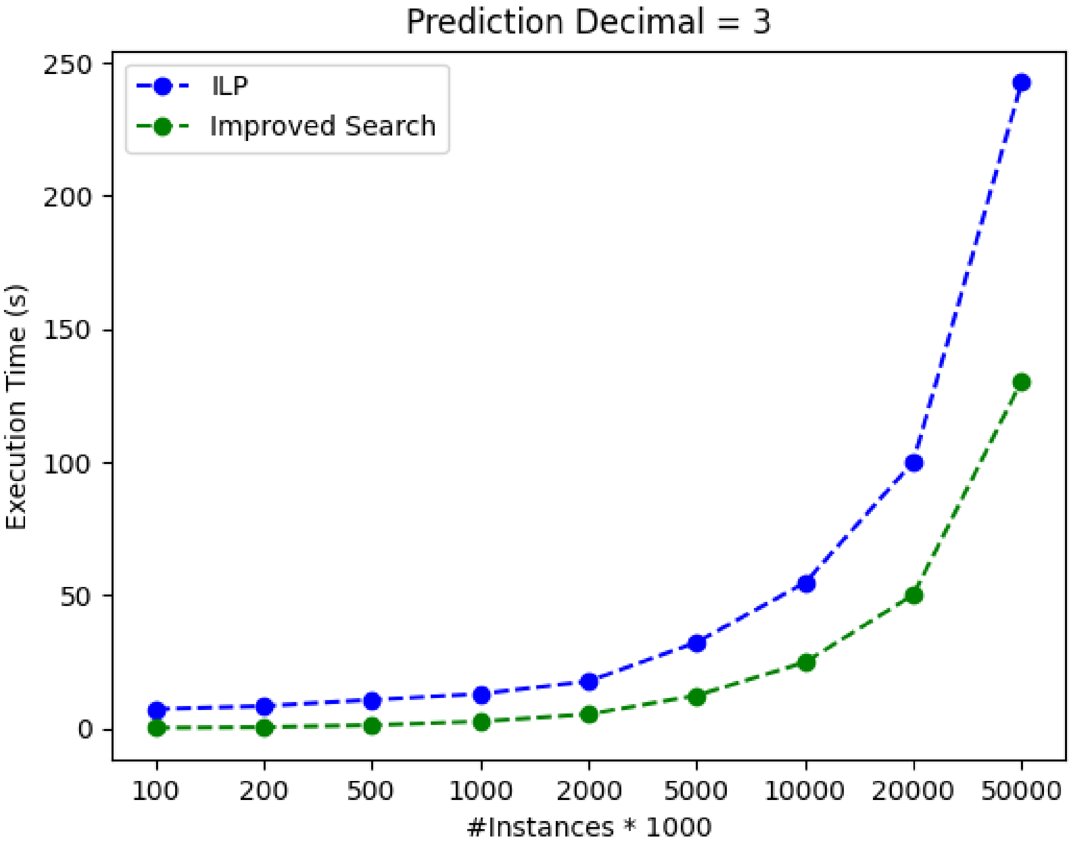

Prediction Decimal = 1
ILP
Search
Execution Time (s)
0
2
4
6
8
1
2
5
10
20
50
100
200
500
1000
2000
#Instances * 1000

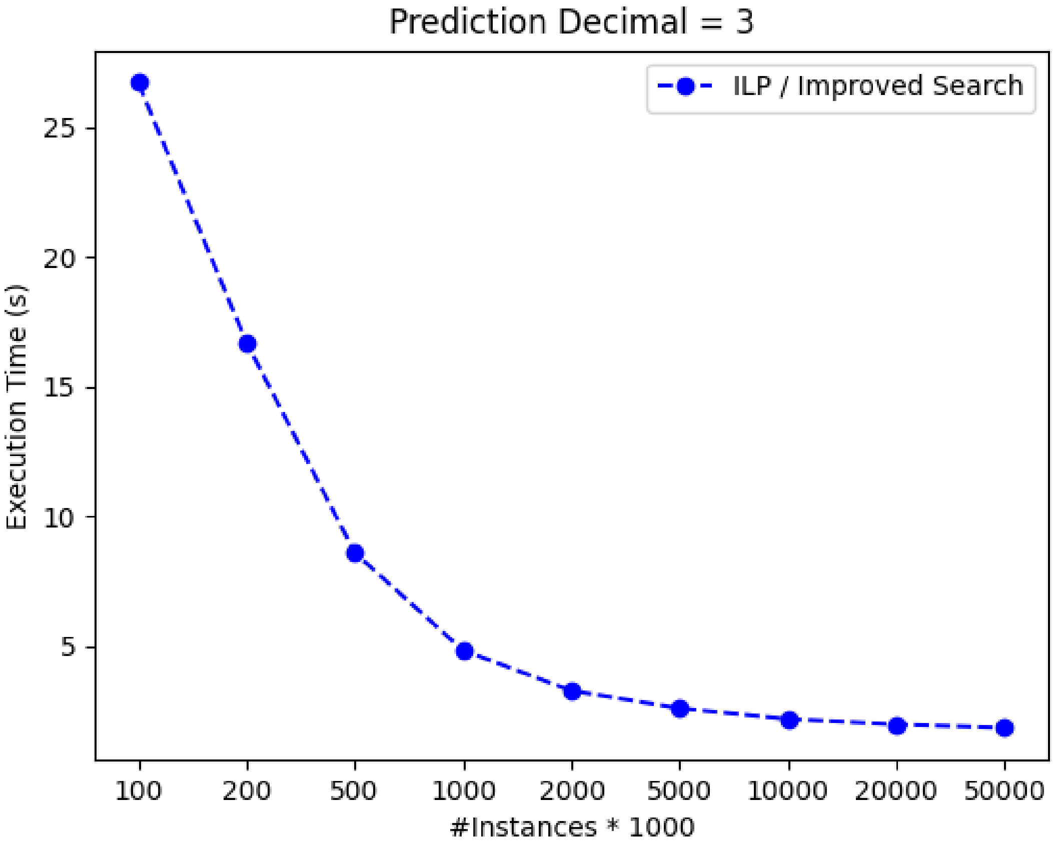

Prediction Decimal = 2
ILP
Search
Execution Time (s)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1
2
5
10
20
50
100
200
500
1000
2000
#Instances * 1000

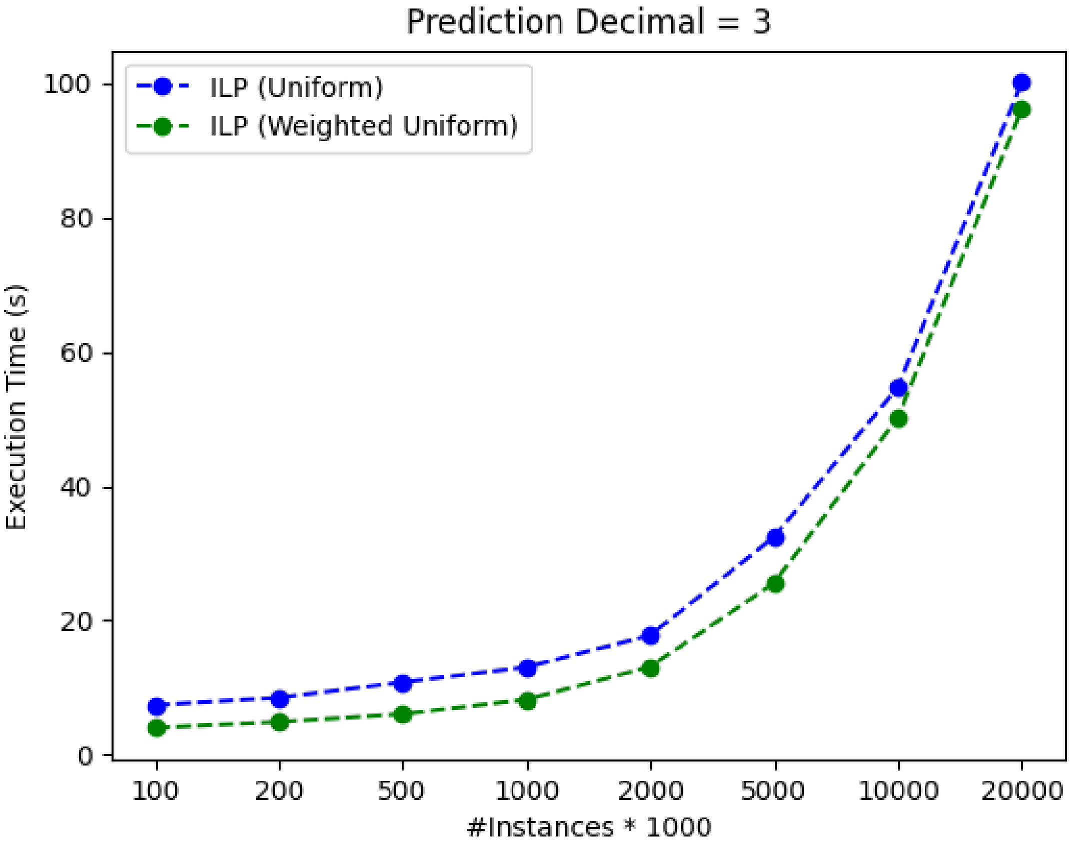

Prediction Decimal = 3
ILP
Search
Execution Time (s)
175
150
125
100
75
50
25
0
1
2
5
10
20
50
100
200
500
1000
2000
#Instances * 1000


Prediction Decimal = 2
ILP
Improved Search
Execution Time (s)
0
50
100
150
200
100
200
500
1000
2000
5000
10000
20000
50000
#Instances * 1000


Prediction Decimal = 2
ILP / Improved Search
Execution Time (s)
2.5
2.4
2.3
2.2
2.1
2.0
1.9
100
200
500
1000
2000
5000
10000
20000
50000
#Instances * 1000


Prediction Decimal = 3
ILP
Improved Search
Execution Time (s)
0
50
100
150
200
250
#Instances * 1000
100
200
500
1000
2000
5000
10000
20000
50000


Prediction Decimal = 3
ILP / Improved Search
Execution Time (s)
25
20
15
10
5
100
200
500
1000
2000
5000
10000
20000
50000
#Instances * 1000


Prediction Decimal = 3
ILP (Uniform)
ILP (Weighted Uniform)
Execution Time (s)
0
20
40
60
80
100
100
200
500
1000
2000
5000
10000
20000
#Instances * 1000

For $N = 2$:

$$\begin{aligned} \text{minimize} \quad & z_1 + z_2 \\ \text{subject to:} \quad & z_1 + \tilde{y}_1 \geq y_1 \\ & z_2 + \tilde{y}_2 \geq y_2 \\ & z_1 - \tilde{y}_1 \geq -y_1 \\ & z_2 - \tilde{y}_2 \geq -y_2 \\ & \hat{y}_1 \tilde{y}_1 + t \geq \hat{y}_1 + \epsilon \\ & \hat{y}_2 \tilde{y}_2 + t \geq \hat{y}_2 + \epsilon \\ & (\hat{y}_1 - 1)\tilde{y}_1 - t \geq -1 \\ & (\hat{y}_2 - 1)\tilde{y}_2 - t \geq -1 \end{aligned} \tag{3}$$

$$\begin{aligned} \text{minimize} \quad & wx \\ \text{subject to:} \quad & Ax \geq b \end{aligned} \tag{4}$$

$$A = \begin{array}{c} \begin{array}{ccccc} z_1 & z_2 & \tilde{y}_1 & \tilde{y}_2 & t \end{array} \\ \left(\begin{array}{cc|cc|c} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ \hline 0 & 0 & \hat{y}_1 & 0 & 1 \\ 0 & 0 & 0 & \hat{y}_2 & 1 \\ \hline 0 & 0 & \hat{y}_1 - 1 & 0 & -1 \\ 0 & 0 & 0 & \hat{y}_2 - 1 & -1 \end{array}\right)_{8\times 5} \end{array}$$

$$x = \left(\begin{array}{c} z_1 \\ z_2 \\ \hline \tilde{y}_1 \\ \tilde{y}_2 \\ \hline t \end{array}\right)_{5\times 1}$$

$$b = \left(\begin{array}{c} y_1 \\ y_2 \\ \hline -y_1 \\ -y_2 \\ \hline \hat{y}_1 + \epsilon \\ \hat{y}_2 + \epsilon \\ \hline -1 \\ -1 \end{array}\right)_{8\times 1}$$

$$A = \left(\begin{array}{cccc|cccc|c}
z_1 & z_2 & \dots & z_N & \tilde{y}_1 & \tilde{y}_2 & \dots & \tilde{y_N} & t \\
1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 \\
0 & 1 & \dots & 0 & 0 & 1 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 & 0 \\
\hline
1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & 0 \\
0 & 1 & \dots & 0 & 0 & -1 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 1 & 0 & 0 & \dots & -1 & 0 \\
\hline
0 & 0 & \dots & 0 & \hat{y}_1 & 0 & 0 & 0 & 1 \\
0 & 0 & \dots & 0 & 0 & \hat{y}_2 & 0 & 0 & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \hat{y_N} & 1 \\
\hline
0 & 0 & \dots & 0 & \hat{y}_1 - 1 & 0 & 0 & 0 & -1 \\
0 & 0 & \dots & 0 & 0 & \hat{y}_2 - 1 & 0 & 0 & -1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \hat{y_N} - 1 & -1
\end{array}\right)_{4N\times(2N+1)}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{\#\text{ class 1}}$$

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{N} y_i \tilde{y}_i & & \\
& \hat{y}_i (1 - \tilde{y}_i) < t & & \forall 1 \leq i \leq N \\
& \tilde{y}_i (\hat{y}_i - 1) \geq t - 1 & & \forall 1 \leq i \leq N \\
& \tilde{y}_i \in \{0, 1\} & & \forall 1 \leq i \leq N \\
& 0 \leq t \leq 1 & &
\end{aligned} \tag{5}$$

$$specificity = \frac{TN}{TN + FP} = \frac{TN}{\#\ \text{class}\ 0}$$

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{N} (1 - y_i)(1 - \tilde{y}_i) & & \\
& \hat{y}_i (1 - \tilde{y}_i) < t & & \forall 1 \le i \le N \\
& \tilde{y}_i (\hat{y}_i - 1) \ge t - 1 & & \forall 1 \le i \le N \\
& \tilde{y}_i \in \{0, 1\} & & \forall 1 \le i \le N \\
& 0 \le t \le 1 & &
\end{aligned} \tag{6}$$

$$\max \quad \alpha \times \text{accuracy} + \beta \times \text{recall} + \gamma \times \text{specificity}$$

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N} \alpha z_i - \beta y_i \tilde{y}_i - \gamma (1 - y_i)(1 - \tilde{y}_i) & \\
\text{subject to:} \quad & z_i \geq y_i - \tilde{y}_i & \forall 1 \leq i \leq N \\
& z_i \geq \tilde{y}_i - y_i & \forall 1 \leq i \leq N \\
& \hat{y}_i (1 - \tilde{y}_i) < t & \forall 1 \leq i \leq N \\
& \tilde{y}_i (\hat{y}_i - 1) \geq t - 1 & \forall 1 \leq i \leq N \\
& \tilde{y}_i \in \{0, 1\} & \forall 1 \leq i \leq N \\
& 0 \leq t \leq 1 &
\end{aligned} \tag{7}$$

$$precision = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{N} y_i \tilde{y}_i}{\sum_{i=1}^{N} \tilde{y}_i}$$

$$\begin{aligned}
\text{maximize} \quad & \frac{\sum_{i=1}^{N} y_i \tilde{y}_i}{\sum_{i=1}^{N} \tilde{y}_i} & \\
\text{subject to:} \quad & t + \hat{y}_i \tilde{y}_i > \hat{y}_i & \forall 1 \leq i \leq N \\
& (\hat{y}_i - 1)\tilde{y}_i - t \geq 5 - 1 & \forall 1 \leq i \leq N \\
& \tilde{y}_i \in \{0, 1\} & \forall 1 \leq i \leq N \\
& 0 \leq t \leq 1 &
\end{aligned} \tag{8}$$

$$\text{maximize} \qquad \frac{c^T x + \alpha}{d^T x + \beta} \equiv \mathbf{R}(x)$$

$$(1.1) \qquad \text{subject to} \qquad Ax \leq b$$

$$x \geq 0,$$

where A is an m × n matrix and b is an m × 1 vector so that the two sets of constants for the constraints are related by the n × 1 vector of variables, x. Similarly, $c^T$ and $d^T$ are transposes of the n × 1 vectors of coefficients, c and d, respectively, while $\alpha$ and $\beta$ are arbitrary scalar constants.

It is assumed, unless otherwise noted, that the constraints of (1.1) are regular[1] so that the solution set

$$(1.2) \qquad \mathbf{X} \equiv \{x; Ax \leq b,\ \ x \geq 0\}$$

is nonempty and bounded.

The following transformation of variables is now introduced:

$$(2.1) \qquad y \equiv t\,x$$

where $t \geq 0$ is to be chosen so that

$$(2.2) \qquad d^T y + \beta t = \gamma$$

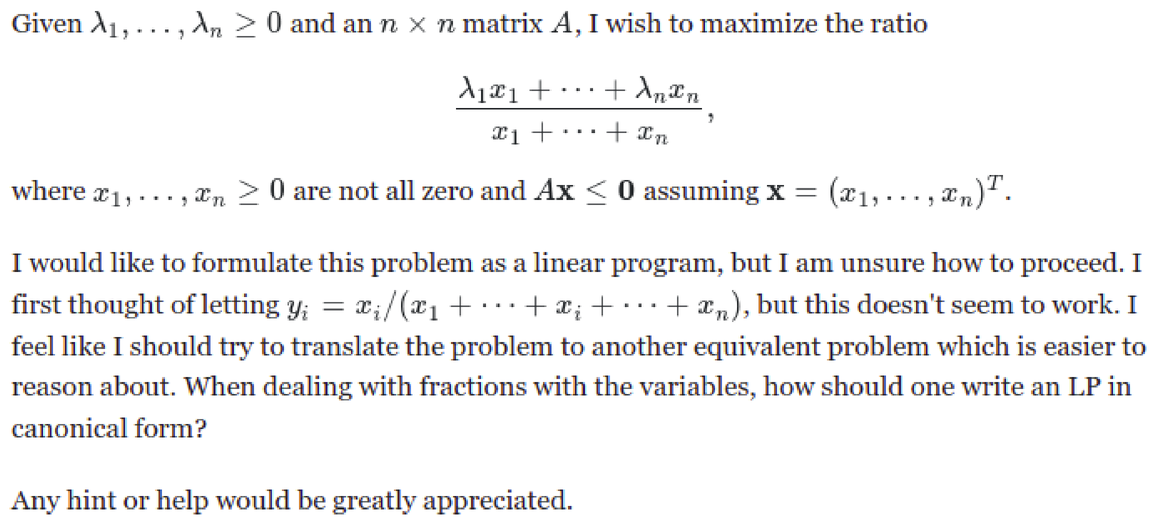
where $\gamma \neq 0$ is a specified number. On multiplying numerator and denominator and the system of inequalities in (1.1) by t and taking (2.2) into account, we obtain the linear programming problem

$$\text{maximize} \qquad c^T y + \alpha t \equiv L(y,t)$$

$$(3) \qquad \text{subject to} \qquad Ay - bt \leq 0$$

$$d^T y + \beta t = \gamma$$

$$y,t \geq 0 .$$



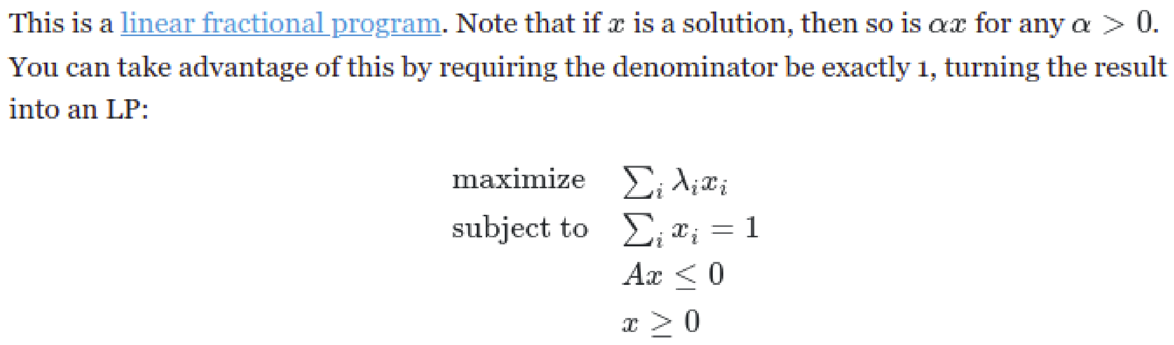
Given $\lambda_1, \ldots, \lambda_n \geq 0$ and an $n \times n$ matrix $A$, I wish to maximize the ratio

$$\frac{\lambda_1 x_1 + \cdots + \lambda_n x_n}{x_1 + \cdots + x_n},$$

where $x_1, \ldots, x_n \geq 0$ are not all zero and $A\mathbf{x} \leq \mathbf{0}$ assuming $\mathbf{x} = (x_1, \ldots, x_n)^T$.

I would like to formulate this problem as a linear program, but I am unsure how to proceed. I first thought of letting $y_i = x_i/(x_1 + \cdots + x_i + \cdots + x_n)$, but this doesn't seem to work. I feel like I should try to translate the problem to another equivalent problem which is easier to reason about. When dealing with fractions with the variables, how should one write an LP in canonical form?

Any hint or help would be greatly appreciated.



This is a linear fractional program. Note that if $x$ is a solution, then so is $\alpha x$ for any $\alpha > 0$. You can take advantage of this by requiring the denominator be exactly 1, turning the result into an LP:

$$\begin{array}{ll} \text{maximize} & \sum_i \lambda_i x_i \\ \text{subject to} & \sum_i x_i = 1 \\ & Ax \leq 0 \\ & x \geq 0 \end{array}$$

And the mathematical formulation of an LFP is as follows:

$$\text{Maximize} \quad Z = \frac{cx+\alpha}{dx+\beta} \tag{5}$$

$$\text{subject to} \quad Ax \le b \tag{6}$$

$$x \ge 0 \tag{7}$$

where, $A = \left(a_1, a_2, \ldots\ldots, a_m, a_{m+1}, \ldots a_n\right)$ is a $m \times n$ matrix, $b \in R^m$, $x, c, d \in R^n$, $\alpha, \beta \in R$.

It is assumed that the feasible region $S = \left\{x \in R^n : Ax \le b, x \ge 0\right\}$ is nonempty and bounded and the denominator $dx + \beta \ne 0$.

Now, If $d = 0$ and $\beta = 1$ then the LFP in (5) to (7) becomes an LP problem. That is (5) can be written as:

$$\begin{aligned} &\text{Maximize} \quad Z = cx + \alpha \\ &\text{subject to} \quad Ax \le b;\ x \ge 0. \end{aligned}$$

This is why we say that LFP is a generalization of an LP in (1) to (3). There are also a few cases when the LFP can be replaced with an appropriate LP. These cases are discussed as follows:

**Case 1:**

If $d = 0$ and $\beta \ne 1$ in (5), then Z becomes a linear function

$$Z = \frac{c}{\beta}x + \frac{\alpha}{\beta} = \frac{Z^1}{\beta} \quad \text{, where } Z^1 = cx + \alpha \text{ is a linear function.}$$

In this case Z may be substituted with $\frac{Z^1}{\beta}$ corresponding to the same set of feasible region S. As a result the LFP becomes an LP.

**Case 2:**

If $c = 0$ in (5) then $Z = \frac{\alpha}{dx+\beta} = \frac{\alpha}{Z^2}$, where $Z^2 = dx + \beta$ is a linear function.

In this case, Z becomes linear on the same set of feasible solution S. Therefore the LFP results in an LP with the same feasible region S.

**Case 3:**

If $c = (c_1, c_2, \ldots\ldots\ldots\ldots c_n)$, $d = (d_1, d_2, \ldots\ldots\ldots\ldots d_n)$ are linearly dependent, there exists $\mu \ne 0$ such that $c = \mu d$, then $Z = \frac{\mu dx + \alpha}{dx + \beta} = \mu + \frac{\alpha - \mu\beta}{dx+\beta}$ Z

(i) if $\alpha - \mu\beta = 0$, then $Z = \mu$ is a constant.

(ii)if $\alpha - \mu\beta > 0$ or $\alpha - \mu\beta < 0$ i.e., if $\alpha - \mu\beta \neq 0$ then Z becomes a linear function. Therefore the LFP becomes an LP with the same feasible region S.

If $c \neq 0$ and $d \neq 0$ then one has to find a new way to convert the LFP into an LP. Assuming that the feasible region

$$S = \{x \in R^n : Ax \leq b, x \geq 0\}$$

is nonempty and bounded and the denominator $dx + \beta > 0$, we develop a method which converts an LFP of this type to an LP.