Using Simulation and Symbolic Computing in Linear Programming

SILVIU GUIASU
Department of Mathematics and Statistics, York University
4700 Keele Street, Toronto, Ontario M3J 1P3
CANADA

Abstract: - N. Karmarkar ([1]) has shown how a linear program with inequality constraints may be transformed into a normed linear program with equality constraints whose feasible solutions are probability distributions. In looking for the optimum of normed linear programs he used standard optimization techniques from multivariate calculus. The objective of this paper is to develop the probabilistic search for the optimum solution of a normed linear program, proposed in [2], using Pearson's ([3]) χ^2-indicator and the symbolic computing. The approximation of the optimum solution based on heuristic simulation is also discussed.

Key-words: - simulation, symbolic computing, linear programs, minimum χ^2 estimation, probabilistic search for optimum.

1 Introduction

The main objective of the paper is to show how simulation and symbolic computing may be used in approximating the optimum solution of linear programs. Different variants of the Simplex Algorithm are currently used for solving linear programs. Basically, this algorithm starts from a corner point of the convex feasible space and moves, step-by-step, to a neighbouring corner point until, eventually, the optimum corner point is found. Nobody has proved thus far that such a search strategy is optimum. In 1984, Karmarkar [1] has proposed another strategy, looking for intermediary solutions inside the feasible space and not on its frontier as the Simplex Algorithm does. In presenting his method, he first ingeniously showed that each linear program with inequality constraints may be put in a normed form, which is a linear program with equality constraints whose feasible solutions are probability distributions, namely:

$$\min_x z = \sum_{j=1}^{n} c_j \, x_j \tag{1}$$

subject to:

$$\textstyle\sum_{j=1}^n a_{ij} x_j = 0, \quad (i = 1, \ldots, m), \tag{2}$$

$$\textstyle\sum_{j=1}^n x_j = 1, \tag{3}$$

$$x_j \geq 0, \qquad (j = 1, \ldots, n), \tag{4}$$

for which the point $(1/n, \ldots, 1/n)$ is a feasible solution and the optimum z-value equals 0. Surprisingly, any linear program may be put in this form and, as shown by the above constraints, the feasible solutions $x = (x_1, \ldots, x_n)$ are probability distributions with n components. However, Karmarkar's method of solving the normed linear program does not use any kind of probabilistic approach and relies on standard optimization techniques from multivariate calculus mainly based on the manipulation of the gradient. Numerical results obtained when dealing with large numbers of variables and constraints have shown that Karmarkar's method is faster than the traditional Simplex Algorithm. However, it is quite laborious. Details about Karmarkar's method may be found in Winston ([4], pp.181-182, 591-599).

The objective of this paper is to discuss a probabilistic search strategy for the optimum of normed linear programs using the classic Pearson's χ^2 indicator and to use, alternatively, heuristic simulation for approximating the optimum solution.

2 A probabilistic search for optimum

Pearson's χ^2 indicator from mathematical statistics, introduced in [3], namely,

$$\chi^2(x:y) = \sum_{j=1}^{n} \frac{(x_j - y_j)^2}{y_j}$$

measures how different a new probability distribution $x = (x_1, \ldots, x_n)$ is with respect to a given probability distribution $y = (y_1, \ldots, y_n)$. Obviously, $\chi^2(x:y) = 0$ if and only if $x_j = y_j$, for all $j = 1, \ldots, n$. Using the classic Lagrange multipliers method it is easy to see that the optimum solution $x = (x_1, \ldots, x_n)$ of the quadratic program:

$$\min_x \chi^2(x:y), \tag{5}$$

subject to the constraint (3) and:

$$\sum_{j=1}^{n} d_{ij} x_j = f_i, \quad (i = 1, \ldots, m),$$

is given by:

$$x_j = \left[1 + \sum_{i=1}^{m} \frac{\alpha_i}{2} \left(d_{ij} - \sum_{k=1}^{n} d_{ik} y_k\right)\right] y_j, \tag{6}$$

for $j = 1, \ldots, n$, where $(\alpha_1, \ldots, \alpha_m)$ is the solution of the linear system of equations:

$$\sum_{\ell=1}^{m} \frac{\alpha_\ell}{2} \sum_{j=1}^{n} \left(d_{\ell j} - \sum_{k=1}^{n} d_{\ell k} y_k\right) d_{ij} y_j =$$

$$= f_i - \sum_{j=1}^{n} d_{ij} y_j, \qquad (i = 1, \ldots, m). \tag{7}$$

It is easy to solve the linear system (7), but the components of the optimum solution (6) are not all necessarily nonnegative.

Going back to the normed linear program given in Section 1, we define the following search algorithm where at each step we are looking for the closest probability distribution to the current solution subject to a certain level of the objective function of the initial normed linear program. Subsequently, this level is minimized by keeping the new solution feasible:

Initial Step: Find the closest probability distribution $x = (x_1, \ldots, x_n)$ to the starting uniform distribution:

$$y = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right),$$

by solving (5), subject to the initial constraints (2), (3) and the new constraint:

$$\sum_{j=1}^{n} c_j x_j = z, \qquad (z \geq 0). \tag{8}$$

As the solution obviously depends on the level z of the initial objective function of the normed linear program, find the minimum value of z such that the corresponding solution $x = (x_1, \ldots, x_n)$ thus obtained also satisfies the initial constraint (4).

Intermediary Step: Find the closest probability distribution $x = (x_1, \ldots, x_n)$ to the current solution $y = (y_1, \ldots, y_n)$, by solving (5), subject to the constraints (2), (3), and (8). As the solution obviously depends linearly on the level z of the initial objective function of the normed linear program, find the minimum nonnegative value of z for which the corresponding solution $x = (x_1, \ldots, x_n)$ also satisfies the nonnegativity constraint (4).

Stop: When no improvement (i.e., decrease) of z may be found.

Analitically, by solving (5), subject to the constraints (2), (3), and (8), we get:

$$x_j = \left(1 + \frac{\alpha}{2} c_j + \sum_{i=1}^{m} \frac{\beta_i}{2} a_{ij} + \frac{\gamma}{2}\right) y_j, \tag{9}$$

for $j = 1, \ldots, n$, where the Lagrange multipliers $\alpha, \beta_1, \ldots, \beta_m, \gamma$ are determined by introducing (9) into the constraints (2), (3), (8). The solution will essentially depend on the level $z \geq 0$, whose minimum value is found such that the nonnegativity constraint (4) is also satisfied.

3 Example

Let us take the following normal linear program:

$$\max_x z = 3x_1 + x_2 \tag{10}$$

subject to:

$$\begin{aligned} 2x_1 - x_2 &\leq 2 && (11) \\ x_1 + 2x_2 &\leq 5 && (12) \\ x_j &\geq 0, \quad (j = 1, 2) && (13) \end{aligned}$$

The normed form of this normal linear program is (Winston [4], pp.597-599):

$$\min_p p_{11}$$

subject to:

$$\begin{aligned} f_2 &= 3p_1 + p_2 - 2p_3 - 5p_4 + 3p_{11} = 0 \\ f_3 &= 2p_1 - p_2 + p_5 - 2p_{10} = 0 \\ f_4 &= p_1 + 2p_2 + p_6 - 5p_{10} + p_{11} = 0 \\ f_5 &= 2p_3 + p_4 - p_7 - 3p_{10} + p_{11} = 0 \\ f_6 &= -p_3 + 2p_4 - p_8 - p_{10} + p_{11} = 0 \\ f_7 &= \sum_{i=1}^{9} p_i - 80p_{10} + 71p_{11} = 0 \\ f_8 &= \sum_{i=1}^{11} p_i - 1 = 0 \\ & p_i \geq 0, \quad (i = 1, \ldots, 11) \end{aligned}$$

where the initial decision variables are $x_1 = 81p_1$ and $x_2 = 81p_2$. Let us notice that the uniform probability distribution $p_i = 1/11$, $(i = 1, \ldots, 11)$ is a feasible solution of the normed linear program. Denote by $f_1 = p_{11} - z$. Let $q = (q_1, \ldots, q_{11})$ be a given, initial probability distribution. We want to solve the quadratic program:

$$\min_p \sum_{i=1}^{11} \frac{(p_i - q_i)^2}{q_i}$$

subject to:

$$f_j = 0, \qquad (j = 1, \ldots, 8). \qquad (14)$$

The powerful symbolic computing packages of MATHEMATICA ([5]) may be used for implementing the steps just discussed. The commands for implementing one iteration towards approximating the optimum solution are given below. Inside the program, $a_1, \ldots, a_8$ are the corresponding Lagrange multipliers and $s1, \ldots, s11$ is the solution obtained at the end of such an iteration.

For our example, using the version 2.1 of MATHEMATICA, an arbitrary iteration contains the following commands:

```
math
In[1]:= p1 = q1 * (1 + 3a2 + 2a3 + a4 + a7 + a8)
In[2]:= p2 = q2 * (1 + a2 - a3 + 2a4 + a7 + a8)
In[3]:= p3 = q3 * (1 - 2a2 + 2a5 - a6 + a7 + a8)
In[4]:= p4 = q4 * (1 - 5a2 + a5 + 2a6 + a7 + a8)
In[5]:= p5 = q5 * (1 + a3 + a7 + a8)
In[6]:= p6 = q6 * (1 + a4 + a7 + a8)
In[7]:= p7 = q7 * (1 - a5 + a7 + a8)
In[8]:= p8 = q8 * (1 - a6 + a7 + a8)
In[9]:= p9 = q9 * (1 + a7 + a8)
In[10]:= p10 = q10 * (1 - 2a3 - 5a4 - 3a5-
        a6 - 80a7 + a8)
In[11]:= p11 = q11 * (1 + a1 + 3a2 + a4 + a5+
        a6 + 71a7 + a8)
In[12]:= f1 = p11 - z
In[13]:= f2 = 3p1 + p2 - 2p3 - 5p4 + 3p11
In[14]:= f3 = 2p1 - p2 + p5 - 2p10
In[15]:= f4 = p1 + 2p2 + p6 - 5p10 + p11
In[16]:= f5 = 2p3 + p4 - p7 - 3p10 + p11
In[17]:= f6 = -p3 + 2p4 - p8 - p10 + p11
In[18]:= f7 = p1 + p2 + p3 + p4 + p5 + p6 + p7+
        p8 + p9 - 80p10 + 71p11
In[19]:= f8 = p1 + p2 + p3 + p4 + p5 + p6 + p7+
        p8 + p9 + p10 + p11 - 1
In[20]:= Solve[{f1 == 0, f2 == 0, f3 == 0,
        f4 == 0, f5 == 0, f6 == 0,
        f7 == 0, f8 == 0},
        {a1, a2, a3, a4, a5, a6, a7, a8}]
In[21]:= p1 = p1/.%20
In[22]:= Expand[N[%]]
In[23]:= p2 = p2/.%20
In[24]:= Expand[N[%]]
In[25]:= p3 = p3/.%20
In[26]:= Expand[N[%]]
In[27]:= p4 = p4/.%20
In[28]:= Expand[N[%]]
In[29]:= p5 = p5/.%20
In[30]:= Expand[N[%]]
In[31]:= p6 = p6/.%20
In[32]:= Expand[N[%]]
In[33]:= p7 = p7/.%20
In[34]:= Expand[N[%]]
In[35]:= p8 = p8/.%20
In[36]:= Expand[N[%]]
In[37]:= p9 = p9/.%20
In[38]:= Expand[N[%]]
In[39]:= p10 = p10/.%20
In[40]:= Expand[N[%]]
```

In[41]:= $p11 = p11/.\%20$
In[42]:= Expand[N[%]]

Out[21],...,Out[42] give the expressions of $p1, \ldots, p11$ as linear functions of z, respectively. Let b be the smallest numerical value such that for $z \geq b$ all $p1, \ldots, p11$ are nonnegative. We continue our MATHEMATICA session with the commands:
In[43]:= $s1 = p1/.z- > b$
In[44]:= $s2 = p2/.z- > b$
In[45]:= $s3 = p3/.z- > b$
In[46]:= $s4 = p4/.z- > b$
In[47]:= $s5 = p5/.z- > b$
In[48]:= $s6 = p6/.z- > b$
In[49]:= $s7 = p7/.z- > b$
In[50]:= $s8 = p8/.z- > b$
In[51]:= $s9 = p9/.z- > b$
In[52]:= $s10 = p10/.z- > b$
In[53]:= $s11 = p11/.z- > b$
Quit

After this iteration, the current solution is $s1, \ldots, s11$. For a new iteration we go back to step In[1] by taking, successively, the current solution $s1, \ldots, s11$ to be the new intial starting solution $q1, \ldots, q11$. The last value of z (ideally zero!) arrived at in the last iteration performed gives the error of approximation of the optimum solution. The first iteration starts from the feasible solution $q1 = 1/11, \ldots, q11 = 1/11$.

In our example, using the version 2.1 of MATHEMATICA, taking $q1 = 1/11, \ldots, q11 = 1/11$, at the initial step, we get:

$$\begin{aligned}
a1 &= -79.0446 + 869.49\,z \\
a2 &= 6.79012 - 74.6914\,z \\
a3 &= -8.55556 + 94.111\,z \\
a4 &= -11.43 + 125.73\,z \\
a5 &= 7.3107 - 80.4177\,z \\
a6 &= 9.07613 - 99.9374\,z \\
a7 &= 0.641289 - 7.05418\,z \\
a8 &= 7.18587 - 79.0446\,z
\end{aligned}$$

$$\begin{aligned}
p1 &= 0.0596708 + 0.343621\,z \\
p2 &= 0.119342 - 0.312757\,z \\
p3 &= 0.0720165 + 0.207819\,z \\
p4 &= 0.0308642 + 0.660494\,z \\
p5 &= 0.0246914 + 0.728395\,z \\
p6 &= -0.236626 + 3.60288\,z \\
p7 &= 0.13786 - 0.516461\,z \\
p8 &= -0.0226337 + 1.24897\,z \\
p9 &= 0.802469 - 7.82716\,z \\
p10 &= 0.0123457 + 0.864198\,z \\
p11 &= z
\end{aligned}$$

All p_i, $(i = 1, \ldots, 11)$, are nonnegative if $z \geq 0.0656769$. Taking this lower bound in the commands:

$$si = pi/.\,z- > 0.0656769, \quad (i = 1, \ldots, 11),$$

we get the current solution:

$$\begin{array}{rl}
s1 = 0.0822388, & s2 = 0.0988006, \\
s3 = 0.0856654, & s4 = 0.0742434, \\
s5 = 0.0725301, & s6 = 5.18313 \times 10^{-7}, \\
s7 = 0.103941, & s8 = 0.0593948, \\
s9 = 0.288405, & s10 = 0.0691035, \\
s11 = 0.0656769. &
\end{array}$$

Taking these values of $s1, \ldots, s11$ as the reference probability distribution $q1, \ldots, q11$, we repeat the steps mentioned above.

After a couple of more iterations we get $z = 0.000109682$ and the solution:

$$\begin{array}{rl}
s1 = 0.0221340 & s2 = 0.0199793, \\
s3 = 0.0124185, & s4 = 0.0123747, \\
s5 = 0.0005923, & s6 = 9.22242 \times 10^{-15}, \\
s7 = 1.73967 \times 10^{-15}, & s8 = -3.81739 \times 10^{-17}, \\
s9 = 0.919951, & s10 = 0.0124405, \\
s11 = 0.000109682. &
\end{array}$$

If at the end of this last iteration we take $z = 0$ we get the solution:

$$\begin{array}{rl}
s1 = 0.0222222 & s2 = 0.0197531, \\
s3 = 0.0123457, & s4 = 0.0123457, \\
s5 = 9.97295 \times 10^{-10}, & s6 = -3.75079 \times 10^{-16}, \\
s7 = -5.54053 \times 10^{-10}, & s8 = 2.18306 \times 10^{-16}, \\
s9 = 0.920982, & s10 = 0.0123457, \\
s11 = 0. &
\end{array}$$

Let us mention that the exact optimum solution is:

$$\begin{aligned} s1 &= 0.0222222 & s2 &= 0.01975309, \\ s3 &= 0.01234568, & s4 &= 0.01234568, \\ s5 &= 0, & s6 &= 0, \\ s7 &= 0, & s8 &= 0, \\ s9 &= 0.92098767, & s10 &= 0.01234568, \\ s11 &= 0, & & \end{aligned}$$

corresponding to the optimum decision values $x_1 = 81s1 = 9/5 = 1.8$, $x_2 = 81s2 = 8/5 = 1.6$ and the optimum dual values $y_1 = y_2 = 1$ of the initial normal linear program.

The normed form of a linear program involves more variables than the initial normal linear program but it contains only equality constraints, the feasible space contains only probability distributions, the uniform distribution is feasible and may be always taken as the starting feasibile solution, and the optimum objective value is always zero.

4 Approximation based on heuristic simulation

In order to approximate the optimum solution of a linear program we may adopt the maximum uncertainty strategy used by the general Monte Carlo method in simulation. As the discrete uniform distribution is the most uncertain, or unbiased, probability distribution when no restrictions are imposed, we cover as uniformly as possible the feasible space generated by the constraints and select those solutions for which the objective function is most improved. The main advantage of the optimization based on heuristic simulation is that it may be applied even when the objective functions and/or the constraints are not necessarily linear. For performig such a heuristic simulation the computer package MINITAB on WINDOWS proves to be efficient and user friendly.

Going back to the example (10)-(13) from the previous Section, the MINITAB 11 on WINDOWS session for approximating its optimum solution is:

Using any text editor, for instance Notepad, we create a text file on a diskette, named bebe.mtb. Its content is:

```
random k1 c1;
uniform 0 2.5.
random k1 c2;
uniform 0 2.5.
let c3=3*c1+c2
let c4=2*c1-c2-2
let c5=c1+2*c2-5
code(-10000:0)1 (0:10000)0 c4 c5 c6 c7
let c8=c6*c7
copy c1 c2 c3 c9 c10 c11;
use c8=1.
let k2=max(c11)
code(k2:k2)1 (-10000:k2)0 (k2:10000)0 c11 c12
copy c9 c10 c13 c14;
use c12=1.
let k3=mean(c13)
let k4=mean(c14)
let c15(k5)=k2
let c16(k5)=k3
let c17(k5)=k4
let k5=k5+1
```

The diskette is put in drive a: and we access MINITAB on WINDOWS, successively clicking the mouse on: Session, Editor, and Enable Command Language, followed by typing:

```
let k1=3000
let k5=1
noecho
execute ”a:bebe” 50
let k6=max(c15)
code(k6:k6)1 (-10000:k6)0 (k6:10000)0 c15 c18
copy c16 c17 c19 c20;
use c18=1.
print k6 c19 c20
stop
```

Generalities about MINITAB may be found in [6]. In the above program: c1 and c2 contain the values of the variables x_1, and x_2, respectively, uniformly distributed in $[0, 2.5]$, therefore compatible with (11)-(13); c3 contains the values of the objective function (10); c4 and c5 contain the corresponding value of the constraints (11) and (12), respectively; k1 is the number of solutions (x_1, x_2) randomly selected in one run of the simulation; any solution (x_1, x_2) randomly generated gets the value 1 in c8 if it is feasible and the value 0 if it is not; c9, c10, and c11 contain only those values of x_1, x_2, and z that correspond to the feasible solutions generated; k2 is the maximum objective value for the feasible solutions generated; c11 and c12 contain those feasible values of x_1 and x_2 for which the corresponding objective value is maximum; k3 and k4 are the mean values of the decision variables x_1 and x_2 for which the corresponding objective value is maximum; k5 is a counter; the values of k2, k3, and k4 are stored as the k5-th component of columns c15, c16, and c17, respectively. The subprogram bebe.mtb, randomly generating 3000 solutions (x_1, x_2), is executed 50 times. The constant k6 contains the maximum of the maximum objective values obtained in those 50 runs, whereas columns c15 and c16 contain the mean values of the decision variables x_1 and x_2, respectively, for which the corresponding objective value is equal to k6. The approximation of the optimum solution of the linear program (10)-(13) obtained at the end of the 50 executions of the subprogram bebe.mtb is given by k6, c19, and c20. One such a MINITAB session gave the results: $z^* = 6.98442$, in k6, $x_1^* = 1.79387$, in c19, and $x_2^* = 1.60282$, in c20. The exact optimum solution of the linear program (10)-(13) is: $z^* = 7$, $x_1^* = 1.8$, and $x_2^* = 1.6$.

It is a sound strategy to shrink the domain of possible values of the decision varaibles, by changing the corresponding scales of values. Even if, mathematically, the intervals $[0, 1]$ and $[0, 2]$ have the same cardinal, it is obvious that by using 1000 random numbers from each of them, we cover better the first interval.

5 Conclusion

The well known Simplex Algorithm is looking for the optimum solution of a linear program by examining only the corner points of the feasible polygonal space. N. Karmarkar has proposed a new approach, moving towards the optimum point inside the feasible space, using optimization methods from multivariate calculus. As every linear program with bounded feasible space may be put in a normed form, which means a linear program with the optimum objective value equal to zero, with equality constraints, whose feasible solutions are probability distributions, and for which the uniform distribution is always feasible, an approach involving probabilistic search techniques is obviously needed. This paper contains some results about approximating the optimum solution of linear programs with bounded feasible space by using both the Monte Carlo Method from heuristic simulation (back to common sense using a lot of random numbers), and the step by step probabilistic search based on the minimization of Karl Pearson's χ^2-indicator from statistical inference. A numerical example is presented, solved by using the statistical computer package MINITAB 11 on WINDOWS, for the heuristic simulation, and the simbolic computing package MATHEMATICA, version 2.1.

References

[1] Karmarkar, N., A new polynomial time algorithm for linear programming, *Combinatorica*, Vol.4, 1984, pp. 373-395.

[2] Guiasu, S., Using probabilistic measures of uncertainty for solving linear programs, *Proceedings of the Tenth International Conference* IPMU 2004, Editrice Universita La Sapienza, Roma, Vol.2, 2004, pp.1211-1216.

[3] Pearson, K., On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, *Philosophical Magazine* (5th series), Vol.50, 1900, pp.157-175.

[4] Winston, W.L., *Operations Research. Applications and Algorithms*, (3rd edition), Duxbury Press, Belmont, California, 1994.

[5] Wolfram, S., *Mathematica. A System for Doing Mathematics by Computer*, (2nd edition), Addison-Wesley Publishing Co., Redwood City, California, 1991.

[6] Ryan, B.F., Joiner, B.L., and Ryan Jr., T.A., *Minitab. Handbook.* (2nd edition), Duxbury Press, Boston, 1985.