Re: Gaussian elimination with guaranteed positive coefficients

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2007−01/msg00216.html

- *From*: "Marshall" <magreenblatt@xxxxxxxxx>
- *Date*: 21 Jan 2007 14:26:40 −0800

Hi Helmut,

Thank you for your reply. It seems that linear optimization is exactly
what I need. More specifically, the "push and pull" and/or "standard
simplex" approaches as described below seem to be a perfect match.

http://home.ubalt.edu/ntsbarsh/Business−stat/opre/partVIII.htm

Regards,
Marshall

On Jan 21, 9:36 am, Helmut Jarausch <jarau...@xxxxxxxxx> wrote:

> I don't thing Gaussian elimination is the right tool for your problem.
> It looks like an optimization problem (or a set of inequalities) where
> your positivity requirements are just further constraints.
> If your functions W(a), etc, are linear, this is a classical linear
> optimization problem, otherwise it's a nonlinear optimization problem.
> Have a look at http://plato.la.asu.edu/guide.html
>
> Marshall wrote:
>
>> Hi All,
>>
>> I'm experimenting with gaussian elimination as a method for solving
>> percentage−based linear systems. In other words, all of the resulting
>> coefficients must be positive values and the sum of all coefficients
>> must be 1.
>>
>> Is anyone familiar with a gaussian elimination algorithm variant (or a
>> completely different computational approach to the problem) that will

always search for a positive coefficient solution?

My concern with the generic triangular echelon/back−substitution implementation is that the algorithm will (I believe) complete on the first match, which might contain negative coefficients, even if an all−positive−coefficient solution exists.

I can, of course, handle the "add up to one" portion by using an equation like "a + b + c = 1".

Eventually, I'll be using numerical iteration in combination with the linear system solver to locate valid coefficient sets. As a trivial example, let's say that we want to light our office with three different light bulbs, and have only one light bulb on at a time. Each light bulb (a, b and c) has a different wattage and lifespan. Say further that we want our average power consumption per day to be between 6 and 7 watts, and we want all of our bulbs to burn out between 200 and 250 days. The application would allow us to specify each light bulb's attributes, plug in the desired ranges, and then solve for the length of time as a percentage of the whole day that each light bulb should be left on. Therefore the equations to solve would be:

6 < W(a) * a + W(b) * b + W(c) * c < 7
200 < L(a) * a + L(b) * b + L(c) * c < 250
a + b + c = 1

Where W(a) and L(a) are respectively the wattage and lifespan of light bulb a, and so on. Any set of positive coefficient values for a, b and c would represent a valid solution to the problem. The application would therefore iterate over the complete range for W and L to create an map of all value combinations that generate valid coefficients.

If there were more than just two range requirements (say we also want our lightbulbs to average a certain number of lumens) then we would calculate the solution map for each combination of two ranges separately and then analyze the resulting maps for overlap. If there were less than two range requirements we would choose arbitrary in−range values for one of the light bulbs to reduce the number of coefficients and then solve for the other light bulb coefficients.

> Does this sound feasible? Can anyone think of a better way to tackle
> this type of problem? Does anyone know of an algorithm for solving
> systems of linear equations that meets these requirements?−−

Helmut Jarausch

Lehrstuhl fuer Numerische Mathematik
RWTH − Aachen University
D 52056 Aachen, Germany

.