Sam Burer <sburer@gmail.com>

Problem about quadprogbb - a bit urgent

9 messages

C.Zor@surrey.ac.uk <C.Zor@surrey.ac.uk> **Tue, Nov 8, 2011 at 1:19 PM**
To: samuel-burer@uiowa.edu

Dear Sam Burer,

I have started using your nonconvex quadratic programming global optimizer function, namely, quadprogbb in my research on the optimization the Error Correcting Output Coding matrix for data classification. However I came across a problem and I will really appreciate it if you can help me as I really am short of time.

I have attached some data for the toy problem I want to consult you about.

The problem is that, in my constraints matrix A, I apparently have a dependent row. When I try to use quadprog, without removing this row, it gives a warning, but comes up with very good solutions when tried with random 10-15 initial point. However, quadprogbb, crashes, by saying, "102-th LP upper bound cannot be obtained: either unbounded or infeasible!"

When I remove the dependent row, quadprogbb works, and even with a time restriction, is better than quadprog (if I also remove this row for quadprog as well). However, these two results are unfortunately not better than quadprog's initial answer, when the row was not removed! At this point, first of all I would like to understand why quadprog is giving so well results when a dependent row is not removed than removed; and also that why quadprogbb is crashing when I dont remove the line.. I would expect it go give me better results than quadprog..

I would like to add that these happen when i give a time restriction like 6 minutes, etc.

The function calls Im using are:

```
opsiyon=struct('max_time',time);
[x, fValCheck, ExitFlag, Output, Lambda]=quadprog(F,c, A2,b2, Aeq, beq, x_L, x_U,x_0);
[x, fValCheck, time, status]=quadprogbb(F, c, A2, b2, Aeq, beq, x_L, x_U,opsiyon);
```

vs

```
opsiyon=struct('max_time',time);
[x, fValCheck, ExitFlag, Output, Lambda]=quadprog(F,c, A2(1:end,:) ,b2(1:end,:) , Aeq, beq, x_L,
x_U,x_0);
[x, fValCheck, time, status]=quadprogbb(F, c, A2(1:end,:) , b2(1:end,:) , Aeq, beq, x_L,
x_U,opsiyon);
```

You can find all the variables stored in mat files attached to the email.

I will appreciate your help a lot really,
Best Wishes,
Cemre

cemre.zip
5K

Sam Burer <samuel-burer@uiowa.edu> **Tue, Nov 8, 2011 at 1:51 PM**
To: C.Zor@surrey.ac.uk

Dear Cemre, thank you for your email. I will try to help as soon as I can.

I don't see the difference between your two calls because I believe A2(1:end,:) is the same as A2 and b2(1:end) is the same as b2. Can you please clarify?

Sam

[Quoted text hidden]

C.Zor@surrey.ac.uk <C.Zor@surrey.ac.uk> **Tue, Nov 8, 2011 at 6:05 PM**
To: samuel-burer@uiowa.edu

Dear Sam,

So sorry, it is (1:end-1,:) instead of (1:end,:) in both cases, meaning I removed the dependent row. And this changes the accuracy a lot in quadprog, whereas it causes a problem in quadprogbb unless I remove it.

Looking forward to hearing from you
Best Wishes

From: Sam Burer [samuel-burer@uiowa.edu]
Sent: 08 November 2011 19:51
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Dear Cemre, thank you for your email. I will try to help as soon as I can.

I don't see the difference between your two calls because I believe A2(1:end,:) is the same as A2 and b2(1:end) is the same as b2. Can you please clarify?

Sam

[Quoted text hidden]

Sam Burer <samuel-burer@uiowa.edu> **Wed, Nov 9, 2011 at 11:45 AM**
To: C.Zor@surrey.ac.uk

Hi again, Cemre,

I noticed that A2 and b2 only have one row, and both are 0! So even (1:end-1) does not really make sense to me. Can you please double check your model carefully? Eventually, QuadProgBB will check some of these things, but we don't have all those features yet.

Thanks!
Sam

[Quoted text hidden]

C.Zor@surrey.ac.uk <C.Zor@surrey.ac.uk> **Wed, Nov 9, 2011 at 11:50 AM**
To: samuel-burer@uiowa.edu

Hello,

Sorry that the more urgent this got, the more mistaken emails Im sent you. It is Aeq and beq, having the last row dependent to the rest, not A2 and b2. So (1:end-1) applies to them.. I am so sorry, really, to give misinformation many times, but just that I'm so short of time, apparently I typed an email full of typos. So sorry, and thanks again and again.

I think now I can look forward to hearing from you :)

From: Sam Burer [samuel-burer@uiowa.edu]
Sent: 09 November 2011 17:45
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Hi again, Cemre,

I noticed that A2 and b2 only have one row, and both are 0! So even (1:end-1) does not really make sense to me. Can you please double check your model carefully? Eventually, QuadProgBB will check some of these things, but we don't have all those features yet.

Thanks!
Sam

On Tue, Nov 8, 2011 at 6:05 PM, <C.Zor@surrey.ac.uk<mailto:C.Zor@surrey.ac.uk>> wrote:
Dear Sam,

So sorry, it is (1:end-1,:) instead of (1:end,:) in both cases, meaning I removed the dependent row. And this changes the accuracy a lot in quadprog, whereas it causes a problem in quadprogbb unless I remove it.

Looking forward to hearing from you
Best Wishes

From: Sam Burer [samuel-burer@uiowa.edu<mailto:samuel-burer@uiowa.edu>]
Sent: 08 November 2011 19:51
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Dear Cemre, thank you for your email. I will try to help as soon as I can.

I don't see the difference between your two calls because I believe A2(1:end,:) is the same as A2 and b2(1:end) is the same as b2. Can you please clarify?

Sam

[Quoted text hidden]

Sam Burer <samuel-burer@uiowa.edu> **Wed, Nov 9, 2011 at 2:37 PM**
To: C.Zor@surrey.ac.uk

Hi Cemre,

I have been able to get QuadProgBB to work with the commands at the end of this file. Theoretically, QuadProgBB needs Aeq to have full row rank, which means it cannot have dependent rows. This (along with the zero A2 and b2) was causing the problem. We will try to eventually add detection of this to QuadProgBB, but in the meantime, can you manually make sure your Aeq is full row rank? This can be done without loss of generality.

The problem itself is taking sometime to solve. You will probably experience the same thing. The best solution found so far has value -2.63714667e+03, but the global gap is 2.674%. I believe this should compare well with quadprog's answer.

Thanks!
Sam

```
load A2
load b2
load Aeq
load beq
load c
load F
load x_0
load x_L
load x_U
opsiyon=struct('max_time',3600000);
opsiyon.use_single_processor = 1;
[x,fValCheck,ExitFlag,Output,Lambda] = quadprog  (F,c,[],[],Aeq(1:end-1,:),beq(1:end-1),x_L,x_U,x_0);
[x,fValCheck,time,status]           = quadprogbb(F,c,[],[],Aeq(1:end-1,:),beq(1:end-1),x_L,x_U, opsiyon);
```

On Wed, Nov 9, 2011 at 11:50 AM, <C.Zor@surrey.ac.uk> wrote:

> Hello,
>
> Sorry that the more urgent this got, the more mistaken emails Im sent you. It is Aeq and beq, having the last row dependent to the rest, not A2 and b2. So (1:end-1) applies to them.. I am so sorry, really, to give misinformation many times, but just that I'm so short of time, apparently I typed an email full of typos. So sorry, and thanks again and again.

> I think now I can look forward to hearing from you :)

C.Zor@surrey.ac.uk <C.Zor@surrey.ac.uk> **Wed, Nov 9, 2011 at 3:31 PM**
To: samuel-burer@uiowa.edu

Dear Sam,

Firstly, thanks a lot for the answer. Surely, I can do the dependancy check. However, let me ask you at this point if you have any idea why "quadprog" works much better without the row removed than it being removed? It shouldnt actualy change anything, but it does. I tried this with many initial points.

Secondly, I'm sure quadprogbb will converge to the best result of quadprog in the end, but it took a lot of time in my case. Runs of 10 like minutes etc, didnt give me reliable results. I need to run the algorithm on much bigger matrices, and thats why I will need to put a limit on the timing. And as long as I have an estimate of the minimum, better than quadprog, I want to use quadprogbb. However I think with time restrictions, quadprogbb is not reaching as good results as quadprog with random initial points? May I ask how long it took you to get -2.63714667e+03 for this problem, more or less?

I do appreciate your help
Best Wishes

From: Sam Burer [samuel-burer@uiowa.edu]
Sent: 09 November 2011 20:37
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Hi Cemre,

I have been able to get QuadProgBB to work with the commands at the end of this file. Theoretically, QuadProgBB needs Aeq to have full row rank, which means it cannot have dependent rows. This (along with the zero A2 and b2) was causing the problem. We will try to eventually add detection of this to QuadProgBB, but in the meantime, can you manually make sure your Aeq is full row rank? This can be done without loss of generality.

The problem itself is taking sometime to solve. You will probably experience the same thing. The best solution found so far has value -2.63714667e+03, but the global gap is 2.674%. I believe this should compare well with quadprog's answer.

Thanks!
Sam

```
load A2
load b2
load Aeq
load beq
load c
load F
load x_0
load x_L
load x_U
opsiyon=struct('max_time',3600000);
opsiyon.use_single_processor = 1;
[x,fValCheck,ExitFlag,Output,Lambda] = quadprog  (F,c,[],[],Aeq(1:end-1,:),beq(1:end-
1),x_L,x_U,x_0);
[x,fValCheck,time,status]           = quadprogbb(F,c,[],[],Aeq(1:end-1,:),beq(1:end-1),x_L,x_U,
opsiyon);
```

[Quoted text hidden]

Sam Burer <samuel-burer@uiowa.edu> **Wed, Nov 9, 2011 at 5:31 PM**
To: C.Zor@surrey.ac.uk

Hi Cemre,

I'm really sorry, but I don't know why quadprog would be giving this behavior. Since it is an internal function of Matlab, I just can't be sure what's going on inside it.

You may very well be right that quadprogbb will take longer to get the same value as quadprog, but I would like to stress that the purpose of quadprogbb is to find a global solution. If you're happy with a non-global (but good) solution, then quadprog is quite reasonable. To actually find and prove the global solution typically takes much, much longer. (This is very common in all types of optimization problems.)

It took me about one hour to find -2.63714667e+03. After 2.5 hours, I found -2.66925620e+03, which is provably within 1.365% of optimality.

If you'd like to get an idea of how our method behaves against other global methods, please see the first reference in the QuadProgBB help file (which you can find with a Google search).

Sam

[Quoted text hidden]

C.Zor@surrey.ac.uk <C.Zor@surrey.ac.uk> **Thu, Nov 10, 2011 at 2:50 AM**
To: samuel-burer@uiowa.edu

Thanks a lot for your time and consideration, I appreciated that a lot.
Best Wishes and Regards
Cemre

From: Sam Burer [samuel-burer@uiowa.edu]
Sent: 09 November 2011 23:31
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Hi Cemre,

I'm really sorry, but I don't know why quadprog would be giving this behavior. Since it is an internal function of Matlab, I just can't be sure what's going on inside it.

You may very well be right that quadprogbb will take longer to get the same value as quadprog, but I would like to stress that the purpose of quadprogbb is to find a global solution. If you're happy with a non-global (but good) solution, then quadprog is quite reasonable. To actually find and prove the global solution typically takes much, much longer. (This is very common in all types of optimization problems.)

It took me about one hour to find -2.63714667e+03. After 2.5 hours, I found -2.66925620e+03, which is provably within 1.365% of optimality.

If you'd like to get an idea of how our method behaves against other global methods, please see the first reference in the QuadProgBB help file (which you can find with a Google search).

Sam

On Wed, Nov 9, 2011 at 3:31 PM, <C.Zor@surrey.ac.uk<mailto:C.Zor@surrey.ac.uk>> wrote:
Dear Sam,

Firstly, thanks a lot for the answer. Surely, I can do the dependancy check. However, let me ask you at this point if you have any idea why "quadprog" works much better without the row removed than it being removed? It shouldnt actualy change anything, but it does. I tried this with many initial points.

Secondly, I'm sure quadprogbb will converge to the best result of quadprog in the end, but it took a lot of time in my case. Runs of 10 like minutes etc, didnt give me reliable results. I need to run the algorithm on much bigger matrices, and thats why I will need to put a limit on the timing. And as long as I have an estimate of the minimum, better than quadprog, I want to use quadprogbb. However I think with time restrictions, quadprogbb is not reaching as good results as quadprog with random initial points? May I ask how long it took you to get -2.63714667e+03 for this problem, more or less?

I do appreciate your help
Best Wishes

From: Sam Burer [samuel-burer@uiowa.edu<mailto:samuel-burer@uiowa.edu>]
Sent: 09 November 2011 20:37
To: Zor C Ms (PG/R - Electronic Eng)
Subject: Re: Problem about quadprogbb - a bit urgent

Hi Cemre,

I have been able to get QuadProgBB to work with the commands at the end of this file. Theoretically, QuadProgBB needs Aeq to have full row rank, which means it cannot have dependent rows. This (along with the zero A2 and b2) was causing the problem. We will try to eventually add detection of this to QuadProgBB, but in the meantime, can you manually make sure your Aeq is full row rank? This can be done without loss of generality.

The problem itself is taking sometime to solve. You will probably experience the same thing. The best solution found so far has value -2.63714667e+03, but the global gap is 2.674%. I believe this should compare well with quadprog's answer.

Thanks!
Sam

```
load A2
load b2
load Aeq
load beq
load c
load F
load x_0
load x_L
load x_U
opsiyon=struct('max_time',3600000);
opsiyon.use_single_processor = 1;
[x,fValCheck,ExitFlag,Output,Lambda] = quadprog  (F,c,[],[],Aeq(1:end-1,:),beq(1:end-
1),x_L,x_U,x_0);
[x,fValCheck,time,status]            = quadprogbb(F,c,[],[],Aeq(1:end-1,:),beq(1:end-1),x_L,x_U,
opsiyon);
```

[Quoted text hidden]