Constraint Qualification Examples

The constraint qualification is important in optimization problems.

LICQ: The binding constraints are linearly independent at the solution

This is assumed in almost all optimization solvers.

Economists ignore CQ. Most have not heard of it!!!

Others assume there is no problem

A Brief Detour: This result guarantees the existence of multipliers that lead to a saddle point. It is often the case that the multipliers have important information about the sensitivity of the optimum to changes in the value of b. For the multipliers to have this information, some variant on a condition called the **constraint qualification (CQ)** must hold. We have never encountered in our own work an optimization problem in which the multipliers were not unique and informative. However, such examples can be constructed if you are careful and have a perverse turn of mind. Having the requisite perverse turn of mind, we give an example in §5.9.d.

Simple example

Maximize utility subject to two constraints - money and time

utility function:	x y
money price of x:	10
money price of y:	1
money budget:	21
time price of x:	1
time price of y:	8
time budget:	10

Since theorems are in terms of minimization, the objective is the negative of the utility function.

In[16]:=
```
obj = - x y;
c1 = 10 x + y - 21;
c2 = x + 8 y - 10;
```

Form Lagrangian

In[19]:=
```
lag = obj + λ1 c1 + λ2 c2
```

Out[19]= $-x\,y+(-21+10\,x+y)\,\lambda1+(-10+x+8\,y)\,\lambda2$

In[20]:=
```
foc1 = D[lag, x]
```

Out[20]= $-y+10\,\lambda1+\lambda2$

In[21]:=
```
foc2 = D[lag, y]
```

Out[21]= $-x+\lambda1+8\,\lambda2$

The solution for the primal variables is x=2 and y=1.

We substitute this into first-order conditions to get equations for the dual variables (the shadow prices)

In[22]:=
```
{foc1, foc2} /. x → 2 /. y → 1
```

Out[22]= $\{-1 + 10\,\lambda 1 + \lambda 2,\ -2 + \lambda 1 + 8\,\lambda 2\}$

In[23]:=
```
Solve[% == 0, {λ1, λ2}][[1]]
```

Out[23]= $\left\{\lambda 1 \to \frac{6}{79},\ \lambda 2 \to \frac{19}{79}\right\}$

We have unique multipliers, both positive, as predicted by KKT

Another “simple” example (or is it “perverse”)

Same problem as above but add constraint that x + y cannot exceed 3. This reflects some other constraint, such as the ability to carry it away from the store.

In[25]:=
```
c3 = x + y - 3;
```

Form Lagrangian

In[29]:=
```
lag = obj + λ1 c1 + λ2 c2 + λ3 c3;
```

Compute first-order conditions

In[27]:=
```
foc1 = D[lag, x]
```

Out[27]= $-y + 10\,\lambda1 + \lambda2 + \lambda3$

In[28]:=
```
foc2 = D[lag, y]
```

Out[28]= $-x + \lambda1 + 8\,\lambda2 + \lambda3$

Note that solution is still x=2 and y=1. So, substitute them into foc1 and foc2 to get equations involving only the shadow prices

In[31]:=
```
focs = {foc1, foc2} /. x → 1 /. y → 1;
focs // TableForm
```

Out[32]//TableForm=

$-1 + 10\,\lambda 1 + \lambda 2 + \lambda 3$

$-1 + \lambda 1 + 8\,\lambda 2 + \lambda 3$

Solve for shadow prices:

In[33]:=
```
Solve[focs == 0, {λ1, λ2, λ3}][[1]]
```

Solve::svars : Equations may not give solutions for all "solve" variables. ≫

Out[33]= $\left\{\lambda 2 \to \frac{9\,\lambda 1}{7},\ \lambda 3 \to 1 - \frac{79\,\lambda 1}{7}\right\}$

The multipliers are not unique!!

Of course they are not unique!! You have three unknown dual variables but only two equations linear in the dual variables!!