Ch 5. Trust Region Algorithms (non-linear)

The fundemental problem with Newton's method for unconstrained minimization of a function is it can fail to converge to a local minimum (due to non-positive definite matrices), especially if x^k isn't close to x^*.

A challenge in implementing Newton's method is globalisation / global convergence. This is modifying the algorithm so it always converges to a minimum point. This challenge is usually addressed by adding a line search to move $x^{k+1} = x^k + \alpha d^k$ instead of just $x^{k+1} = x^k + d^k$. This makes the method more robust, and will make Newton's method converge to a local minimum when conditions are imposed on the line search eg. Wolfe, armijo etc.

Trust region methods are the alternative approach to globalizing Newton's method. Here, we construct an approximation of f near a point x^k using the start of the taylor series:

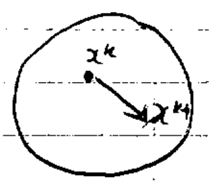
$$m_k(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k)$$

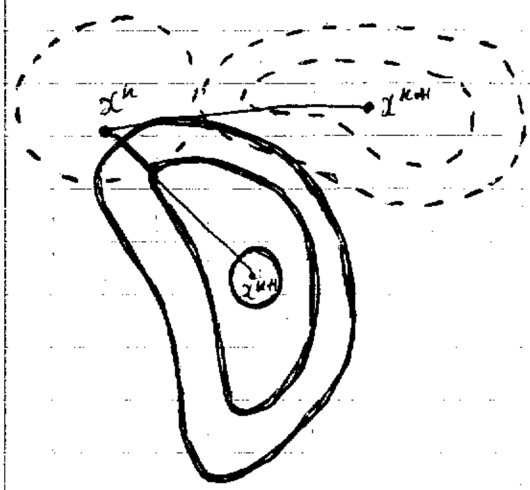
This approximation is only valid near x^k so we set a bound Δ^k on how large we let $x - x^k$ get, and then minimize the set:

$$\text{eg.} \quad \left.\begin{aligned} &\min_x \; m_x(x) \\ &\text{such that } \|x - x^k\| \leq \Delta^k \end{aligned}\right\} \text{ trust region subproblem.}$$

Note: if Δ^k large enough, then the solution to the minimization problem is the same point we get by performing a step of Newton's method without line search.

Once we've solved the subproblem, we move to the point, adjust Δ^k and repeat. With an appropriate strategy for adjusting tolerance Δ^k, we can construct an algorithm with the same global convergence properties as Newton's method with line search (eg. can converge from any x^0).





= region of trust
= Quadratic approximation of $f(x)$, m_k
= contours of $f(x)$
= result via line search
= direction of trust method
= approximate improvement of x^k

Note, we can rewrite:

$$m_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T G^k d$$

$$\min m_k(d) : \quad \|d\| \leq \Delta^k.$$

Therefore,
$d^* = d^k$ minimizes the ball of radius Δ^k.

If $G^k = \nabla^2 f(x^k)$ is the Hessian at x^k, the method is the trust region Newton method
If it is an approximation to the Hessian, it's called the trust region Quasi Newton method.

Notice $m_k(d)$ is a quadratic function and so is its constraint, thus:

$$\|d\| \leq \Delta^k$$

$$d^T d \leq (\Delta^k)^2$$

$$\Rightarrow \sqrt{d^T d} \leq \Delta^k$$

When G^k is positive definite and $\|G^{(k)-1} \nabla f\| \leq \Delta^k$, then the minimizer of m_k is $d^k = \frac{-\nabla f}{G^k}$. This is called d^k full step, and takes us to the boundary of the trust region. Note, d^k is bounded.

If G^k isn't positive definite, the solution for d^k isn't so trivial, but we then only need an approximate solution.

Trust Region Algorithm

Rewrite the problem with a general matrix B^k, which can be a hessian or an approximation from the Quasi-Newton method, or a finite-difference approximation.

$$m_k(d) = f(x^k) + \nabla f(x^k)^T d + d^T B^k d$$

$$\min m_k(d) \quad \text{where } \|d\| \le \Delta^k.$$

The trust region method can be motivated by noting that the quadratic $m_k(x)$ is a useful model of $f(x)$ only near x^k.

When the Hessian matrix is indefinite, the quadratic function $m_k(d)$ is unbounded below. Thus, it's a poor model of f when d is large. ← why?
Thus, it's reasonable to select the step by solving the subproblem:

$$\min m_k(d) \;:\; \|d\| \le \Delta^k$$

The trust region parameter Δ^k is adjusted between iterations, according to an agreement between predicted and actual reduction in the function, as a ratio:

$$\rho^k = \frac{f(x^k) - f(x^{k+1})}{f(x^k) - m_k(d)} = \frac{f(x^k) - f(x^k + d)}{m_k(0) - m_k(d)} \quad \begin{array}{l} \rightarrow \text{actual reduction} \\ \rightarrow \text{predicted reduction} \end{array}$$

- If $\rho^k \approx 1$, good agreement $\Rightarrow$ increase Δ^k
- If ρ^k small/negative, poor agreement. This happens via
  1. $f(x^{k+1}) > f(x^k)$
  2. $m_k(d) > m_k(0)$
  3. Predicted > actual

Thus, Δ^k decreased.

Usually, $x^{k+1} = x^k + d^k$ if ρ_k is a good choice
Else, we let $x^{k+1} = x^k$ and choose a new direction

ALGORITHM

1. Choose $\bar{\Delta} > 0$
   $\Delta_0 \in (0, \bar{\Delta})$
   $\eta \in [0, \frac{1}{4})$

2. Obtain d_k by solving $\min m_k(d)$ where $\|d\| \leq \Delta^k$
   Evaluate ρ^k
   If $f(x^k) = m_k(d^k)$, stop.

3. If $\rho^k > \frac{1}{4}$, then $\Delta_k = 0.25\|d^k\|$
   Else if $\rho^k > \frac{3}{4}$ and $\|d^k\| = \Delta_k$, then $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
   Else, $\Delta_{k+1} = \Delta_k$

4. If $\rho^k > \eta$, $x^{k+1} = x^k + d^k$ and go to step 2
   Else, $x^{k+1} = x^k$ and go to step 2.

$\bar{\Delta}$ is an overall bound on the step length. The radius is increased only if $\|d^k\|$ actually reaches the boundary of the trust region. If the step stays strictly in the region, then Δ^k isn't interfering with the algorithm, so we leave it unchanged in the next iteration.

We need to solve m_k approximately; a way to do this is via the dogleg method, to be used when B^k is positive definite. The solution then obtained is at least as much reduction in m_k as gotten via the Cauchy point.

The Cauchy Point

For global convergence, it suffices to find an approximate d^k to minimize $m_k(d)$ that lies in the trust region and gives sufficient reduction in the model. This sufficient reduction is quantified in terms of the Cauchy point, denoted d_c^k.

The Cauchy point calculation involves 2 steps:

1. Find d_s^k that solves a linear version of $m_k(d)$:

$$d_s^k = \underset{d \in \mathbb{R}^n}{\arg\min}\ f(x^k) + \nabla f(x^k)^T d \qquad \|d\| \leq \Delta^k$$

2. Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau_k d_s^k)$ subject to satisfying the trust region bound:

$$\tau_k = \arg\min_{\tau > 0} m_k(\tau d_s^k) \qquad \|\tau d_s^k\| \leq \Delta^k$$

The Cauchy point is given by: $d_c^k = \tau_k d_s^k$
The closed form is:

$$d_s^k = \frac{-\Delta^k}{\|\nabla f(x^k)\|} \nabla f(x^k)$$

If $\nabla f(x^k)^T B^k \nabla f(x^k) \leq 0$:
$m_k(\tau d_s^k)$ decreases monotonically with τ whenever $\nabla f(x^k) \neq 0$
$\Rightarrow \tau_k$ largest value that satisfies the trust region bound $\tau_k = 1$

If $\nabla f(x^k)^T B^k \nabla f(x^k) > 0$:
$m_k(\tau d_s^k)$ is convex quadratic in τ
$\Rightarrow \tau_k$ is either the unconstrained minimizer $\frac{\|\nabla f(x^k)\|^3}{\Delta^k \nabla f(x^k)^T B_k \nabla f(x^k)}$
or boundary value 1 (whichever comes first).

- Advantage: → d_c^k is inexpensive to calculate (due to closed form).
  → Cauchy point is NB in deciding if an approximate solution of the subproblem is acceptable
  if $(m_k(0) - m_k(d^k)) \geq c(m_k(0) - m_k(d_c^k))$ $c \in (0,1)$, accepted.
  → Trust region methods are globally convergent if d^k attain sufficient decrease in m_k eg. they give a reduction in m_k that's some fixed multiple of the decrease attained by the Cauchy point.

- Disadvantage: → If we implement the Cauchy point as our step always, this is simply steepest descent, which performs poorly in optimizing.

More on Trust Region @ back

Dogleg Method

When B^k is positive definite and d^k remains in the bounds of the subproblem the step taken is called a full step, denoted:

$$d_b^k = -(B^k)^{-1} \nabla f(x^k) \qquad \| d_b^k \| \leq \Delta^k$$

When Δ^k is small, then $\|d^k\| \leq \Delta^k$ ensures the quadratic term in m_k has little effect in the solution of the subproblem. So, it makes sense to minimize the linear function:

$$f(x^k) + \nabla f(x^k)^T d^k \qquad \|d^k\| \leq \Delta^k$$

$$\Rightarrow d^k = -\Delta^k \frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$$

For intermediate values of Δ^k, the dogleg method finds an approximate solution d_d^k using 2 line segments.

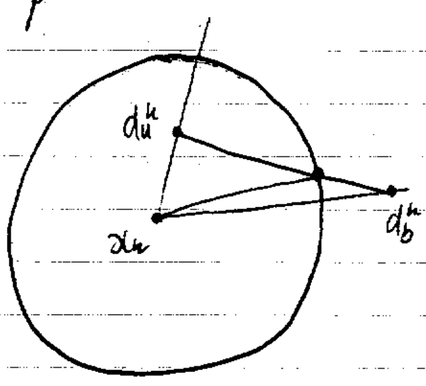
Segment 1 – from origin to the unconstrained minimizer along the steepest descent direction:

$$d_u^k = \frac{-\nabla f(x^k)^T \nabla f(x^k)}{\nabla f(x^k)^T B^k \nabla f(x^k)} \nabla f(x^k)$$

Segment 2 – from d_u^k to d_b^k:

$$d^k(\tau) = \begin{cases} \tau d_u^k & 0 \leq \tau \leq 1 \\ d_u^k + (\tau - 1)(d_b^k - d_u^k) & 1 \leq \tau \leq 2. \end{cases}$$

The dogleg method minimizes m_k along this path, subject to the trust region bound. Since m_k is a decreasing function along the path, the chosen $d_d^k = d_b^k$ when $\|d_b^k\| \leq \Delta^k$. τ by: $\|d_u^k + (\tau-1)(d_b^k - d_u^k)\|^2 = (\Delta^k)^2$. Else, the point of intersection of the dogleg + trust region boundary.



= steepest descent direction
= line segment 2
= length of vector = trust region radius

2. Theory of constrained Optimization

We consider to minimize a function subject to constraints on the variable.

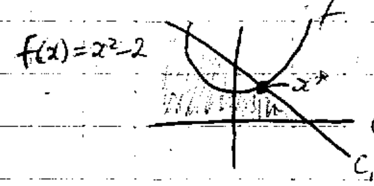
A general formulation of the problem:

$$\min_{x \in \mathbb{R}^n} f(x) \qquad (1)$$

$$\text{subject to: } \begin{cases} c_i(x) = 0 & i \in \mathcal{E} \\ c_i(x) \geq 0 & i \in \mathcal{I} \end{cases}$$

The functions f and c_i are smooth, differentiable, real-valued functions on a subset of $\mathbb{R}^n$. $\mathcal{I}$ and $\mathcal{E}$ are two sets of indeces, which are finite.

At a feasible point x, the inequality constraint $i \in \mathcal{I}$ is said to be active if $c_i(x) = 0$, and inactive if strictest inequality holds ie. $c_i(x) > 0$ is satisfied.



$c_2(x)$ has no effect on the problem $\therefore$ inactive $\therefore \lambda = 0$
$c_1(x)$ = active, solution lies on it.

<u>Example</u>

A single equality constraint problem:

$$\min \; x_1 + x_2$$

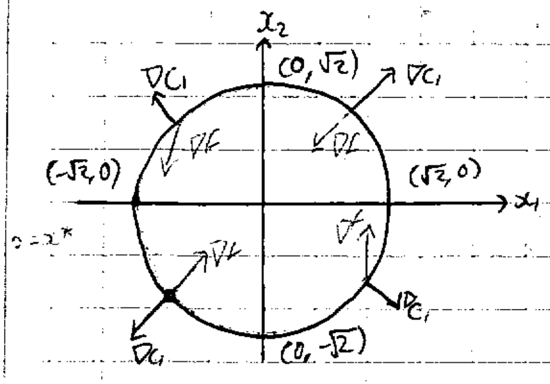
$$\text{ST: } x_1^2 + x_2^2 - 2 = 0$$

In the language of (1): $f(x) = x_1 + x_2$

$\mathcal{I} = \emptyset$

$\mathcal{E} = \{1\}$

$c_1(x) = x_1^2 + x_2^2 - 2$



eg. just the boundary

At points of extrema, ∇c_i, ∇f are parallel. We find x^* where f, c_i intersect eg. have same slope $\Rightarrow$ parallel.

The minimum of $f(x)$ on the circle is achieved at: $(-1,-1) = x^*$

Now, $\nabla f(x) = (1,1)$
$\nabla c_1(x) = (2x_1, 2x_2)$

At the solution x^*: $\nabla f(x^*) = (1,1)$
$\nabla c_1(x) = (-2,-2)$

Thus, $\nabla f(x^*), \nabla c_1(x^*)$ are parallel.

Therefore, there exists a scalar λ_1^* such that:

$$\nabla f(x^*) = \lambda_1^* \nabla c_1(x^*) \qquad (3)$$

In particular, $\lambda_1^* = \frac{-1}{2}$:

$$\nabla f(x^*) = (1,1) = -\tfrac{1}{2}(-2,-2) = \lambda_1^* \nabla c_1(x^*)$$

Condition (3) can be determined by examining the first order Taylor expansion of f and c_1.

For c_1, we need to retain:

$c_1(x) = 0$
$c_1(x+d) = 0$ (eg. active no matter what direction we move in).

$$\Rightarrow c_1(x+d) = 0 \approx \underbrace{c_1(x)}_{=0} + \nabla c_1(x)^T d$$

$$\Leftrightarrow \nabla c_1(x)^T d = 0 \qquad (5)$$

Also, for f:

$$f(x+d) \approx f(x) + \nabla f(x)^T d$$

$$\Leftrightarrow f(x+d) - f(x) \approx \nabla f(x)^T d \qquad \text{(function decrease is wanted).}$$

Since we want to decrease f,

$$0 > f(x+d) - f(x) \approx \nabla f(x)^T d$$

yields,

$$\nabla f(x)^T d < 0 \qquad (6)$$

If there exists a direction d that satisfies both ⑤ and ⑥, we conclude that improvement on our current choice of x is possible.

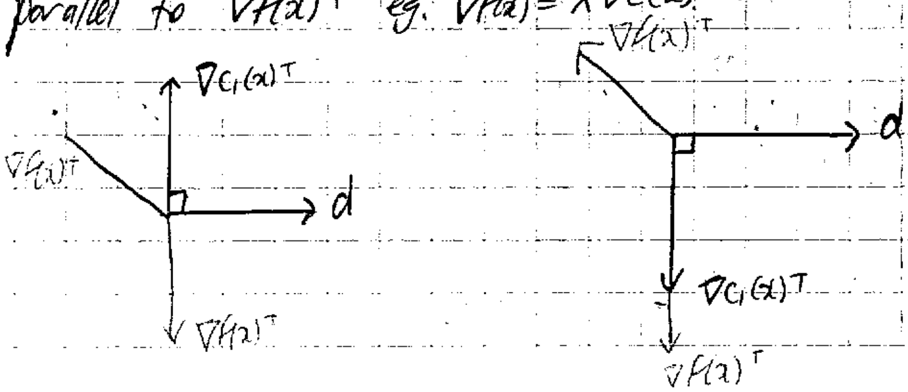
It follows that a necessary condition of optimality is if there's no direction d that satisfied both ⑤ and ⑥.

We want a d that doesn't satisfy

- $\nabla c_1(x)^T d = 0$
- $\nabla f(x)^T d < 0$

} If both satisfied ⇒ there's still room for decrease. We want ⑤ to hold, ⑥ not to hold ⇒ minimizer, must find such a d

It turns out that such a direction d is possible when $\nabla c_i(x)^T$ is parallel to $\nabla f(x)^T$ eg. $\nabla f(x) = \lambda \nabla c(x)$



If this "parallel" condition doesn't hold,

$$d = -\left(I - \frac{\nabla c_1(x)\nabla c_1(x)^T}{\|\nabla c_1(x)\|^2}\nabla f(x)\right) \qquad I = \text{identity matrix}$$

By introducing the Lagrangean $f(x)$,

$$\mathcal{L}(x, \lambda_1) = f(x) - \lambda_1 c_1(x) \qquad (8)$$

Note that: $\nabla_x L(x, \lambda_1) = \nabla f(x) - \lambda_1 \nabla c_1(x)$

The condition (3):

$$\nabla f(x^*) = \lambda_1 \nabla c_1(x^*)$$

is equivalent to

$$\nabla_x L(x^*, \lambda_1) = 0$$

This suggests that solutions for equality constrained problems is at the stationary points of L.

The scalar λ_1 is called the Lagrange Multiplier for $c_1(x)=0$.
Thus,

(3) / ⑨ is necessary for optimality (of the equality constraints) but not sufficient.

In example 1, at point $(1,1)$, we can see that:

$$(1,1) = \nabla f(x) = \tfrac{1}{2}(2,2) = \tfrac{1}{2}\nabla c_1(x)$$

But, $(1,1)$ isn't a minimum. In fact, it's a maximum.

Moreover, in the case of equality constrained problems, we can't turn ③ into a sufficient condition simply by "replacing" the sign of λ_1.

To see this, consider replacing the constraint with $2-x_1^2-x_2^2=0$

eg. $x^*=(-1,-1)$

But

$\lambda_1^* = \frac{1}{2}$

eg. $\nabla f(x) = (1,1)$ $\quad \nabla c_1(x) = (-2x_1, -2x_2)$

So, at x^*:

$\nabla c_1(x^*) = (2,2)$

$$\therefore \nabla f(x^*) = (1,1) = \tfrac{1}{2}(2,2) = \lambda \nabla c_1(x^*)$$

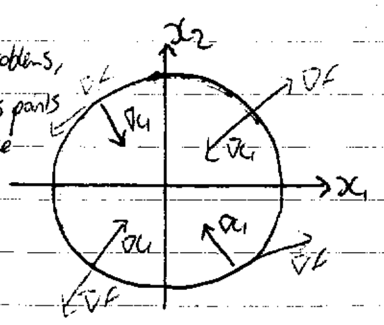
Example

A single inequality constraint problem:

NB: Turn constraints so $c \geq 0$, never $c \leq 0$

$\min \; x_1 + x_2$
$\text{st:} \; 2 - x_1^2 - x_2^2 \geq 0$

Inequality problems, ∇c always points into feasible region.



eg. feasible region = disk of radius $\sqrt{2}$, centred at 0

$$x^* = (-1,-1)$$

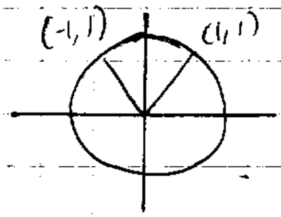
$\nabla f(x) = \nabla f(x^*) = (1,1)$ ∴ $\nabla c_1(x) = (-2x_1, -2x_2)$ ∴ $\nabla c_1(x^*) = (2,2)$

$\Rightarrow \lambda_1^* = \frac{1}{2}$

Remarks

1. ∇c_1 points towards the interior of the feasible region at the boundary of the circle at $(1,1)$

$$\nabla c_1(1,1) = (-2,-2)$$



$$\nabla c_1(1,1) = (-2,-2)$$

$$\nabla c_1(-1,1) = (2,-2).$$

2. This inequality constrained problem differs with the equality constrained problem (although the solution is the same in this case).
In particular, the Lagrange multiplier will play a significant role in this problem.

Back to our first order Taylor expansion of f and c_1, we conjecture that a point x isn't optimal if we can find a direction d that both retains feasibility and decreases f.

We still decrease f: $\nabla f(x)^T d < 0$ (eg. $f(x+d) - f(x) \approx \nabla f(x)^T d$)

Now,

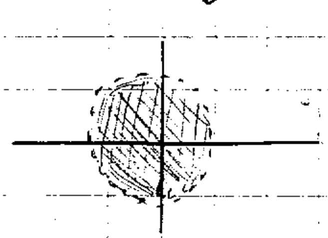
$$c_1(x+d) \geq 0 \quad \text{and} \quad c_1(x) \geq 0$$

$$\Rightarrow 0 \leq c_1(x+d) \approx c_1(x) + \nabla c_1(x)^T d$$

$$\Rightarrow c_1(x) + \nabla c_1(x)^T d \geq 0$$

- Case 1

x lies strictly inside the circle



We have $c_1(x) > 0$

Any vector d will satisfy ⑪, provided the length of d must be sufficiently small.

When $\nabla f(x^*) \neq 0$, we can obtain d that satisfies ⑥ and ⑪,

$$d = -c_1(x)\frac{\nabla f(x)}{\|\nabla f(x)\|}$$ Exercise: show d satisfies ⑥ and ⑪.

Then, the only situation in which a direction fails to exist:

$$\nabla f(x) = 0$$

$\Rightarrow x = x^*$ is optimal.

- Case 2

x lies on the boundary of the circle $c_1(x) = 0$

eg. ⑥ $\nabla f(x) < 0$
⑪ $\nabla c_1(x)^T d \geq 0$ $\quad (c_1(x) + \nabla c_1(x)^T d \geq 0) \Rightarrow \nabla c_1(x)^T d \geq 0)$

These conditions fail to hold when ∇f, ∇c_1 point in the same direction:

$$\nabla f(x) = \lambda_1 \nabla c_1(x). \quad \lambda_1 \geq 0$$

Summary for both cases

When no first order descent direction exists at some point x^*, we have:

$$\nabla_x L(x^*, \lambda_1^*) = 0 \quad \text{for some } \lambda^* \geq 0$$

Complementarity
If $c(x) = 0$, $\lambda \geq 0$
If $c(x) > 0$, $\lambda = 0$

We also require that $\lambda_1^* c_1(x) = 0$ ⑮

Condition 15 is called the complementary condition. It implies the lagrange multiplier λ_1 can be strictly positive only when c_1 is active. Conditions of this type play a central role in constrained optimization.

In case 1: $c_1(x^*) \geq 0$ so ⑮ holds when $\lambda_1^* = 0$

$$0 = \nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - \lambda_1^* \nabla c_1(x^*)$$

$\Rightarrow$ ⑭ becomes $\nabla f(x^*) = 0$

In case 2: $c_1(x)=0$

(15) allows $\lambda_i^* \geq 0$, which means (14)

$$\nabla f(x^*) - \lambda_i^* \nabla c_1(x^*) = 0$$
$$\Leftrightarrow \nabla f(x^*) = \lambda_i^* \nabla c_1(x^*) \qquad \text{for } \lambda_i^* \geq 0$$

<u>Statement of first order necessary conditions</u>

→ The previous examples suggest that a number of conditions are important to characterize the solutions of (1)

→ $\nabla_x L(x,\lambda) = 0$

→ The Lagrange multipliers $\lambda_i \geq 0$ (for all inequality constraints $c_i \geq 0$)

→ $\lambda_i c_i(x) = 0 \quad \forall c_i$

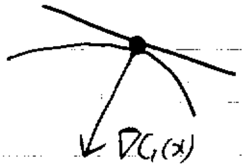
The Lagrangian function is now:

$$L(x,\lambda) = f(x) - \sum_{i \in E \cap I} \lambda_i c_i(x)$$

The active set $A(x)$

$$A(x) = E \cup \{ i \in I \mid c_i(x) = 0 \}$$

$\nabla c_i(x)$ – normal to the constraint c_i at the point x.
(usually a vector that's perpendicular to the contours of c_i



In the case of inequality constraints, it points towards the feasible side of this constraint.

It's possible however that $\nabla c_i(x)$ vanishes due to the algebraic representation of c_i. This implies that $\lambda_i c_i(x)$ vanishes $\forall \lambda_i$ and doesn't play a role in the Lagrangian gradient $\nabla_x L(x,\lambda)$.

Example

min x_1+x_2

ST: $(x_1^2+x_2^2-2)^2=0$ (*)

Now:

$$C_1(x)=(x_1^2+x_2^2-2)^2$$

$$\frac{\partial c_1}{\partial x_1} = 2\underline{(x_1^2+x_2^2-2)}2x_1 = 0$$ makes the partial derivatives $=0$

$$\frac{\partial c_1}{\partial x_2} = 2\underline{(x_1^2+x_2^2-2)}2(x_2) = 0$$ $\therefore =0\ \forall x$ feasible

$$\therefore \nabla c_1(x) = (0,0) = 0$$

$\nabla f(x) = \lambda_1 \nabla c_1(x)$ no longer holds (at $(-1,-1)$)

Assumption: a constraint qualification to ensure that such degenerate behavior doesn't occur, at all x.

Definition

Given x^* and an active set $A(x^*)$. We say that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\nabla c_i(x^*)$, $i \in A(x^*)$ is linearly independent

Note, implicitly, this implies that $\nabla c_i(x) \neq 0\ \forall i$, as a zero vector makes things linearly dependent.

Theorem (FONC under LICQ)

Suppose that x^* is a local solution of our constrained problem, and that the LICQ holds at x^*. Then, there is a Lagrange multiplier vector λ^* with components λ_i^*, $i \in E \cup I$ such that the following are satisfied at (x^*, λ^*):

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$$
$$c_i(x^*) = 0 \quad \forall i \in \mathcal{E}$$
$$c_i(x^*) \geq 0 \quad \forall i \in \mathcal{I}$$
$$\lambda_i \geq 0 \quad \forall i \in \mathcal{I}$$
$$\lambda_i c_i(x^*) = 0 \quad \forall i \in \mathcal{E} \cup \mathcal{I}$$

* Always satisfied. case 1: $c_i = 0$, case 2: $\lambda_i = 0$ } Complimentarity conditions.

These are often known as the KKT conditions.
Since the complimentarity conditions implies λ_i corresponding to inactive $c_i(x)$ are 0, we can omit terms for $i \notin A(x^*)$:

$$0 = \nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i \in A(x^*)} \lambda_i \nabla c_i(x^*)$$

Definition:
Given a local solution x^* and λ^* satisfying the KKT conditions, we say that the strict complementarity condition holds if exactly one of λ_i and $c_i(x^*) = 0$ for each $i \in \mathcal{I}$. In other words, $\lambda_i > 0 \quad \forall i \in \mathcal{I} \cap A(x^*)$.
This ensures our solution is unique.

Definition:
Given x^* and $A(x^*)$ the active set, we say the Mangasarian-Fromovitz constraint qualification (MFCQ) holds if $\exists$ a vector $w \in \mathbb{R}^n$ such that

$$\nabla c_i(x^*)^T w > 0 \quad \forall i \in A(x^*) \cap \mathcal{I}$$
$$\nabla c_i(x^*)^T w = 0 \quad \forall i \in \mathcal{E}$$

and the set of equality constraints gradients $\{\nabla c_i(x^*), i \in \mathcal{E}\}$ is linearly independent.

The MFCQ is a weaker condition that LICQ.
If LICQ satisfied, then:

$$\nabla c_i(x^*)^T w = 1 \quad \forall i \in A(x^*) \cap \mathcal{I}$$
$$\nabla c_i(x^*)^T w = 0 \quad \forall i \in \mathcal{E}$$

has a solution w, by the full rank of the active constraint gradients.

Example

Show that the feasible region defined by:

$$(x_1-1)^2+(x_2-1)^2 \le 2$$
$$(x_1-1)^2+(x_2+1)^2 \le 2$$
$$x_1 \ge 0$$

The MFCQ is satisfied at $x^*=(0,0)$ but the LICQ isn't.

C_1: $2-(x_1-1)^2-(x_2-1)^2 \ge 0$
C_2: $2-(x_1-1)^2-(x_2+1) \ge 0$
C_3: $x_1 \ge 0$

$$\nabla C = \begin{bmatrix} -2(x_1-1) & -2(x_2-1) \\ -2(x_1-1) & -2(x_2+1) \\ 1 & 0 \end{bmatrix}\Bigg|\binom{0}{0} = \begin{bmatrix} 2 & 2 \\ 2 & -2 \\ 1 & 0 \end{bmatrix}$$

Now: $\nabla C_1 w = \begin{bmatrix} 2w_1+2w_2 \\ 2w_1-2w_2 \\ w_1 \end{bmatrix} > 0 \quad \Rightarrow w_1 > w_2$
$\Rightarrow w_1 > 0$

We can thus define: $w=[w_1\ 0]$, $w_1>0$, and the MFCQ holds.

Now,

$$k_1\nabla C_1(x)+k_2\nabla C_2(x)+k_3\nabla C_3(x)=0$$

$$k_1\begin{pmatrix}2\\2\end{pmatrix}+k_2\begin{pmatrix}2\\-2\end{pmatrix}+k_3\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}$$

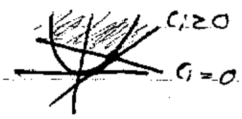
$2k_1+2k_2+k_3=0 \quad \therefore 4k_1+k_3=0 \Rightarrow k_3=-4k_1$
$2k_1-2k_2=0 \quad \therefore k_1=k_2$
$\therefore$ not independent. else $k_1=k_2=k_3=0$ which it doesn't.

eg. $k_1=1$ $k_2=1$ $k_3=-4$

Second Order Conditions

We know: $\nabla f(x)^T d = 0$
$$\left.\begin{array}{ll} c_i(x) \geq 0 & i \in \mathcal{I} \\ c_i(x) = 0 & i \in E \end{array}\right\} \nabla \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \lambda^T \nabla c(x^*) = 0$$

and
$$\lambda_i^* c_i(x^*) = 0$$



Constraints can be: Inactive $c_i > 0$ $\lambda = 0$
Active $c_i = 0$ $\lambda > 0$
Weakly active $c_i = 0$ $\lambda = 0$ (active but doesn't affect x^* selection)

Definition
Given a point x^* and the active constraint set $A(x^*)$, the set F is:

$$F_1 = \left\{ \alpha d \;\middle|\; \begin{array}{lll} \alpha > 0, & d^T \nabla c_i(x^*) = 0 & \forall i \in E \\ \alpha > 0, & d^T \nabla c_i(x^*) \geq 0 & \forall i \in A(x^*) \cap \mathcal{I} \end{array} \right\}$$

When constraint qualification is satisfied, F_1 is the tangent cone to the feasible set at x^*.

We define $F_2 \subseteq F_1$, with λ^* satisfying the KKT conditions, by:

$$F_2(\lambda^*) = \{ w \in F_1 \mid \nabla c_i(x^*)^T w = 0, \; \forall i \in A(x^*) \cap \mathcal{I}, \; \lambda_i > 0 \}$$
w - direction vector.

OR
$$w \in F_2(\lambda^*) \Rightarrow \begin{cases} \nabla c_i(x^*)^T w = 0 & \forall i \in E \\ \nabla c_i(x^*)^T w = 0 & \forall i \in A(x^*) \cap \mathcal{I}, \; \lambda_i > 0 \\ \nabla c_i(x^*)^T w \geq 0 & \forall i \in A(x^*) \cap \mathcal{I}, \; \lambda_i = 0 \end{cases}$$

It follows,
$$w \in F_2(\lambda^*) \Rightarrow \lambda_i^* \nabla c_i(x^*)^T w = 0 \quad \forall i \in E \cup \mathcal{I}$$

So,
$$\nabla c_i(x^*)^T w = 0$$
$$\Rightarrow \lambda_i \nabla c_i(x^*)^T w = 0 \qquad \lambda_i > 0$$

But
$$\lambda_i \nabla c_i(x^*) = \nabla f(x^*)$$

$\Rightarrow \nabla f(x^*)^T w = \lambda_i \nabla c_i(x^*)^T w = 0$

$\therefore$ w directions satisfy FONC.

eg.

$$w \in F_2(\lambda^*) \Rightarrow w^T \nabla f(x^*) = \sum_{i \in E \cup I} \lambda_i^* \nabla c_i(x^*)^T w = 0$$

Hence, F_2 has direction of F_1 for which it isn't clear if from the first derivatives, it'll increase or decrease.

Theorem (SONC)

Suppose x^* is a local solution of our problem and that the LICQ condition is satisfied. Let λ^* be a Lagrange multiplier vector satisfying the KKT conditions, and let F_2 be as usually defined. Then,

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w \geq 0 \qquad \forall w \in F_2(\lambda^*)$$

A strict local solution satisfies:

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w > 0 \qquad \forall w \in F_2(\lambda^*) \quad w \neq 0$$

Example

Consider the following

$$\min \ x_1 + x_2$$
$$\text{ST: } 2 - x_1^2 - x_2^2 \geq 0$$

$$\mathcal{L} = x_1 + x_2 - \lambda(2 - x_1^2 - x_2^2)$$

$$\nabla \mathcal{L} = \begin{pmatrix} 1 + 2\lambda x_1 \\ 1 + 2\lambda x_2 \\ 2 - x_1^2 - x_2^2 \end{pmatrix} = 0 \quad \Rightarrow \quad \begin{aligned} x_1 &= \tfrac{-1}{2\lambda} \\ x_2 &= \tfrac{-1}{2\lambda} \\ 2 - (\tfrac{-1}{2\lambda})^2 - (\tfrac{-1}{2\lambda})^2 &= 0 \quad \therefore \lambda = \pm \tfrac{1}{2} \end{aligned}$$

For active inequality constraints, $\lambda > 0$

$$\Rightarrow \lambda^* = \tfrac{1}{2}$$

$$\therefore \quad x^* = (-1, -1)$$

Now, we must check if it's a max/min

Find the hessian:

$$\nabla^2_{xx}\mathcal{L} = \begin{bmatrix} 2\lambda & 0 \\ 0 & 2\lambda \end{bmatrix}\Bigg|_{\lambda=\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Positive definite $\Rightarrow$ $\forall w,\ w^T \nabla^2_{xx}\mathcal{L}(x^*, \lambda^*) w > 0$

$\Rightarrow$ x^* is a strict local solution.

Example

$$\min \ -0.1\,(x_1 - 4)^2 + x_2^2$$
$$\text{ST: } x_1^2 + x_2^2 - 1 \geq 0$$

$$\mathcal{L} = -0.1\,(x_1 - 4)^2 + x_2^2 - \lambda\,(x_1^2 + x_2^2 - 1)$$

$$\nabla\mathcal{L} = \begin{pmatrix} -0.2(x_1 - 4) - 2\lambda x_1 \\ 2x_2 - 2\lambda x_2 \\ x_1^2 + x_2^2 - 1 \end{pmatrix} = 0$$

- $-0.2x_1 + 0.8 - 2\lambda x_1 = x_1(-0.2 - 2\lambda) + 0.8 = 0$ ①
- $x_2(1 - \lambda) = 0$ ②
- $x_1^2 + x_2^2 - 1 = 0$ ③

From ②:

$x_2 = 0$ or $\lambda = 1$

If $\lambda = 1$:

①: $-2.2x_1 = -0.8$

$x_1 = \frac{-4}{11}$

③: $x_2^2 = \frac{105}{121}$

$x_2 = 0.9315$

$\therefore \left(\frac{-4}{11}, 0.9315, 1\right)$

If $x_2 = 0$:

③: $x_1^2 - 1 = 0$

$\therefore x_1 = \pm 1$

If $x_1 = 1$

① $0.6 - 2\lambda = 0$

$\lambda = 0.3$ $\qquad (1, 0, 0.3)$

If $x_1 = -1$

① $1 + 2\lambda = 0$

$\lambda = -\frac{1}{2}$

disregard as $\lambda > 0$

Thus, we have:

$(1, 0, 0.3)$ ①

$(-0.3637, 0.9315, 1)$ ②

Now,
$$\nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) = \begin{bmatrix} -0.8 & 0 \\ 0 & 1.4 \end{bmatrix}$$

$C(x^*) = x_1^2 + x_2^2 - 1$ $\qquad \nabla C(x^*) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$

At ①: $C(x^*) = 1 - 1 = 0 \in A(x^*)$ $\qquad$ At ②: $C(x^*) = 0.36^2 + 0.93^2 - 1 = 0 \in A(x^*)$

For ①

$w \in F_2 \Rightarrow w^T \nabla C(x_1^*) = 0$

$\nabla C(x^*) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ let $w = [w_1, w_2]$

$w^T \nabla C(x^*) = 2w_1 + 0w_2 = 0$

Let $w_1 = 0$, $w_2 \in \mathbb{R}$, $w_2 \neq 0$

eg. $w = (0, w_2)$

Now, $(0, w_2) \begin{bmatrix} -0.8 & 0 \\ 0 & 1.4 \end{bmatrix} \begin{bmatrix} 0 \\ w_2 \end{bmatrix} = 1.4 w_2^2 > 0$

$\therefore x^* = (1, 0)$ strict solution.

For ②:

$\nabla C(x^*) = \begin{bmatrix} -0.7272 \\ 1.83 \end{bmatrix}$ let $w = (w_1, w_2)$

$$(w_1 \; w_2)\begin{bmatrix} -0.7272 \\ 1.83 \end{bmatrix} = -0.7272w_1 + 1.83w_2 = 0$$

$$\Rightarrow 1.83w_2 = 0.7272w_1$$
$$w_2 = 0.3974w_1$$

eg. if $w_1 = 1$ $w_2 = 0.3974$
Then satisfied

Now, $(1, 0.3974)\begin{bmatrix} -0.8 & 0 \\ 0 & 1.4 \end{bmatrix}\begin{pmatrix} 1 \\ 0.3974 \end{pmatrix} = -0.5789 < 0$

$\therefore$ not a solution.

Algorithms For Nonlinear Constrained Optimization

We now consider methods for solving the non-linear problem:

$$\min_{x \in \mathbb{R}^n} f(x) \quad ST: \begin{cases} c_i(x) = 0 & i \in E \\ h_j(x) \geq 0 & j \in I \end{cases}$$

f, c_i, h_j are all smooth, real valued functions on $\mathbb{R}^n$
$c = (c_1 \ldots c_m)^T \qquad h = (h_1 \ldots h_n) \qquad m \leq n$

Unconstrained Sequential Methods

1. Quadratic Penalty Method

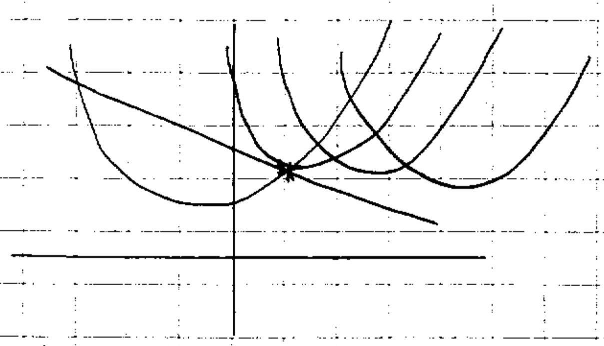
We transform our original problem into:

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{\varepsilon} \{ \|\max\{-h(x), 0\}\|^2 + \|c(x)\|^2 \}$$

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{\varepsilon} \sum_{i \in E} c_i^2(x) + \frac{1}{\varepsilon} \sum_{j \in I} \left([h_j(x)]^- \right)^2$$

where $[h_j(x)]^-$ denotes $\max\{-h_j(x), 0\}$.

We square it in order to penalise the constraint.
If $h_j(x)$ is feasible, then $h_j(x) \geq 0$
$\Rightarrow -h_j(x) \leq 0$



= approximations
As $\varepsilon \to 0$, the penalty pushes it into the feasible region and eventually x^*

The drawback:
We need to reduce ε greatly before we reach our minimizer
Can cause ill-conditioned functions.

2. Logarithmic Barrier Method

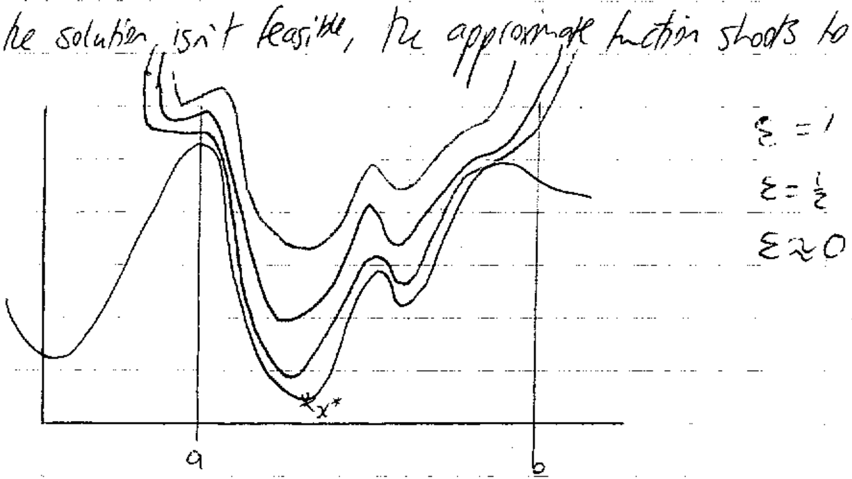
This method only applies when only inequality constraints are present, that are bonded

eg. $a \leq x \leq b$

$$\min_{x \in \mathbb{R}^n} f(x) - \varepsilon \sum_{i=1}^{k} \ln(h_i(x))$$

Noting, $\ln(0) = \infty$

When the solution isn't feasible, the approximate function shoots to ∞.



It can be seen that the approximates shoot to ∞ as ε gets closer to 0.
It doesn't approximate the function well except at near x^*

3. The Augmented Lagrangian Method

This method only applies when only equality constraints exist.

We know our Lagrangian: $\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x)$

The augmented one:

$$\mathcal{L}_a(x, \lambda, \varepsilon) = \mathcal{L}(x, \lambda) + \frac{1}{\varepsilon} \|c(x)\|^2.$$

$$\min_{x \in \mathbb{R}^n} f(x) - \lambda^T c(x) + \frac{1}{\varepsilon} \|c(x)\|^2$$

How will we get tested on this ~~stuff~~? Tuts?

Pros: • Convergence is much faster than the quadratic penalty method
• We don't need $\varepsilon \to 0$ to converge.
• Well behaved generally.

Unconstrained Exact Penalty Method (UEPM)

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{\varepsilon}\left(\sum_{i=1}^{k} \max(-h_i(x), 0) + \sum_{j=1}^{m} |c_j(x)| \right)$$

Sequential Quadratic Programming Methods (SQM)

It's a generalisation of constrained optimization of Newton's method, for unconstrained, by minimizing a quadratic approximation of the model, and a linear approximation of $c(x)$.

eg. $\min f(x)$ ST $c(x) = 0$

A penalty function method is an indirect way of attempting to solve this. Although $f(x)$, $c(x)$ are approximated separately, there's a connection between the quadratic programming problem of $f(x)$ and $c(x)$, and the Newton iteration of the Lagrangian $\mathcal{L}(x,\lambda) = f(x) - \lambda^T c(x)$.

PROOF

$$\mathcal{L}(x,\lambda) = f(x) - \lambda^T c(x).$$

The KKT first order NOC of this implies:

$$\nabla \mathcal{L}(x,\lambda) = \begin{pmatrix} \nabla_x \mathcal{L} \\ \nabla_\lambda \mathcal{L} \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \lambda^T \nabla c(x) \\ c(x) \end{pmatrix} = 0$$

The Newton iteration is:

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \begin{pmatrix} d_x^k \\ d_\lambda^k \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \begin{pmatrix} x - x^k \\ \lambda - \lambda^k \end{pmatrix}$$

Approximating $\mathcal{L}$ at $\begin{pmatrix} x_k + d_k \\ \lambda_k + d_k \end{pmatrix}$ yields:

$$\mathcal{L}(x^k + d, \lambda^k + d) = \mathcal{L}(x^k, \lambda^k) + \nabla \mathcal{L}(x^k, \lambda^k)^T d + \tfrac{1}{2} d^T \nabla^2 \mathcal{L}(x^k, \lambda^k) d$$

$$\Rightarrow \nabla_d \mathcal{L} = \nabla \mathcal{L}(x^k, \lambda^k) + \nabla^2_{xx} \mathcal{L}(x^k, \lambda^k) d = 0$$

$$\Rightarrow \nabla \mathcal{L}(x^k, \lambda^k) = \begin{pmatrix} \nabla f(x) - \nabla_x c(x^k)^T \lambda^k \\ c(x^k) \end{pmatrix}$$

$$\Rightarrow \nabla^2 \mathcal{L}(x^k, \lambda^k) = \begin{pmatrix} \nabla^2_{xx} \mathcal{L} & -\nabla_x c(x) \\ \nabla_x c(x) & 0 \end{pmatrix}$$

$$\Rightarrow \nabla^2 \mathcal{L}(x^k, \lambda_k) d = -\nabla \mathcal{L}(x^k, \lambda^k)$$

Solve for d:

$$\begin{pmatrix} \nabla_{xx}^2 \mathcal{L} & -\nabla c(x) \\ \nabla c(x) & 0 \end{pmatrix} \begin{pmatrix} x - x^k \\ \lambda - \lambda^k \end{pmatrix} = \begin{pmatrix} \nabla f(x^k) - \nabla c(x)^T \lambda \\ c(x^k) \end{pmatrix} \qquad (8)$$

Solving for d here yields the Newton step.
By the SQM,

$$\min_x \ \tfrac{1}{2} d^T \nabla_{xx}^2 \mathcal{L}(x_k, \lambda^k) d + \nabla_x f(x^k) - \nabla_x c(x)^T \mu \qquad \mu = \text{multiplier}$$

ST:

$$\nabla c(x)^T d + c(x) = 0$$

Now, $F(d, \mu) = \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k) d + \nabla f(x^k)^T d - \mu^T(\nabla_x c(x)^T d + c(x^k))$

$$\Rightarrow \nabla_d F = \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k)^T d + \nabla f(x^k) - \nabla_x c(x^k)^T \mu = 0$$

$$\Rightarrow \nabla_\mu f = \nabla_x c(x^k)^T d + c(x^k) = 0$$

Written in matrix form:

$$\begin{pmatrix} \nabla_{xx}^2 \mathcal{L} & -\nabla c(x^k)^T \\ \nabla c(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ \mu \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ c(x^k) \end{pmatrix} \qquad (9)$$

Add and subtract $c(x)^T \lambda$ to (8), we see:

$$\begin{pmatrix} d^k \\ \mu^k \end{pmatrix} = \begin{pmatrix} d_x^k \\ \lambda^{k+1} \end{pmatrix}$$