1 16th of November 2018 — A. Frangioni

What happens when the Hessian isn't strictly positive definite? If there are some negative eigenvaues, I can desume that there are some directions in which the function goes to $-\infty$. The model has no minimum, unless we restrict to a **compact set**.

In particular, we may decide to restrict to a part of the space where we can trust our model, which is called **trust region**.

Finding such a region is a NP-hard problem, if we don't restrict to a ball.

Definition 1.1 (Karush-Khun-Tucker conditions)**.** *Any optimal solution of the problem x^{i+1} must satisfy that $\equiv \exists\, \lambda \geq 0$ s.t.:*

KARUSH: $[H^i + \lambda I]x^{i+1} = -\nabla f(x^i)$ *[linear];*

KUHN: $H^i + \lambda I \ \succeq\ 0$ *[semidefinite];*

TUCKER: $\lambda(r - \|x^{i+1}\|) = 0$ *[nonlinear].*

Where the last condition means that λ has to be 0, unless the solution we get is exaclty on the border of the ball.

What's the difference between this approach and what we used to do before? We have two different cases:

- $\|x^{i+1}\| < r \Longrightarrow \lambda = 0 \Longrightarrow$ normal Newton step ($\mathcal{T}$ has no effect);
- $\lambda > 0(=$ small radius r$) \Longrightarrow$ like in line search with $\varepsilon^i = \lambda$.

The problem is computing these systems of equations taking less than $O(n^3)$.

Key idea

We don't need to compute the Hessian, we can use the first order information to infer things on the second order matrix, although we don't really need to compute the Hessian.

1.1 Quasi-newton methods

At a step we have: $m^i(x) = \nabla f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^T H^i(x - x^i)$, $x^{i+1} = x^i + \alpha^i d^i$.

At the next step we recompute the gradient in x^{i+1} and the matrix H^{i+1}: $m^{i+1}(x) = \nabla f(x^{i+1})(x - x^{i+1}) + \frac{1}{2}(x - x^{i+1})^T H^{i+1}(x - x^{i+1})$.

How should H^i be chosen? It should satisfy the following:

1. positive definite ($H^{i+1} \succ 0$);
2. we know the gradient in the previous point, we know the gradient in the current point, we construct H^{i+1} such that the model works: $\nabla m^{i+1}(x^i) = \nabla f(x^i)$;

3. $\|H^{i+1} - H^i\|$ is “small”.

This new model isn't too different from the previous one, because of the third preerty. Or equivalently $H^{i+1}(\, x^{i+1} - x^i \,) = \nabla f(x^{i+1}) - \nabla f(x^i)$, which we call **secant equation** and we denote (S).

To ease notation we define s^i such that: $s^i = x^{i+1} - x^i = \alpha^i d^i$ and $y^i = \nabla f(x^{i+1}) - \nabla f(x^i)$. s^i is chosen, while y^i is decided by the function.

In order to have a matrix H^i that satifies the first two condition we could check that $H^{i+1}s^i = y^i$, because this implies $s^i y^i = {(s^i)}^T H^{i+1} s^i$ and this implies 1. and 2., hence we obtain the **curvature condition** $s^i y^i > 0$.

Theorem 1.1. *Wolf condition implies $s^i y^i > 0$, using the notation we inroduced: (W) $\implies$ (C).*

Proof.

$$\varphi'(\alpha^i) = \nabla f(x^{i+1}) d^i \geq m_3 \varphi'(0) = m_3 \nabla f(x^i) d^i$$

$$\Downarrow$$

$$(\nabla f(x^{i+1}) - \nabla f(x^i)) d^i \geq (m_3 - 1)\varphi'(0) > 0$$

□

Observation 1.1. *We may observe that this theorem implies that if we perform Armijo Wolf exact line search condition (C) can always be satisfied.*

1.1.1 Davidson-Fletcher-Powell

How can we choose a H^i that satisfies the three conditions enumerated above? Taking $H^{i+1} = argmin\{\|H - H^i\| \ : \ H \in \text{(S)}, \ H \succeq 0\}$ is a good idea and for this minimum problem holds the following:

Theorem 1.2 (Davidson-Feltcher-Powell)**.** *The new matrix is obtained at each step constructing a rank two matrix, obtained from H^i as a rank to correction, as follows: $H^{i+1} = (I - \rho^i y^i {(s^i)}^T) H^i (I - \rho^i s^i {(y^i)}^T) + \rho^i y^i {(y^i)}^T$*

Let us denote $B^i = {H^i}^{-1}$. At any step we need to compute $B^{i+1} = {(H^{i+1})}^{-1}$, because we need to solve the system. We have some fomulas that give us a way to compute ${(H^{i+1})}^{-1}$ from ${H^i}^{-1}$.

Theorem 1.3 (Sherman-Morrison-Woodbury)**.** *The inverse of a matrix of the form $A + ab^T$ has the following shape: ${(A + ab^T)}^{-1} = \frac{A^{-1} - A^{-1}ab^T A^{-1}}{1 - b^T A^{-1} a}$.*

Observation 1.2. *From Theorem 1.3 we can conclude that $B^{i+1} = \frac{B^i + \rho^i s^i {(s^i)}^T - B^i y^i {(y^i)}^T B^i}{{(y^i)}^T B^i y^i}$.*

It's important to notice that this operation has a cost of $O(n^2)$.

We can do better, in terms of computational complexity.

1.1.2 Broyden-Fletcher-Goldfarb-Shanno

We can use directly B^i, the inverse of H^i. Write (S) for B^{i+1}: $s^i = B^{i+1}y^i \implies B^{i+1} = \text{argmin} \{\|B - B^i\| \ : \ \dots \}$.

$B^{i+1} = B^i + \rho^i[(1 + \rho^i (y^i)^T B^i y^i) s^i (s^i)^T - (B^i y^i (s^i)^T + s^i (y^i)^T B^i)]$

This formula proves to be more stable than the other one.

This method takes $O(n^2)$.

The two B^is, obtained from DFP and BFGS, are different although both sensible, but we can use a convex combination of the two.

Observation 1.3. *How can we choose H^1? The value we choose will make a differencein the results, at least for the first steps.*

Let us see a couple of choices for B^1:

- scalar multiples of identity, but how to choose the scalar?
- compute the gradient in every direction and approximate H. This will cost $O(n^3)$, but it should be done only once.

Let us compute the space needed to store the B^is: order of n^2 is still a lot. What happens if we restrict to working with information of the last k operations?

1.1.3 Poorman's approach

At each step we only consider B^{i-k} and k rank one operations. This operations cost n each, and we have k lines. The problem is that B^{i-k} takes $O(n^2)$ space. We can optimize if we choose B^{i-k} to be simpler, say a multiple of the identity, or finite difference of the gradient. Then the space complexity is $O(kn)$.

I need to tune the algorithm to find the right k which gives me enough precision and also keeps the computational cost low.

Final observation of quasi Newton methods

We may notice that this variation of Newton method doesn't get trapped in local minima, as Newton method did. In the end, the fact that quasi Newton isn't that precise at the beginning may be a good feature.

1.2 Conjugate gradient method

Do you recall?

In the gradint method, the angle between two consecutive directions is exactly 90°, as can be seen in Figure 1.1.

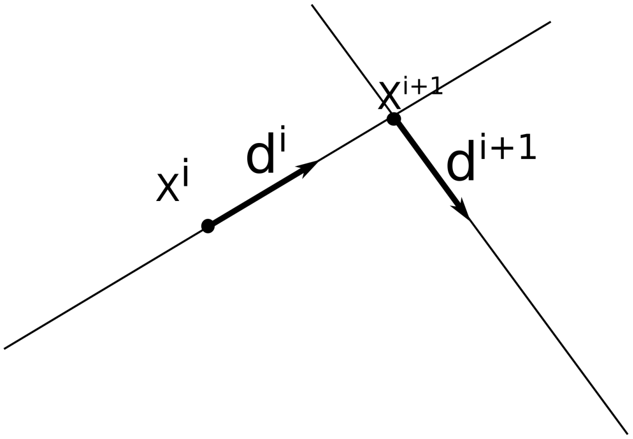

FIGURE 1.1: Geometric idea on how the new direction is chosen.

We would like to take into account not only the subspace spanned by d^{i+1} but we would like to optimize over larger and larger subspaces (spanned by d^i and d^{i+1}).

Definition 1.2 (Q-conjugate)**.** *Let v and w be vectors in $\mathbb{R}$. We say that v ad w are Q-**conjugate** if $(v)^T Qw = 0$.*

We would like pick a direction to be Q-conjugate with all the previous iterations. The point is that we can't take into account all the previous directions, but we will see that we only need the previous direction to obtain all the information we need.

ALGORITHM 1.1 Pseudocode for conjugate gradient method for quadratic functions.

```
procedure CGQ($Q, q, x, \varepsilon$)
    $d^- \leftarrow 0$;
    while ($\|\nabla f(x)\| > \varepsilon$) do
        if ($d^- = 0$) then
            $d \leftarrow -\nabla f(x)$;
        else
            $\beta = (\nabla f(x)^T Qd^-)/((d^-)^T Qd^-)$;
            $d \leftarrow -\nabla f(x) + \beta d^-$;
        end if
        $\alpha \leftarrow (\nabla f(x)^T d)/(d^T Qd)$;
        $x \leftarrow x + \alpha d$;
        $d^- \leftarrow d$;
    end while
end procedure
```

The number of iterations needed to converge is proportional to the clusterization of the eigenvalues of the matrix Q.

The algorithm that was presented is for quadratic functions, but the same algorithm works for non quadratic function as well, as long as we change the fomula for β.

The pseudocode of Algorithm 1.2 is referred to Fletcher-Reeves definition of $\beta^i = \|\nabla f(x^i)\| / \|\nabla f(x^{i-1})\|^2$.

This algorithm converges in at most n iterations.

Algorithm 1.2 Pseudocode for conjugate gradient method for arbitrary functions

```
procedure CGA($Q, q, x, \varepsilon$)
    $\nabla f^- = 0$;
    while ($\|\nabla f(x)\| > \varepsilon$) do
        if ($\nabla f^- = 0$) then
            $d \leftarrow -\nabla f(x)$;
        else
            $\beta = \|\nabla f(x^i)\|^2 / \|\nabla f^-\|^2$;
            $d \leftarrow -\nabla f(x) + \beta d^-$;
        end if
        $\alpha \leftarrow \mathrm{AWLS}(f(x + \alpha d))$;
        $x \leftarrow x + \alpha d$;
        $d^- \leftarrow d$;
        $\nabla f^- \leftarrow \nabla f(x)$;
    end while
end procedure
```

We have three different formulas for β^i, which coincide in the quadratic case.

1. Polak-Ribière: $\beta^i = \frac{\nabla f(x^i)^T (\nabla f(x^i) - \nabla f(x^{i-1}))}{\|\nabla f(x^{i-1})\|^2}$
2. Hestenes-Stiefel: $\beta^i = \frac{\nabla f(x^i)^T (\nabla f(x^i) - \nabla f(x^{i-1}))}{(\nabla f(x^i) - \nabla f(x^{i-1}))^T d^{i-1}}$
3. Dai-Yuan: $\beta^i = \frac{\|\nabla f(x^i)\|^2}{(\nabla f(x^i) - \nabla f(x^{i-1}))^T d^{i-1}}$

Some of these algorithms require some hypothesis on the function in order for the conjugate method to converge.

1. Fletcher-Reeves requires $m_1 < m_2 < \frac{1}{2}$ for (A) $\cap$ (W') to work;
2. (A) $\cap$ (W') $\not\Longleftarrow$ d^i of Polak-Ribière is of descent, unless $\beta^i_{PR} = \max\{\beta^i, 0\}$.

Sometimes it's important to restart from scratches if the algorithm isn't converging, because many bad choices may lead to a bad result.

The idea of taking the gradient and modify it instead of multiplying by a factor, adding the previous direction.

It's possibile to design hybrids between quasi-Newton and conjugate method.