OSM Boot Camp **Math Notes**

Harrison Beard

Summer 2018

Mon, 23 Jul. 2018

- **Topic**. Nonlinear Optimization.
  -

    $$\xcancel{\begin{array}{rl} \max\limits_{\mathbf{x}} & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \preceq \mathbf{b} \\ & \mathbf{x} \succeq \mathbf{0} \end{array}}$$

    $$\Downarrow$$

    $$\text{nonlinear } \min_{\mathbf{x}} f : \mathbb{R}^n \to \mathbb{R}$$

    Start with a guess $\mathbf{x}_0$. This yields through the algorithm an $\mathbf{x}_1$, then $\mathbf{x}_1 \mapsto \mathbf{x}_2$, and $\mathbf{x}_2 \mapsto \mathbf{x}_3$, and so forth. Eventually we get convergence. This is all according to the rule

    $$\mathbf{x}_{i+1} = f(\mathbf{x}_i).$$

    Typically, f does one of two things: It could move in a direction that decreases the objective function (**descent function**) or it could approximate the objective function near $\mathbf{x}_i$ with some simpler function, and then that function itself is then optimized (**local approximation methods**).
- **Topic**. Convergence.
  - What does it look like?

(*i*). $\|\mathbf{x}_{i+1} - \mathbf{x}_i\| < \varepsilon$

(*ii*). $\frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{\|\mathbf{x}_i\|} < \varepsilon$

(*iii*). $\|\mathbf{D}f(\mathbf{x}_i)\| < \varepsilon$, by the FONC

(*iv*). $|f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i)| < \varepsilon$

  - Quadratic Optimization: f is minimized where g is minimized where

    $$g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{Q}\mathbf{x} - \mathbf{b}^{\mathsf{T}}\mathbf{x} + c,$$

    where $\mathbf{Q} = \mathbf{A}^{\mathsf{T}} + \mathbf{A}$. A minimizer exists only if $\mathbf{Q} > 0$. The minimizer is the solution to $\mathbf{Q}\mathbf{x} = \mathbf{b}$, and $\mathbf{0} = \mathbf{D}g(\mathbf{x}) = \mathbf{Q}\mathbf{x} - \mathbf{b}$.
  - In general, we find a solution to the linear system of equations of n equations with n unknowns:

(*i*). LU-Decomposition

(*ii*). QR-Decomposition

(*iii*). Cholesky

  - All the above algorithms are $\mathcal{O}(n^3)$ in time.
- **Topic**. Standard Least Squares.

– For $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathrm{M}_{m \times n}(\mathbb{R})$, the problem of finding an $\mathbf{x}^* \in \mathbb{R}^n$ to minimize $\|\mathbf{Ax} = \mathbf{b}\|_2$ is the same as minimizing

$$\mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{x} - 2\mathbf{A}\mathbf{x}\mathbf{b}.$$

Note

$$\langle \mathbf{Ax} - \mathbf{b}, \mathbf{Ax} - \mathbf{b} \rangle = \mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{x} - 2\mathbf{A}\mathbf{x}\mathbf{b} + \mathbf{b}^\mathsf{T}\mathbf{b}.$$

We also have

$$\mathbf{A}^\mathsf{T}\mathbf{A} \ = \mathbf{A}^\mathsf{T}\mathbf{b}.$$

The solution is the same as minimizing $g(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{x} - 2\mathbf{A}\mathbf{x}\mathbf{b}$:

$$\begin{aligned} \mathbf{0} &= \mathbf{D}g(\mathbf{x}) \\ &= \mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{x} \\ &= \mathbf{A}^\mathsf{T}\mathbf{b}. \end{aligned}$$

- **Topic**. Gradient Descent.
  – Move in the direction of $-\mathbf{D}f^\mathsf{T}(\mathbf{x}_i)$, the direction of steepest descent. The new approximation:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha \mathbf{D}f^\mathsf{T}(\mathbf{x}_i),$$

for some value of α. To choose α_i, choose

$$\alpha_i^* = \arg\min_{\alpha_i} f\left(\mathbf{x}_i - \alpha \mathbf{D}f^\mathsf{T}(\mathbf{x}_i)\right),$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i^* \mathbf{D}f^\mathsf{T}(\mathbf{x}_i).$$

This policy of proceeding down the surface is called **steepest descent.**

- **Topic**. Newton's Method: multivariate version. Note that the Hessian $\mathbf{D}^2 f$ has to be positive definite.

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left(\mathbf{D}^2 f(\mathbf{x}_i)\right)^{-1} \mathbf{D}f^\mathsf{T}(\mathbf{x}_i).$$

Converges quadratically.

– Problems with Newton:

(i). If $\mathbf{x}_0$ is too far from $\mathbf{x}^*$.

(ii). When $\mathbf{D}^2 f(\mathbf{x}_i)$ is not positive definite $\left(\mathbf{D}^2 f(\mathbf{x}_i) \not> 0\right)$.

(iii). When $\left(\mathbf{D}^2 f(\mathbf{x}_i)\right)^{-1} \mathbf{D}f^\mathsf{T}(\mathbf{x}_i)$ is too expensive to compute or unstable, or impossible.

Wed, 25 Jul. 18

Conjugate Gradient Methods.

- Different than Quasi Newton Method in that they don't store the $n \times n$ Hess (or approximations)
- Most useful when obj. fn. is of form:
$$\frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{Q}\mathbf{x} - \mathbf{b}^{\mathsf{T}}\mathbf{x} + c,$$
where $\mathbf{Q}$ is symmetric, $\mathbf{Q} > 0$, and $\mathbf{Q}$ is sparse (most of the entries are zero).
- Each step of Conj. Grad. has temporal and spatial complexity $\mathcal{O}(m)$, where m is the number of nonzero entries.

Nonlinear Least Squares.

- Of the form
$$f = \mathbf{r}^{\mathsf{T}}\mathbf{r}.$$

1. If the dimension is not too big:
   (a) if $\mathbf{x}_0$ is close to $\mathbf{x}^*$:
      i. If computing $\left(\mathbf{D}^2 f(\mathbf{x})\right)^{-1}\mathbf{D}f^{\mathsf{T}}(\mathbf{x})$ is cheap and feasible, then use Newton's.
      ii. Else,
         - If $f = \mathbf{r}^{\mathsf{T}}\mathbf{r}$, use Gauss-Newton.
         - Use BFGS.
   (b) Else, use a gradient descent until you get a better "$\mathbf{x}_0$".
   (c) If all other methods are not converging rapidly, then try conjugate gradient.
2. If dimension large and Hess sparse, use conj. grad.

Gradient Methods.

Proposition 9.2.1.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that is differentiable at $\mathbf{x} \in \mathbb{R}^n$. Among all unit vectors in $\mathbb{R}^n$, the unit vector $\mathbf{u} \in \mathbb{R}^n$ has the gradient directional derivative $\mathbf{D}_{\mathbf{u}}f(\mathbf{x})$ at $\mathbf{x}$ and has the normalized gradient
$$\mathbf{u} = \mathbf{D}f(\mathbf{x})^{\mathsf{T}} / \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\|.$$

□ **Proof.** By C-S, for $\mathbf{u} \in \mathbb{R}^n$, we have
$$\begin{aligned} |\mathbf{D}f_{\mathbf{u}}(\mathbf{x})| &= |\mathbf{D}f(\mathbf{x})\mathbf{u}| \\ &= \left|\left\langle \mathbf{D}f(\mathbf{x})^{\mathsf{T}}, \mathbf{u}\right\rangle\right| \\ &\leq \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\|. \end{aligned}$$

But if we let $\mathbf{u} = \mathbf{D}f(\mathbf{x})^{\mathsf{T}} / \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\|$ we have
$$\begin{aligned} \mathbf{D}f_{\mathbf{u}}(\mathbf{x}) &= \left\langle \mathbf{D}f(\mathbf{x})^{\mathsf{T}}, \mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\rangle / \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\| \\ &= \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\|, \end{aligned}$$

so the normalized gradient $\mathbf{u} = \mathbf{D}f(\mathbf{x})^{\mathsf{T}} / \left\|\mathbf{D}f(\mathbf{x})^{\mathsf{T}}\right\|$ maximizes the directional derivative. ■