CONJUGATE GRADIENTS

MAKSIM LEVENTAL

Part 1. Quadratic Forms

The gradient of quadratic form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} - \mathbf{b}^t\mathbf{x} + \mathbf{c}$$

is

$$\begin{aligned}\nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} &= \left[\begin{array}{c}\frac{\partial}{\partial x_1} f \\ \vdots \\ \frac{\partial}{\partial x_n} f\end{array}\right]\Bigg|_{\mathbf{x}=\mathbf{x}_0} \\ &\equiv \nabla f(\mathbf{x}_0) \\ &= \frac{1}{2}\left(\mathbf{A}^t + \mathbf{A}\right)\mathbf{x}_0 - \mathbf{b}\end{aligned}$$

If $\mathbf{A}^t = \mathbf{A}$ then

$$\nabla f(\mathbf{x}_0) = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$$

hence minimizing f is equivalent to solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ and vice-versa. If $\mathbf{A}^t = \mathbf{A}$ and positive semi-definite and $\mathbf{x}$ is such that $\mathbf{A}\mathbf{x} = \mathbf{b}$ and let $\mathbf{y} = \mathbf{x} + \boldsymbol{\delta}$. Then

$$\begin{aligned}f(\mathbf{y}) &= \frac{1}{2}(\mathbf{x}+\boldsymbol{\delta})^t\mathbf{A}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{b}^t(\mathbf{x}+\boldsymbol{\delta}) + \mathbf{c} \\ &= \frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\mathbf{x} + \frac{1}{2}\mathbf{x}^t\mathbf{A}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\boldsymbol{\delta} - \mathbf{b}^t\mathbf{x} - \mathbf{b}^t\boldsymbol{\delta} + \mathbf{c} \\ &= \frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} + \boldsymbol{\delta}^t\mathbf{A}\mathbf{x} + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\boldsymbol{\delta} - \mathbf{b}^t\mathbf{x} - \mathbf{b}^t\boldsymbol{\delta} + \mathbf{c} \text{ by symmetry of } \mathbf{A} \\ &= \frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} + \boldsymbol{\delta}^t\mathbf{b} + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\boldsymbol{\delta} - \mathbf{b}^t\mathbf{x} - \mathbf{b}^t\boldsymbol{\delta} + \mathbf{c} \\ &= \frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} - \mathbf{b}^t\mathbf{x} + \mathbf{c} + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\boldsymbol{\delta} \\ &= f(\mathbf{x}) + \frac{1}{2}\boldsymbol{\delta}^t\mathbf{A}\boldsymbol{\delta} \geq f(\mathbf{x}) \text{ by positive semi-definiteness of } \mathbf{A}\end{aligned}$$

Therefore for positive semi-definite and symmetric f there exists a unique minimum at $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Part 2. Gradient Descent

How to minimize f? Most naive strategy is to pick an arbitrary point $\mathbf{x}_0$ and head down the gradient, i.e. head in the direction

$$-\nabla f(\mathbf{x}_0) = \mathbf{b} - \mathbf{A}\mathbf{x}_0$$

Define $\mathbf{r}_0 \equiv -\nabla f(\mathbf{x}_0)$. Then

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{r}_0$$

But how to pick how far in the direction $\mathbf{r}_0 \equiv -\nabla f(\mathbf{x}_0)$? Do a "line search" to minimize f on the line $\mathbf{x}_0 + \alpha_0 \mathbf{r}_0$, i.e. find where

$$\frac{d}{d\alpha} f(\mathbf{x}_0 + \alpha_0 \mathbf{r}_0) = 0$$

By the chain rule

$$\begin{aligned}\frac{d}{d\alpha} f(\mathbf{x}_0 + \alpha_0 \mathbf{r}_0) &= (\nabla f(\mathbf{x}_0))^t \frac{d}{d\alpha}(\mathbf{x}_0 + \alpha_0 \mathbf{r}_0) \\ &= (\nabla f(\mathbf{x}_0))^t \mathbf{r}_0\end{aligned}$$

Therefore where $\nabla f(\mathbf{x}_0)$ and $\mathbf{r}_0$ are orthogonal is minimum of f on the line. We can use this to determine α_0

$$\begin{aligned}-(\nabla f(\mathbf{x}_0))^t \mathbf{r}_0 &= 0 \\ \mathbf{r}_1^t \mathbf{r}_0 &= 0 \\ (\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \alpha_0 \mathbf{r}_0))^t \mathbf{r}_0 &= 0 \\ (\mathbf{b} - \mathbf{A}\mathbf{x}_0)^t \mathbf{r}_0 - \alpha_0 (\mathbf{A}\mathbf{r}_0)^t \mathbf{r}_0 &= 0 \Rightarrow \\ \alpha_0 (\mathbf{A}\mathbf{r}_0)^t \mathbf{r}_0 &= (\mathbf{b} - \mathbf{A}\mathbf{x}_0)^t \mathbf{r}_0 \Rightarrow \\ \alpha_0 &= \frac{\mathbf{r}_0^t \mathbf{r}_0}{\mathbf{r}_0^t \mathbf{A}\mathbf{r}_0}\end{aligned}$$

Therefore the gradient descent algorithm is a set of update rules

$$\begin{aligned}\mathbf{r}_i &= \mathbf{b} - \mathbf{A}\mathbf{x}_i \\ \alpha_i &= \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{r}_i^t \mathbf{A}\mathbf{r}_i} \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i \mathbf{r}_i\end{aligned}$$

The complexity of the algorithm is dominated by matrix multiplications $\mathbf{A}\mathbf{x}_i$ and $\mathbf{A}\mathbf{r}_i$ but we can eliminate one: multiply the last update rule by $-\mathbf{A}$ and add $\mathbf{b}$:

$$\begin{aligned}\mathbf{b} - \mathbf{A}\mathbf{x}_{i+1} &= \mathbf{b} - \mathbf{A}\mathbf{x}_i - \alpha_i \mathbf{A}\mathbf{r}_i \\ \mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{r}_i\end{aligned}$$

The disadvantage of using this recurrence relation is accumulation of floating point roundoff error in computation of $\mathbf{x}_i$, not corrected by this recurrence since it's computed isolated from $\mathbf{x}_i$. Solution is to periodically use the naive update rule for $\mathbf{r}_i$.

So what's the problem with this? Why can't we just always use gradient descent? Let

$$\begin{aligned}f(\mathbf{x}) &= \frac{1}{2}\mathbf{x}^t \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \mathbf{x} \\ &= x_1^2 + x_1 x_2 + x_2^2 + x_3^2\end{aligned}$$

with $\mathbf{x}_0 = (1, 2, 3)$. Minimizing direction $\mathbf{e}_1$ we get to $\mathbf{x}_1 = (-1, 2, 3)$ and after minimizing in direction $\mathbf{e}_2$ we get to $\mathbf{x}_2 = (-1, 1/2, 3)$. But now the minimizing in the $\mathbf{e}_1$ direction has been "messed up", in that if the line search along $\mathbf{e}_1$ is repeated

it's shifted to $x_1 = -1/4$. So the problem turns out to be that picking line search directions in this manner selects directions that interfere with each other, undo already effected minimizations. How can we pick directions that don't behave this way?

Part 3. Conjugate Directions

We want to choose directions $\mathbf{v}$ such that if we've just performed a minimization along direction $\mathbf{u}$ it won't be undone I.e. we want ∇f to be perpendicular to $\mathbf{u}$ before and after the minimization. This will be true if *the change in* ∇f *is perpendicular to* $\mathbf{u}$. Let $\mathbf{x}_i$ be the point from which we set out in the direction $\mathbf{u}$. Then we want

$$\begin{aligned}
\mathbf{u} \cdot \delta\left(\nabla f\right) &= 0 \\
\mathbf{u} \cdot \nabla f\left(\mathbf{x}_{i+1}\right) &= 0 \text{ given that } \mathbf{u} \cdot \nabla f\left(\mathbf{x}_i\right) = 0 \\
\mathbf{u} \cdot \left(\nabla f\left(\mathbf{x}_{i+1}\right) - \nabla f\left(\mathbf{x}_i\right)\right) &= 0 \\
\mathbf{u} \cdot \left(\left(\mathbf{b} - \mathbf{A}\mathbf{x}_{i+1}\right) - \left(\mathbf{b} - \mathbf{A}\mathbf{x}_i\right)\right) &= 0 \\
\mathbf{u} \cdot \left(\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_{i+1}\right) &= 0 \\
-\mathbf{u} \cdot \mathbf{A}\mathbf{v} &= 0
\end{aligned}$$

This is kind of not correct (since we don't compute the gradients the points we arrive at) but oh well it's still true. We need a set of directions $\mathbf{d}_i$ such that

$$\mathbf{d}_i \cdot \mathbf{A} \cdot \mathbf{d}_j = 0 \text{ for } i \neq j$$

Such a set of directions is called *conjugate*. So if we had in hand a set of conjugate directions we could line search all each one in sequence and minimize f (in at most number of steps equal to the number of conjugate directions). Very quickly a "technical" lemma

Lemma: If $\mathbf{A}$ is positive definite then a set of conjugate vectors/directions $\{\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}\}$ is linearly independent and hence forms a basis for $\mathbb{R}^n$.
Proof: Let

$$\sum_{i=0}^{n-1} c_i \mathbf{d}_i = 0$$

Then

$$\begin{aligned}
\mathbf{d}_i^t \mathbf{A} \sum_{i=0}^{n-1} c_i \mathbf{d}_i &= 0 \\
\sum_{i=0}^{n-1} c_i \mathbf{d}_i^t \mathbf{A} \mathbf{d}_i &= 0 \\
c_i \mathbf{d}_i^t \mathbf{A} \mathbf{d}_i &= 0
\end{aligned}$$

and hence $c_i = 0$ since $\mathbf{A}$ is positive definite.

Now we just need some conjugate directions. Gram-Schmidt to the rescue. Start

with a set of already linearly independent vectors/directions $\{\mathbf{u}_i\}$ (the standard basis vectors $\{\mathbf{e}_i\}$ will suffice) and use "conjugate" Gram-Schmidt to "$\mathbf{A}$-orthogonalize"

$$\mathbf{d}_i = \mathbf{u}_i - \sum_{k=0}^{i-1} \beta_{ik} \mathbf{d}_k$$

where β_{ik} is the standard

$$\beta_{ik} = \frac{\mathbf{u}_i \mathbf{A} \mathbf{d}_k}{\mathbf{d}_k^t \mathbf{A} \mathbf{d}_k}$$

Here's are the update rules if we have such a set $\{\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}\}$

$$\begin{aligned} \mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_i \mathbf{A} \mathbf{d}_i \\ \alpha_i &= \frac{\mathbf{d}_i^t \mathbf{r}_i}{\mathbf{d}_i^t \mathbf{A} \mathbf{d}_i} \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i \mathbf{d}_i \end{aligned}$$

The computation of $\mathbf{r}_i$ is for the purposes of the line search. So we're set right? We can cook up some conjugate directions and run the regular gradient descent right? Unfortunately this scheme has the slight flaw that all conjugate vectors to have forever be kept in memory and the quite serious flaw that all of the repetitive matrix multiplications mean an $O\left(n^3\right)$ runtime.

Part 4. Conjugate Gradients

We just need to pick a more convenient set of vectors to Gram-Schmidt-$\mathbf{A}$-orthogonalize. I have an idea (not really me): let's use gradients/residuals! Why? Well firstly if we already have a set of mutually $\mathbf{A}$-orthogonal directions $\{\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}\}$ then the gradient computed on each line search will be orthogonal (straight-up orthogonal) to preceeding line search directions. Here's some proof: in general (keep in mind we already have $\{\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}\}$ in hand)

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 \\ \mathbf{x}_2 &= (\mathbf{x}_1) + \alpha_1 \mathbf{d}_1 \\ &= (\mathbf{x}_0 + \alpha_0 \mathbf{d}_0) + \alpha_1 \mathbf{d}_1 \\ &\vdots \\ \mathbf{x}_{i+1} &= \mathbf{x}_0 + \sum_{j=0}^{i} \alpha_j \mathbf{d}_j \\ &\vdots \\ \mathbf{x}_{n-1} &= \mathbf{x}_0 + \sum_{j=0}^{n-1} \alpha_j \mathbf{d}_j \end{aligned}$$

Note that $\nabla f(\mathbf{x}_{n-1}) = 0$ since $\mathbf{x}_{n-1}$ minimizes the quadratic form, so

$$\begin{aligned}
\mathbf{A}\mathbf{x}_{n-1} - \mathbf{b} &= \mathbf{A}\mathbf{x}_0 - \mathbf{b} + \sum_{j=0}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j \\
\nabla f(\mathbf{x}_{n-1}) &= \nabla f(\mathbf{x}_0) + \sum_{j=0}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j \\
0 &= \nabla f(\mathbf{x}_0) + \sum_{j=0}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j \\
&\Rightarrow \\
\nabla f(\mathbf{x}_0) &= -\sum_{j=1}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j
\end{aligned}$$

and hence

$$\begin{aligned}
\mathbf{A}\mathbf{x}_{i+1} - \mathbf{b} &= \mathbf{A}\mathbf{x}_0 - \mathbf{b} + \sum_{j=1}^{i} \alpha_j \mathbf{A}\mathbf{d}_j \\
\nabla f(\mathbf{x}_{i+1}) &= -\sum_{j=1}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j + \sum_{j=1}^{i} \alpha_j \mathbf{A}\mathbf{d}_j \\
&= -\sum_{j=i+1}^{n-1} \alpha_j \mathbf{A}\mathbf{d}_j
\end{aligned}$$

Finally

$$\begin{aligned}
\mathbf{d}_k^t \nabla f(\mathbf{x}_{i+1}) &= -\sum_{j=i+1}^{n-1} \alpha_j \mathbf{d}_k \mathbf{A}\mathbf{d}_j \\
&= 0 \text{ by } \mathbf{A}\text{-orthogonality for } k < i
\end{aligned}$$

Intuitively this makes sense because on every line search we're choosing α_i so that it's orthogonal to the search direction (though of course this in and of itself isn't a guarantee that orthogonality will be preserved - cue the above proof). Succinctly with $\mathcal{D}_{i-1} = \text{span}(\{\mathbf{d}_0, \ldots, \mathbf{d}_{i-1}\})$ it's the case that the hyperplane/linear-variety $\mathbf{x}_0 + \mathcal{D}_i$ is tangent to the level surface of $f(\mathbf{x}_i)$, an ellipsoid, and hence the gradient $\nabla f(\mathbf{x}_i) \cdot \mathcal{D}_i = 0$. Furthermore (remember $\mathbf{r}_i \equiv \nabla f(\mathbf{x}_i)$) using the direction update rule from conjugate directions with $\mathbf{u}_i = \mathbf{r}_i$

$$\begin{aligned}
\mathbf{d}_i &= \mathbf{r}_i - \sum_{k=0}^{i-1} \beta_{ik} \mathbf{d}_k \\
&\Rightarrow \\
\mathbf{d}_i^t \mathbf{r}_j &= \mathbf{r}_i^t \mathbf{r}_j - \sum_{k=0}^{i-1} \beta_{ik} \mathbf{d}_k^t \mathbf{r}_j \\
&= \mathbf{r}_i^t \mathbf{r}_j
\end{aligned}$$

It gets even better. If we use gradients to **construct** the search directions then (remember $\mathbf{r}_i \equiv \nabla f(\mathbf{x}_i)$)

$$\text{span}(\{\mathbf{r}_0, \ldots, \mathbf{r}_{i-1}\}) = \text{span}(\{\mathbf{d}_0, \ldots, \mathbf{d}_{i-1}\})$$

remember that $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{r}_i$ so actually

$$\text{span}(\{\mathbf{r}_0, \ldots, \mathbf{r}_{i-1}\}) = \text{span}\left(\left\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{-1}\mathbf{r}_0\right\}\right)$$

This is a *Krylov* subspace, a subspace created by repeatedly applying a matrix to a vector. The use is that since $\mathbf{A}\mathcal{D}_{i-1} \subset \mathcal{D}_i$ and that $\mathbf{r}_i \cdot \mathcal{D}_i = -\nabla f(\mathbf{x}_i) \cdot \mathcal{D}_i = 0$ means that $\mathbf{r}_i \cdot \mathbf{A}\mathcal{D}_{i-1} = 0$! That means Gram-Schmidt-**A**-orthogonalization reduces to just one term (just the direction purely in $\mathcal{D}_i$) ! From Gram-Schmidt

$$\mathbf{d}_i = \mathbf{u}_i - \sum_{k=0}^{i-1} \beta_{ik} \mathbf{d}_k$$

but from conjugate directions and the result above $\mathbf{d}_i^t \mathbf{r}_j = \mathbf{r}_i^t \mathbf{r}_j$ and that $\mathbf{r}_i^t \mathbf{r}_j = 0$ if $i \neq j$

$$\begin{aligned}
\mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{d}_i \\
&\Rightarrow \\
\mathbf{r}_j^t \mathbf{r}_{i+1} &= \mathbf{r}_j^t \mathbf{r}_i - \alpha_i \mathbf{r}_j^t \mathbf{A}\mathbf{d}_i \\
&\Rightarrow \\
\alpha_i \mathbf{r}_j^t \mathbf{A}\mathbf{d}_i &= \mathbf{r}_j^t \mathbf{r}_i - \mathbf{r}_j^t \mathbf{r}_{i+1} \\
\mathbf{r}_j^t \mathbf{A}\mathbf{d}_i &= \begin{cases} \frac{1}{\alpha_i} \mathbf{r}_i^t \mathbf{r}_i & \text{if } i = j \\ -\frac{1}{\alpha_{i-1}} & \mathbf{r}_i^t \mathbf{r}_i \text{if } i = j+1 \end{cases} \\
&\Rightarrow \\
\beta_{ij} &= \begin{cases} \frac{1}{\alpha_i} \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{d}_{i-1}^t \mathbf{A}\mathbf{d}_{i-1}} & \text{if } i = j+1 \\ 0 & \text{if } i > j+1 \end{cases}
\end{aligned}$$

Simplifying further since only $\beta_{i,i-1} \equiv \beta_i$ are non-zero

$$\begin{aligned}
\beta_i &= \frac{1}{\alpha_i} \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{d}_{i-1}^t \mathbf{A}\mathbf{d}_{i-1}} \\
&= \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{d}_{i-1}^t \mathbf{r}_{i-1}}
\end{aligned}$$

Finally the entire set of update rules

$$\begin{aligned}
\mathbf{d}_0 &= \mathbf{r}_0 \\
\alpha_i &= \frac{\mathbf{r}_i^t \mathbf{r}_i}{\mathbf{d}_i^t \mathbf{A}\mathbf{d}_i} \\
\mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i \mathbf{d}_i \\
\mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{d}_i \\
\beta_{i+1} &= \frac{\mathbf{r}_{i+1}^t \mathbf{r}_{i+1}}{\mathbf{d}_i^t \mathbf{r}_i} \\
\mathbf{d}_{i+1} &= \mathbf{r}_i - \beta_{i+1} \mathbf{d}_i
\end{aligned}$$