Statistical
Computing
Winter 2001
W. Stuetzle

# The Conjugate Gradient Method

The conjugate gradient method was orininally invented to minimize a quadratic function

$$F(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T A\mathbf{x} + \mathbf{b}\cdot\mathbf{x} + c$$

with $\mathbf{x} \in R^m$ and $A$ symmetric and positive definite. This is equivalent to finding $\mathbf{x}$ with $\nabla F(\mathbf{x}) = \mathbf{0}$, i.e. solving the linear system

$$A\mathbf{x} + \mathbf{b} = \mathbf{0}.$$

Linear systems are typically solved using Gauss elimination or Choleski decomposition. The CG method is an alternative solution method. It is of interest for several reasons:

- In the CG method, the only operation involving $A$ is multiplication of $A$ with a vector. If $A$ is large and sparse, this can be advantageous, because it may not be necessary to ever store $A$ in an $m \times m$ matrix. Instead, $A$ can be stored in a compressed form or computed on the fly. One then only needs to write a special purpose function for multiplying $A$ with a vector.

- CG is an iterative method. In theory, the number of iterations needed by the GC method is equal to the number of different eigenvalues of $A$, i.e. at most $m$. To get the solution to machine precision usually requires more than $m$ iterations. In practice, a good approximation is frequently reached much earlier, which makes the CG method attractive for large problems.

- The CG method can be used to solve least squares problems with large and sparse design matrix.

- The CG method can be generalized to a minimization method for general (non quadratic) smooth functions.

## 1. Minimizing quadratic functions

Our goal is to minimize a quadratic function

$$F(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T A\mathbf{x} + \mathbf{b}\cdot\mathbf{x} + c$$

with $\mathbf{x} \in R^m$ and $A$ symmetric and positive definite. Let $\mathbf{x}_\star = -A^{-1}\mathbf{b}$ be the location of the minimum, and note that $\nabla F(\mathbf{x}) = A\mathbf{x} + \mathbf{b} = A(\mathbf{x} - \mathbf{x}_\star)$.

The basis of the CG method is the following

**Lemma 1:** If we go from some point $\mathbf{x}_0$ in direction $\mathbf{p}$ till we reach the minimum:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + \tilde{t}\,\mathbf{p} \qquad \text{with} \\ \tilde{t} &= \text{argmin}_t\, F(\mathbf{x}_0 + t\mathbf{p}) \end{aligned}$$

then $\mathbf{x}_1 - \mathbf{x}_\star$ is $A$-conjugate to $\mathbf{p}$: $\mathbf{p}^T A(\mathbf{x}_1 - \mathbf{x}_\star) = \mathbf{0}$.

**Proof:** Because $\mathbf{x}_1$ minimizes $F$ along $p$, the gradient $\nabla F(\mathbf{x}_1)$ has to be orthogonal to $\mathbf{p}$: $\mathbf{p}^T A(\mathbf{x}_1 - \mathbf{x}_\star) = 0$.

It is easy to verify that the optimal step is

$$\tilde{t} = -\frac{\mathbf{p}^T A(\mathbf{x}_0 - \mathbf{x}_\star)}{\mathbf{p}^T A\mathbf{p}} = -\frac{\mathbf{p}^T \nabla F(\mathbf{x}_0)}{\mathbf{p}^T A\mathbf{p}}.$$

**Note 1:** Because the minimum of $F$ is reached at $\mathbf{x}_\star$, Lemma 1 implies that, once we go along a direction $\mathbf{p}$ to the minimum, we can restrict further steps towards the minimum to be $A$-conjugate to $\mathbf{p}$.

The following is a simple consequence of Lemma 1:

**Lemma 2:** Let $\mathbf{p}_1, \ldots, \mathbf{p}_k$ be mutually $A$-conjugate directions. Then $F(\mathbf{x}_0 + t_1\mathbf{p}_1 + \cdots + t_k\mathbf{p}_k)$ can be minimized by stepping from $\mathbf{x}_0$ along $\mathbf{p}_1$ to the minimum $\mathbf{x}_1$, stepping from $\mathbf{x}_1$ along $\mathbf{p}_2$ to the minimum $\mathbf{x}_2$, etc.

**Note 2:** Because $\mathbf{x}_k$ minimizes $F$ among all vectors of the form $\mathbf{x}_0 + \sum_{i=1}^{k} t_i\mathbf{p}_i$, $\nabla F(\mathbf{x}_k) \cdot \mathbf{p}_i = \mathbf{p}_i^T A(\mathbf{x}_k - \mathbf{x}_\star) = 0$ for $i = 1, \ldots, k$. Therefore $\mathbf{x}_k - \mathbf{x}_\star$ is $A$-conjugate to $\mathbf{p}_1, \ldots, \mathbf{p}_k$.

As there are only $m$ mutually conjugate directions, Lemma 2 shows that the following algorithm finds the minimum of $F$:

**Alg 1:** Pick $\mathbf{p}_1, \ldots, \mathbf{p}_m$ mutually $A$-conjugate, and starting guess $\mathbf{x}_0$.

For $i = 1$ to $m$ {
$\tilde{t} = -\mathbf{p}_i \cdot (A\mathbf{x}_{i-1} + \mathbf{b}) \;/\; (\mathbf{p}_i^T A\mathbf{p}_i)$
$\mathbf{x}_i = \mathbf{x}_{i-1} + \tilde{t}\mathbf{p}_i$
}
Return $\mathbf{x}_m$

The conjugate gradient method is Alg 1 with a particular choice of $\mathbf{p}_1, \ldots, \mathbf{p}_m$. Let $\mathbf{g}_k = -\nabla F(\mathbf{x}_k) = -(A\mathbf{x}_k + \mathbf{b})$ be the negative gradient at $\mathbf{x}_k$. The CG method picks $\mathbf{p}_{k+1}$ as the component of $\mathbf{g}_k$ $A$-conjugate to $\mathbf{p}_1, \ldots, \mathbf{p}_k$:

$$\mathbf{p}_{k+1} = \mathbf{g}_k - \sum_{i=1}^{k} \frac{\mathbf{g}_k^T A\mathbf{p}_i}{\mathbf{p}_i^T A\mathbf{p}_i}\, \mathbf{p}_i$$

This makes sense: At each step we try to go downhill as steeply as possibly, subject to the step being conjugate to the previous steps.

What makes the GC method work nicely is that in the above equation,

$$\mathbf{g}_k^T A\mathbf{p}_i = 0 \qquad \text{for } i = 1, \ldots, k-1;$$

thus $\mathbf{p}_{k+1}$ is a linear combination only of $\mathbf{g}_k$ and $\mathbf{p}_k$. We will now show this fact.

**Lemma 3:** $[\mathbf{p}_1, \ldots, \mathbf{p}_s] = [\mathbf{g}_0, \ldots, \mathbf{g}_{s-1}] = [A(\mathbf{x}_0 - \mathbf{x}_\star), A^2(\mathbf{x}_0 - \mathbf{x}_\star), \ldots, A^s(\mathbf{x}_0 - \mathbf{x}_\star)]$

**Proof:**

$$\begin{aligned}
\mathbf{g}_0 &= -A(\mathbf{x}_0 - \mathbf{x}_\star) \\
\mathbf{p}_1 &= -A(\mathbf{x}_0 - \mathbf{x}_\star) \\
\mathbf{x}_1 &= \mathbf{x}_0 + t_1\mathbf{p}_1 \\
\mathbf{g}_1 &= -A(\mathbf{x}_1 - \mathbf{x}_\star) = -A(\mathbf{x}_0 - \mathbf{x}_\star) + t_1 A^2(\mathbf{x}_0 - \mathbf{x}_\star) \\
\mathbf{p}_2 &= \text{lincom of } \mathbf{g}_1, \mathbf{p}_1 \Rightarrow \mathbf{p}_2 \in [A(\mathbf{x}_0 - \mathbf{x}_\star), A^2(\mathbf{x}_0 - \mathbf{x}_\star)] \\
\mathbf{g}_2 &= -A(\mathbf{x}_2 - \mathbf{x}_\star) \in [A(\mathbf{x}_0 - \mathbf{x}_\star), A^2(\mathbf{x}_0 - \mathbf{x}_\star), A^3(\mathbf{x}_0 - \mathbf{x}_\star)] \\
\mathbf{p}_3 &= \text{lincom of } \mathbf{g}_1, \mathbf{p}_1, \mathbf{p}_2 \Rightarrow \mathbf{p}_3 \in [A(\mathbf{x}_0 - \mathbf{x}_\star), A^2(\mathbf{x}_0 - \mathbf{x}_\star), A^3(\mathbf{x}_0 - \mathbf{x}_\star)],
\end{aligned}$$

and so on. This could easily be made more formal by induction.

**Note 3:** Lemma 3 also shows that the gradients $\mathbf{g}_0, \mathbf{g}_1, \ldots$ are mutually orthogonal: $\mathbf{g}_k \perp \mathbf{p}_1, \ldots, \mathbf{p}_k$ and $[\mathbf{p}_1, \ldots, \mathbf{p}_k] = [\mathbf{g}_1, \ldots, \mathbf{g}_{k-1}]$.

The following lemma gives the desired result:

**Lemma 4:** $\mathbf{g}_k^T A\mathbf{p}_i = 0$ for $i = 1, \ldots, k-1$.

**Proof:** $\mathbf{g}_k^T A\mathbf{p}_i = -(A(\mathbf{x}_k - \mathbf{x}_\star))^T A\mathbf{p}_i$. Now $\mathbf{x}_k - \mathbf{x}_\star$ is $A$-conjugate to $\mathbf{p}_1, \ldots, \mathbf{p}_k$ and $[A\mathbf{p}_1, \ldots, A\mathbf{p}_{k-1}] \subseteq [\mathbf{p}_1, \ldots, \mathbf{p}_k]$ (Lemma 3). Thus $\mathbf{x}_k - \mathbf{x}_\star$ is $A$-conjugate to $A\mathbf{p}_1, \ldots, A\mathbf{p}_{k-1}$.

We thus have the following first version of the CG algorithm:

**Alg. 2:** Pick starting guess $\mathbf{x}_0$.

for $i = 1$ to $m$ {
  $\mathbf{g}_{i-1} = -(A\mathbf{x}_{i-1} + \mathbf{b})$; if $\mathbf{g}_{i-1} = 0$ return $\mathbf{x}_{i-1}$
  if $(i > 1)$ then $\beta_i = (\mathbf{g}_{i-1}^T A\mathbf{p}_{i-1})$ / $(\mathbf{p}_{i-1}^T A\mathbf{p}_{i-1})$
  if $i = 1$ then $\mathbf{p}_i = \mathbf{g}_0$ else $\mathbf{p}_i = \mathbf{g}_{i-1} - \beta_i\mathbf{p}_{i-1}$
  $\alpha_i = (\mathbf{g}_{i-1} \cdot \mathbf{p}_i)$ / $(\mathbf{p}_i^T A\mathbf{p}_i)$
  $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
return $\mathbf{x}_m$

The formulas in the algorithm can be simplified somewhat. For example the $\mathbf{g}_i$ can be computed recursively:

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

The expressions for $\beta_i$ and $\alpha_i$ can also be simplified. Multiplying the above gradient formula by $\mathbf{g}_i$ and $\mathbf{g}_{i-1}$, we see that

$$\begin{aligned} \|\mathbf{g}_i\|^2 &= -\alpha_i \mathbf{g}_i^T A\mathbf{p}_i \\ \|\mathbf{g}_{i-1}\|^2 &= \alpha_i \mathbf{g}_{i-1}^T A\mathbf{p}_i \end{aligned}$$

As $\mathbf{g}_{i-1} = \mathbf{p}_i + \beta_i \mathbf{p}_{i-1}$,

$$\begin{aligned} \|\mathbf{g}_{i-1}\|^2 &= \alpha_i \mathbf{g}_{i-1}^T A\mathbf{p}_i \\ &= \alpha_i \mathbf{p}_i^T A\mathbf{p}_i \end{aligned}$$

Therefore

$$\beta_{i+1} = -\frac{\|\mathbf{g}_i\|^2}{\|\mathbf{g}_{i-1}\|^2}$$

This gives the following simplified algorithm:

**Alg. 2:** Pick starting guess $\mathbf{x}_0$, and set $\mathbf{g}_0 = -(A\mathbf{x}_0 + \mathbf{b})$

for $i = 1$ to $m$ {
  if $\mathbf{g}_{i-1} = 0$ then return $\mathbf{x}_{i-1}$
  if $(i > 1)$ then $\beta_i = -\|\mathbf{g}_{i-1}\|^2 \,/\, \|\mathbf{g}_{i-2}\|^2$
  if $i = 1$ then $\mathbf{p}_i = \mathbf{g}_0$ else $\mathbf{p}_i = \mathbf{g}_{i-1} - \beta_i \mathbf{p}_{i-1}$
  $\alpha_i = \|\mathbf{g}_{i-1}\|^2 \,/\, (\mathbf{p}_i^T A\mathbf{p}_i)$
  $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \mathbf{p}_i$
  $\mathbf{g}_i = \mathbf{g}_{i-1} - \alpha_i A\mathbf{p}_i$
return $\mathbf{x}_m$

This algorithm requires one matrix $\times$ vector multiplication per iteration.

## 2. Conjugate Gradients for Least Squares Problems

We are given a response vector $\mathbf{y} \in R^n$, and a $n \times m$ design matrix $C$. We want to find a parameter vector $\mathbf{x}$ with

$$\|\mathbf{y} - C\mathbf{x}\|^2 = min!$$

This equivalent to solving the normal equations

$$C^T C\mathbf{x} - C^T\mathbf{y} = \mathbf{0}$$

or minimizing the quadratic function

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T C^T C\mathbf{x} - \mathbf{x}^T C^T\mathbf{y}.$$

We will use the CG method to minimize $F(\mathbf{x})$. Let $\mathbf{x}_k$ be the parameter vector after the $k$-th iteration. The residual vector after the $k$-th iteration is $\mathbf{r}_k = \mathbf{y} - C\mathbf{x}_k$, and the negative gradient is $\mathbf{g}_k = -\nabla F(\mathbf{x}_k) = C^T\mathbf{r}_k$.
Note that $\mathbf{p}_k^T C^T C\mathbf{p}_k = \|C\mathbf{p}_k\|^2$; so $\alpha_k$ can be obtained without calculating $C^T C$. Note also that $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{p}_k$, and therefore $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k C\mathbf{p}_k$.
This gives the following algorithm:

**Alg. 3:** Pick starting guess $\mathbf{x}_0$.

$\mathbf{r}_0 = \mathbf{y} - C\mathbf{x}_0$
$\mathbf{g}_0 = C^T\mathbf{r}_0$
for $i = 1$ to $m$ {
  if $\mathbf{g}_{i-1} = 0$ then return $\mathbf{x}_{i-1}$
  if $(i > 1)$ then $\beta_i = -\|\mathbf{g}_{i-1}\|^2 \;/\; \|\mathbf{g}_{i-2}\|^2$
  if $i = 1$ then $\mathbf{p}_i = \mathbf{g}_0$ else $\mathbf{p}_i = \mathbf{g}_{i-1} - \beta_i\mathbf{p}_{i-1}$
  $\alpha_i = \|\mathbf{g}_{i-1}\|^2 \;/\; \|C\mathbf{p}_i\|^2$
  $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
  $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i C\mathbf{p}_i$
  $\mathbf{g}_i = C^T\mathbf{r}_i$
return $\mathbf{x}_m$

This algorithm requires one multiplication of $C$ with an $m$-vector and one multiplication of $C^T$ with an $n$-vector.

### The CG Algorithm for Minimizing General Smooth Functions

Let $F(\mathbf{x}) : R^m \to R$ be a general smooth function (at least twice differentiable). A typical minimization procedure does a sequence of univariate minimization along suitably chosen directions:

**Alg 4:** Pick starting guess $\mathbf{x}_0$ and maximum number of iterations $maxit$.

for $i = 1$ to $maxit$ {
    Choose search direction $\mathbf{p}_i$
    $\alpha_i = \text{argmin}_t F(\mathbf{x}_{i-1} + t\mathbf{p}_i)$
    $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \mathbf{p}_i$
    if converged, return $\mathbf{x}_i$
    }
return "failed to converge"

An obvious choice for the search direction $\mathbf{p}_i$ is the negative gradient $-\nabla F(\mathbf{x}_{i-1})$. Alg 4 with this choice of directions is called *steepest descent* minimization. However, steepest descent is usually a terrible procedure. It quickly drops down into a valley and then makes very slow progress.

It is much better to use conjugate gradients. Close to a local minimum, every smooth function is approximately quadratic with positive definite Hessian. Once it has gotten close enough, CG will home in on the solution very rapidly.

At first glance it seems that in order to apply CG, we need to have the Hessian of $F$, because it would take the role of $A$. This can be avoided. Inspection of Alg 2 reveals that $A$ is used in two places, namely in the calculation of $\nabla F(\mathbf{x}_i)$, and the calculation of the step size $\alpha_i$. If we can provide a procedure to calculate $\nabla F$, the first use can be avoided. We can find the step size $\alpha_i$ by performing a univariate numerical minimization of $F(\mathbf{x}_i + t\mathbf{p}_i)$. The CG method with these two modifications is called the Fletcher-Reeves method.