Conjugate Gradient Tutorial

Prof. Chung-Kuan Cheng

Computer Science and Engineering Department
University of California, San Diego

ckcheng@ucsd.edu

December 1, 2015

Overview

Introduction: Overview

Conjugate Gradient is an extension of steepest gradient descent. For steepest gradient, we step in one direction per iteration. Through the iterations, we found that the new directions may contain the component of the old directions and the process walks in zig-zag patterns. For conjugate gradient, we consider multiple directions simulteneously. Hence, we avoid to repeat the old directions. In 1952, Hestenes and Stiefel independently introduced conjugate gradient formula to simplify the multiple direction search.

Introduction: Overview

- Steepest Gradient Descent: We derive the method and properties of the steepest descent method. We view the steepest descent method as an one-direction per iteration approach. The method suffers slow zig-zag winding in a narrow valley of equal potential terrain.
- Preconditioning: From the properties of the steepest descent method, we find that preconditioning improves the convergence rate.
- Conjugate Gradient in Global View: We view conjugate gradient method from the aspect of gradient descent. However, the descent method considers multiple directions simultaneously.
- Conjugate Gradient Formula: We state the formula of conjugate gradient.
- Conjugate Gradient Method Properties: We show that the global view of conjugate gradient method can be used to optimize each step independent of the other steps. Therefore, the process can repeat recursively and converge after n iterations, where n is the number of variables. Finally, we show and prove the property that validates the formula.

Introduction: Formulation

The original problem is to solve a simultaenous linear equation, $Ax = b$, where matrix A is symmetric and positive definite. Calculating the inverse $x = A^{-1}b$ can be complicated, e.g. n is huge. To avoid a direct solver, we formulate the problem with a quadratic convex objective function.

- Formulation

$$\text{minimize} \quad \frac{1}{2}x^T A x - b^T x, \quad A \in S^n_{++}$$

- Solution: $x = A^{-1}b$.
- To avoid direct solvers, use Gradient Descent iteratively to find the answer.

Steepest Descent Formula

Given initial $k = 0, x_k = x_0$. We descent one direction per iteration along the gradient of the objective function.

- Derive residual $r_k = -\nabla f(x_k) = b - Ax_k$
- Set $x_{k+1} = x_k + \alpha_k r_k$, where step size α_k is derived analytically.
- Step size $\alpha_k = \arg\min_{s \geq 0} f(x_k + sr_k)$,
  From $\frac{\partial f(x_k + \alpha r_k)}{\partial \alpha_k} = 0$, we have $\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}$
- Therefore, we have $x_{k+1} = x_k + \frac{r_k^T r_k}{r_k^T A r_k} r_k$
- Repeat the above steps with $k = k + 1$ until the norm of r_k is within tolerance.

Steepest Descent Properties

- Formula: $x_{k+1} = x_k + \alpha_k r_k = x_k + \frac{r_k^T r_k}{r_k^T A r_k} r_k$
- Objective function: $f(x_k) - f(x_k + \alpha_k r_k) = \frac{(r_k^T r_k)^2}{2 r_k^T A r_k}$
- Residual $r_{k+1} = (I - \alpha_k A) r_k = (I - \frac{(r_k^T r_k)^2}{r_k^T A r_k} A) r_k$

  Proof:

$$\begin{aligned} r_{k+1} &= b - A x_{k+1} = b - A(x_k + \alpha_k r_k) \\ &= r_k - \alpha_k A r_k = (I - \alpha_k A) r_k \end{aligned}$$

- Property of the next direction: $r_{k+1} \perp r_k$

  Proof: $r_k^T r_{k+1} = r_k^T (I - \frac{(r_k^T r_k)^2}{r_k^T A r_k} A) r_k = 0.$

Steepest Descent: Convergence

- We denote $x = x^* + e$, where x^* is the optimal solution and e is the error that we try to reduce.
- We try to decrease the residual so that e can be reduced.

$$r_k = b - Ax_k = b - Ax^* - Ae_k = -Ae_k$$

As $r \to 0, \ e \to 0.$

Gradient Descent: Preconditioning

We want to reduce the residual $r_k = -Ae_k$.

- Let $e_k = \sum_{i=1}^{n} \xi_i v_i$, where v_i are the eigenvectors of A, $\forall i = 1, 2, \ldots, n$.
- Then, we have $r_k = -Ae_k = -\sum_{i=1}^{n} \lambda_i \xi_i v_i$, where λ_i are the eigenvalues of A.
- Thus, the next residual becomes

$$\begin{aligned} r_{k+1} &= \left(I - \frac{r_k^T r_k}{r_k^T A r_k} A \right) r_k \\ &= -\sum_{i=1}^{n} \lambda_i \xi_i v_i + \frac{\sum_{i=1}^{n} \lambda_i^2 \xi_i^2}{\sum_{i=1}^{n} \lambda_i^3 \xi_i^2} \sum_{i=1}^{n} \lambda_i^2 \xi_i v_i. \end{aligned}$$

Suppose that all eigenvalues are equal, i.e. $\lambda_i = \lambda, \ \forall i$. We have

$$r_{k+1} = -\lambda \sum_{i=1}^{n} \xi_i v_i + \frac{\lambda^2 \sum_{i=1}^{n} \xi_i^2}{\lambda^3 \sum_{i=1}^{n} \xi_i^2} \lambda^2 \sum_{i=1}^{n} \xi_i v_i = 0$$

Gradient Descent: Preconditioning

- $\nabla f(x) = Ax - b = 0 \Rightarrow Ax = b$
- Preconditioning: To transform $Ax = b$ into another system with more favorable properties for it to be iteratively solved
- With the preconditioner M, $M^{-1}Ax = M^{-1}b$ (e.g. incomplete LU scaling)

Conjugate Gradient: Descent with Multiple Vectors

For conjugate gradient, we consider multiple vectors $V_k = [v_0, v_1, \ldots, v_k]$ in stage k.

- Let $x_{k+1} = x_k + V_k y$, where $y = [y_1, y_2, \ldots, y_k]^T$ is a vector of parameters. We can write $V_k y = \sum_{i=1}^{k} y_i v_i$.
- To minimize $f(x_{k+1})$, the solution is $y = (V_k^T A V_k)^{-1} V_k^T r_k$. Therefore, $x_{k+1} = x_k + V_k y = x_k + V_k (V_k^T A V_k)^{-1} V_k^T r_k$.

Proof: To minimize $f(x_{k+1})$, we want $\nabla_y f(x_{k+1}) = 0$.
We have

$$
\begin{aligned}
\nabla_y f(x_{k+1}) &= \nabla_y \left\{ \frac{1}{2}(x_k + V_k y)^T A (x_k + V_k y) - b^T (x_k + V_k y) \right\} \\
&= V_k^T A V_k y + V_k^T A x_k - V_k^T b = V_k^T A V_k y - V_k^T r_k = 0 \\
\Rightarrow y &= (V_k^T A V_k)^{-1} V_k^T r_k.
\end{aligned}
$$

Conjugate Gradient: Multiple Vector Optimization

For the descent on multiple directions, we have the following properties.

- Function: Since $y = (V_k^T A V_k)^{-1} V_k^T r_k$, we have

$$f(x_{k+1}) = f(x_k) + \frac{1}{2} y^T V_k^T A V_k y + y^T V_k^T (Ax - b)$$
$$= f(x_k) - \frac{1}{2} r_k^T V_k (V_k^T A V_k)^{-1} V_k^T r_k.$$

- Residual:

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + V_k (V_k^T A V_k)^{-1} V_k^T r_k)$$
$$= (I - AV_k (V_k^T A V_k)^{-1} V_k^T) r_k.$$

- Property A: $r_{k+1} \perp V_k$. The proof is independent of the choice of V_k.

$$\text{Proof:} V_k^T r_{k+1} = V_k^T (I - AV_k (V_k^T A V_k)^{-1} V_k^T) r_k$$
$$= (V_k^T - V_k^T) r_k = 0$$

Global Procedure in Matrix Form V_k

Through iterations, we want to increase the size of matrix $V_k = [v_0, v_1, \ldots, v_k]$ to V_{k+1} by adding a new vector v_{k+1} at the last column for iteration $k+1$.

Initial $k = 0, v_0 = r_0 = b - Ax_0$.

Repeat:

- Update $x_{k+1} = x_k + V_k(V_k^T A V_k)^{-1} V_k^T r_k$ and $r_{k+1} = b - x_{k+1}$.
- Exit if the norm of $r_{k+1} <$ tolerance.
- Derive v_{k+1} as a function of r_{k+1} and V_k (to be described in CG formula).
- Construct V_{k+1} by appending v_{k+1} to the last column of V_k.
- $k = k + 1$.

Property B (independent of the choice of v_k): According to the procedure, we have $V_k^T r_k = [0, \ldots, 0, v_k^T r_k]^T$.

Proof: From Property A, we have $V_{k-1}^T r_k = 0$, thus $V_k^T r_k = [0, \ldots, 0, v_k^T r_k]^T$.

Conjugate Gradient: Wish List

We hope that $V^T AV = D = diag\, d_i$ is a diagonal matrix. In this case, we call that the vectors v_i in V are mutually conjugate with respect to matrix A.

- If $V^T AV = D = diag\, d_i$, we have $d_i = v_i^T Av_i$
- Therefore, we have $x_{k+1} = x_k + V_k(V_k^T AV_k)^{-1}V_k^T r_k = x_k + V_k D^{-1}[0, \ldots, 0, v_k^T r_k]^T = x_k + \alpha_k v_k$ (Property B), where $\alpha_k = \frac{v_k^T r_k}{v_k^T Av_k}$
- Hopefully, for the new matrix V_{k+1}, the conjugate property remains to be true. Then, we can repeat the steps by increasing $k = k + 1$.
- When $k = n - 1$, we have $r_n^T V_{n-1} = 0$ (property A). The last residual $r_n = 0$, since matrix V_{n-1} is full ranked. Thus, we have the solution $x_n = x^*$.

Conjugate Gradient Descent Formula

Given x_0, we set initial: $k = 0, v_k = r_k = b - Ax_0$.

- $x_{k+1} = x_k + \alpha_k v_k$, where $\alpha_k = \frac{v_k^T r_k}{v_k^T A v_k} \quad (= \frac{r_k^T r_k}{v_k^T A v_k})$.
- $r_{k+1} = b - Ax_{k+1} = b - Ax_k - \alpha_k A v_k = r_k - \alpha_k A v_k$.
- $v_{k+1} = r_{k+1} + \beta_{k+1} v_k$, where $\beta_{k+1} = \frac{1}{\alpha_k} \frac{r_{k+1}^T r_{k+1}}{v_k^T A v_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$.
- Repeat the iteration with $k = k + 1$ until the residual is smaller than the tolerance.

Lemma: $v_k^T r_k = r_k^T r_k$.
Proof: From Property A, we have $v_k^T r_k = (r_k + \beta_k v_{k-1})^T r_k = r_k^T r_k$.

Validation of the Properties

Theorem: The solution x_{k+1} of the conjugate gradient formula is consistent with the global procedure, i.e. vectors v_i produced by the formula are mutually conjugate. The consistence is based on the following three equalities.

- Property A: $r_i^T v_j = 0, \forall i > j$.
- Residuals: $r_i^T r_j = 0, \forall i > j$.
- Conjugates: $v_i^T A v_j = 0, \forall i > j$.

Proof: We prove the three equalities by induction. For the case when index $i = 1$, we have

- Property A: $r_1^T v_0 = 0$
- Residuals: $r_1^T r_0 = 0 \quad (r_0 = v_0)$
- Conjugates:

$$
\begin{aligned}
v_1^T A v_0 &= (r_1 + \beta_1 v_0)^T A v_0 = r_1^T A v_0 + \beta_1 v_0^T A v_0 \\
&= r_1^T \left(\frac{r_0 - r_1}{\alpha_0}\right) + \frac{1}{\alpha_0} \frac{r_1^T r_1}{v_0^T A v_0} v_0^T A v_0 = 0 \quad (r_1^T v_0 = 0, r_0 = v_0)
\end{aligned}
$$

Validation of the Wish List

Proof by induction (continue): Suppose that the statement is true up to index $i = k$. By assumption of the three equalities, the conjugate gradient formula is consistent with the global procedure up to $x_{k+1} = x_k + \alpha_k v_k$. When index is $i = k + 1$, we have

- Property A: $r_{k+1}^T V_k = 0$
- Residuals: $r_{k+1}^T r_j = r_{k+1}^T (v_j - \beta_j v_{j-1}) = 0, \forall j < k$
- Conjugates:

$$
\begin{aligned}
\text{Case } j = k{:}\, v_{k+1}^T A v_k &= (r_{k+1} + \beta_{k+1} v_k)^T A v_k = r_{k+1}^T A v_k + \beta_{k+1} v_k^T A v_k \\
&= r_{k+1}^T \left(\frac{r_k - r_{k+1}}{\alpha_k}\right) + \frac{1}{\alpha_k} \frac{r_{k+1}^T r_{k+1}}{v_k^T A v_k} v_k^T A v_k \\
&= 0 \quad (r_{k+1}^T r_k = 0).
\end{aligned}
$$

$$
\begin{aligned}
\text{Case } j < k{:}\, v_{k+1}^T A v_j &= (r_{k+1} + \beta_{k+1} v_k)^T A v_j = r_{k+1}^T A v_j \\
&= r_{k+1}^T \left(\frac{r_j - r_{j+1}}{\alpha_j}\right) = 0, \forall j < k.
\end{aligned}
$$

Summary

We view the conjugate gradient method as an extension from one-direction descent of steepest gradient method to multiple-direction descent. From the global procedure of the multiple vector search, we can derive the basic properties of the optimization. The optimization result shows that the inversion of $V^T AV$ is one main cause of the zig-zag winding of the steepest descent approach. The formula of conjugate gradient method transforms the product $V^T AV$ into a diagonal matrix and thus simplifies the optimization procedure. Consequently, we can achive the desired properties and the convergence of the solution.

Acknowledgement: The note is scribed by YT Jerry Peng for class CSE291, Fall 2015.

References

- J.R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," CMU Technical Report, 1994.
- Convex optimization, by S. Boyd and L. Vandenberghe, Cambridge University Press, 2004.
- Matrix computations, G.H. Golub and C.F. Van Loan, Johns Hopkins, 2013.
- Numerical Recipes: The Art of Scientific Computing, by W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Cambridge University Press, 2007.