Conjugate Gradient Algorithm

Mo Li

December 9, 2018

The values of parameter $\lambda_0 = (\lambda_1, \lambda_2, ..., \lambda_n)$ can be obtained by solving the following linear equation:

$$\Gamma_0 \lambda_0 = \gamma_0 \tag{1}$$

where Γ_0 is a symmetric $n \times n$ matrix,

$$[\Gamma_0]_{ij} = \gamma(x_i - x_j),$$

and

$$\gamma_0 = (\gamma(x_0 - x_1), ..., \gamma(x_0 - x_n)).$$

Because the parameter λ_0 is the solution of the linear system (eq. 1), we have $\lambda_0 = \gamma_0^{-1}\gamma_0$. **However, for large data sets, obtaining the inverse of Γ_0 is generally difficult.** We propose here to minimize the function $\phi(x)$, defined by

$$\phi(x) = \frac{1}{2} X^T \Gamma_0 X - X^T \gamma_0 \tag{2}$$

The minimum value of ϕ is $-\gamma_0^T \Gamma_0^{-1} \gamma_0 / 2$, obtained by setting $x = \Gamma_0^{-1}\gamma_0$. Thus, minimizing ϕ and solving (eq. 2) are equivalent problems. The *conjugate gradient algorithm* is an iterative method used to minimize (eq. 2). If Γ_0 is a $n \times n$ matrix the conjugate gradient algorithm will converge in n iterations to the solution. This method is considers the successive minimization of ϕ along a set of directions $\{p_1, p_2, ...\}$ If x_{k-1} is the current approximation to the solution in (eq. 2) for a direction p_k, then $x_k = x_{k-1} + \alpha p_k$ will be the new conjugate gradient iterate, where we choose α to minimize (eq. 2). It is easy to show that to minimize $\phi(x_{k-1} + \alpha p_k)$ with respect to α, we merely set

$$\alpha = \alpha_k = p_k^T(\gamma_0 - \Gamma_0 x_{k-1}) / p_k^T \Gamma_0 p_k.$$

The convergence rate of the algorithm will depend on the starting vector.