Non-Negative Quadratic Pursuit

Author Name

Affiliation, Country.

A pre-print of the NQP algorithm related to a recently submitted conference paper, as provided by the authors. Author names will be updated after the reviewing process is finished.

Abstract

We propose the Non-negative Quadratic Pursuit (NQP) algorithm to approximately minimize a quadratic function in the presence of the $l0$-norm constraint. In this document, we explain the algorithm's exact steps along with its convergence proof and complexity analysis.

1 None-negative Quadratic Pursuit Algorithm

Consider a quadratic function $f(\vec{\gamma}) := \vec{\gamma}^\top \mathbf{Q}\vec{\gamma} + \vec{c}^\top\vec{\gamma}$, in which $\vec{\gamma} \in \mathbb{R}^n$, $\vec{c} \in \mathbb{R}^n$, and $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a hermitian positive semidefinite matrix. NQP algorithm is an extended form of the Matching Pursuit problem [1], and its objective is to approximately minimize $f(\vec{\gamma})$ in the NP-hard optimization problems similar to Eq. (1).

$$\begin{aligned} \vec{\gamma} = \quad & \arg\min_{\vec{\gamma}} \tfrac{1}{2}\vec{\gamma}^\top \mathbf{Q}\vec{\gamma} + \vec{c}^\top\vec{\gamma} \\ & s.t. \quad \|\vec{\gamma}\|_0 \leq m \ , \ \gamma_i \geq 0, \forall i \end{aligned} \tag{1}$$

where at most $m \ll n$ elements from $\vec{\gamma}$ are permitted to be positive while all other elements are forced to be zero.

As presented in Algorithm 1, at each iteration of NQP we compute $\nabla_{\vec{\gamma}} f(\vec{\gamma})$ to guess about the next promising dimension of $\vec{\gamma}$ (denoted as γ_j) which may lead to the biggest decrease in the current value of $f(\vec{\gamma}_{\mathcal{I}})$; where $\mathcal{I}$ denotes the set of currently chosen dimensions of $\vec{\gamma}$ based on the previous iterations. We look for $\vec{\gamma} \geq 0$ solutions, and also the current value of $\vec{\gamma}$ entries for new dimensions are zero; therefore, similar to the Gauss-Southwell rule in coordinate descent optimization [2] we choose the dimension j related to the smallest negative entry of $\nabla_{\vec{\gamma}} f(\vec{\gamma})$

$$j = \arg\min_{j\in S} \ \vec{q}_j^\top \vec{\gamma} + c_j \qquad s.t. \ \vec{q}_j^\top \vec{\gamma} + c_j < 0 \tag{2}$$

where $\vec{q}_j$ is the j-th column of $\mathbf{Q}$. Then by adding j to $\mathcal{I}$, the resulting unconstrained quadratic problem will be solved using the closed form solution

$\vec{\gamma}_{\mathcal{I}} = -\mathbf{Q}_{\mathcal{I}\mathcal{I}}{}^{-1}\vec{c}_{\mathcal{I}}$, and generally we repeat this process until reaching $\|\vec{\gamma}\|_0 = m$ criterion. $\mathbf{Q}$ and $\vec{c}_{\mathcal{I}}$ denote the principal submatrix of $\mathbf{Q}$ and the subvector of $\vec{c}$ corresponding to the set $\mathcal{I}$.

In order to preserve non-negativity of the solution $\vec{\gamma}$ in each iteration t of NQP, in case of having a negative entry in $\vec{\gamma}^t_{\mathcal{I}}$, a simple line search is performed between $\vec{\gamma}^t_{\mathcal{I}}$ and $\vec{\gamma}^{(t-1)}_{\mathcal{I}}$. The line search chooses the nearest zero-crossing point to $\vec{\gamma}^{(t-1)}_{\mathcal{I}}$ on the connecting line between $\vec{\gamma}^{(t-1)}_{\mathcal{I}}$ and $\vec{\gamma}^t_{\mathcal{I}}$.

In addition, to reduce the computational cost, we use the Cholesky factorization [3] $\mathbf{Q}_{\mathcal{I}\mathcal{I}} = \mathbf{L}\mathbf{L}^\top$ to compute $\vec{\gamma}$ with a back-substitution process.

Furthermore, because matrix $\mathbf{Q}$ in equations (1) is PSD, theoretically its principal sub-matrix $\mathbf{Q}_{\mathcal{I}\mathcal{I}}$ should be either PD or PSD [4], where the first case is a requirement the Cholesky factorization. However, in practice by choosing $m << rank(Q)$ we have never confronted a singular condition. Nevertheless, to avoid such rare conditions, we do a non-singularity check for the selected dimension j which is to have $q_{jj} \neq v^\top v$ right after obtaining v. In case the resulted v does not fulfill that condition, we choose another j based on Eq. (2)

1.1 The Convergence of NQP

NQP does not guarantee the global optimum as it is a greedy selection of rows/columns of matrix $\mathbf{Q}$ to provide a sparse approximation of Eq. (1) under the hard-sparsity constraint; Nevertheless its convergence to an optimum point is guaranteed. The algorithm consists of 3 main parts:

1. The gradient based dimension selection
2. Closed form solution
3. Non-negative line search and updating $\mathcal{I}$.

It is clear that the closed-form solution $\vec{\gamma}$ via selecting a negative direction of the gradient $\nabla_{\vec{\gamma}} f(\vec{\gamma})$ always reduces the current value of $f(\vec{\gamma}^t)$ as $\vec{\gamma}^t$ has to be non-negative and initially $\gamma_j = 0$. In addition, The zero-crossing line search in iteration t can guarantee to strictly reduce the value of $f(\vec{\gamma}^{(t-1)})$. It finds a non-negative $\vec{\gamma}^t_{new}$ between the line connecting $\vec{\gamma}^{(t-1)}_{\mathcal{I}}$ to $\vec{\gamma}^t_{\mathcal{I}}$, and since $f(\vec{\gamma})$ is convex, $f(\vec{\gamma}^t_{new}) < f(\vec{\gamma}^{(t-1)}_{\mathcal{I}})$

Consequently, each of the steps guarantees a monotonic decrease in the value of $f(\vec{\gamma})$, therefore if $\|\vec{\gamma}^{(t+i)}\|_0 > \|\vec{\gamma}^{(t)}\|_0 \implies f(\vec{\gamma}^{(t+i)}) < f(\vec{\gamma}^{(t)})$. Also the algorithm structure guarantees that in any iteration t, $\mathcal{I}_t \neq \mathcal{I}_i \ \forall i < t$ meaning that NQP never gets trapped into a loop of repeated dimension selections. Furthermore we have $\|\vec{\gamma}\|_0 < m$, meaning that the total number of possible selections in $\mathcal{I}$ is bounded. Concluding from the aboves, the NQP algorithm converges in a limited number of iterations.

Algorithm 1: Non-negative Quadratic Pursuit

Parameters: m: cardinality threshold, ϵ : stopping threshold
Input: $\mathbf{Q} \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$ when $f(\vec{\gamma}) = \frac{1}{2}\vec{\gamma}^\top \mathbf{Q}\vec{\gamma} + \vec{c}^\top \vec{\gamma}$
Output: An approximate solution $\vec{\gamma}$
Initialization: $\vec{\gamma} = 0$, $\mathcal{I} = \{\}$, $\mathcal{S} = \{1, ..., n\}$, $t = 1$
repeat
  $j = \arg\min_{j \in S} \vec{q}_j^\top \vec{\gamma} + c_j \qquad s.t. \;\; \vec{q}_j^\top \vec{\gamma} + c_j < 0$
  if $j = \varnothing$ **then** Convergence.
  $\mathcal{I} := \mathcal{I} \cap j$;
  $\vec{q}_{\mathcal{I}j}$:= a vector based on selecting rows $\mathcal{I}$ and column j of matrix $\mathbf{Q}$.
  $\vec{c}_{\mathcal{I}}$:= a subvector of c based on selecting entries $\mathcal{I}$ of vector $\vec{c}$.
  if $t > 1$ **then**
    $v :=$ Solve for v $\left\{\mathbf{L}v = \vec{q}_{\mathcal{I}j}\right\}$;
    $\mathbf{L} := \begin{bmatrix} \mathbf{L} & 0 \\ v^\top & \sqrt{q_{jj} - v^\top v} \end{bmatrix}$
  else
    $\mathbf{L} = q_{jj}$
  end
  $\vec{\gamma}_{\mathcal{I}}^t :=$ Solve for x $\left\{\mathbf{L}\mathbf{L}^\top x = \vec{c}_{\mathcal{I}}\right\}$;
  if $\exists j \in \mathbb{N};\ (\gamma_j^t < 0)$ **then**
    $\vec{\gamma}_{\mathcal{I}}^t :=$ the nearest zero-crossing to $\vec{\gamma}_{\mathcal{I}}^{(t-1)}$ via a line search.
    $\mathcal{S} := \mathcal{S} - \{$zeros entries in $\vec{\gamma}_{\mathcal{I}}^t\}$
  end
  $\mathcal{S} := \mathcal{S} - j$
  $t = t + 1$
until $(\mathcal{S} = \{\}) \ \vee \ (\|\vec{\gamma}\|_0 = m) \ \vee \ (\frac{1}{2}\vec{\gamma}^\top Q\vec{\gamma} + c^\top \vec{\gamma} < \epsilon)$;
Convergence.

1.2 The Computational Complexity of NQP

We can calculate computational complexity of NQP by considering its individual steps. Iteration t contains computing $\mathbf{Q}\vec{\gamma} + \vec{c}$ ($nt + t$ operation), finding minimum of $\nabla_{\vec{\gamma}} f(\vec{\gamma})$ w.r.t the negative constraint ($2n$ operations), computing v (t^2 operation for the $t \times t$ back-substitution), computing $\vec{\gamma}_{\mathcal{I}}^t$ (two back-substitutions resulting in $2t^2$ operation), and checking negativity of entries of $\vec{\gamma}_{\mathcal{I}}^t$ along with the probable line-search which has $3t$ operations in total. Computing for all m iterations, the total runtime of NQP is obtained as

$$\mathbf{T}_{NQP} = (n+4)m^2 + 2mn + m^3$$

Considering that in practice $m \ll n$, the algorithm's computational complexity is $\mathcal{O}(nm^2)$.

References

[1] M. B. A. Aharon, M.and Elad, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[2] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[3] C. F. Van Loan, "Matrix computations (johns hopkins studies in mathematical sciences)," 1996.

[4] C. R. Johnson and H. A. Robinson, "Eigenvalue inequalities for principal submatrices," *Linear Algebra and its Applications*, vol. 37, pp. 11–22, 1981.