Non-Negative Quadratic Pursuit

Babak Hosseini
Bielefeld University, Germany

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 \( f(\bar{\gamma}) := \bar{\gamma}^T Q \bar{\gamma} + \bar{c}^T \bar{\gamma} \), in which \( \bar{\gamma} \in \mathbb{R}^n \), \( \bar{c} \in \mathbb{R}^n \), and \( Q \in \mathbb{R}^{n \times n} \) 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 \( f(\bar{\gamma}) \) in the NP-hard optimization problems similar to Eq. (1).

\[
\bar{\gamma} = \arg \min_{\bar{\gamma}} \frac{1}{2} \bar{\gamma}^T Q \bar{\gamma} + \bar{c}^T \bar{\gamma} \\
\text{s.t. } \|\bar{\gamma}\|_0 \leq m , \gamma_i \geq 0, \forall i
\]  

where at most \( m \ll n \) elements from \( \bar{\gamma} \) 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 \( \nabla_{\bar{\gamma}} f(\bar{\gamma}) \) to guess about the next promising dimension of \( \bar{\gamma} \) (denoted as \( \gamma_j \)) which may lead to the biggest decrease in the current value of \( f(\bar{\gamma}_{\mathcal{I}}) \); where \( \mathcal{I} \) denotes the set of currently chosen dimensions of \( \bar{\gamma} \) based on the previous iterations. We look for \( \bar{\gamma} \geq 0 \) solutions, and also the current value of \( \bar{\gamma} \) 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 \( \nabla_{\bar{\gamma}} f(\bar{\gamma}) \)

\[
j = \arg \min_{j \in \mathcal{S}} q_j^T \bar{\gamma} + c_j \text{ s.t. } q_j^T \bar{\gamma} + c_j < 0
\]  

where \( q_j \) is the \( j \)-th column of \( Q \). Then by adding \( j \) to \( \mathcal{I} \), the resulting unconstrained quadratic problem will be solved using the closed form solution \( \bar{\gamma}_{\mathcal{I}} = -Q_{\mathcal{I} \mathcal{I}}^{-1} \bar{c}_{\mathcal{I}} \), and generally we repeat this process until reaching \( \|\bar{\gamma}\|_0 = m \) criterion. \( Q \) and \( \bar{c}_{\mathcal{I}} \) denote the principal submatrix of \( Q \) and the subvector of \( \bar{c} \) 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 $\vec{\gamma}_t^{(t-1)}$, a simple line search is performed between $\vec{\gamma}_I^{(t-1)}$ and $\vec{\gamma}_I^t$. The line search chooses the nearest zero-crossing point to $\vec{\gamma}_I^{(t-1)}$ on the connecting line between $\vec{\gamma}_I^{(t-1)}$ and $\vec{\gamma}_I^t$.

In addition, to reduce the computational cost, we use the Cholesky factorization $Q_{II} = LL^\top$ to compute $\vec{\gamma}$ with a back-substitution process.

Furthermore, because matrix $Q$ in equations (1) is PSD, theoretically its principal sub-matrix $Q_{II}$ should be either PD or PSD [4], where the first case is a requirement the Cholesky factorization. However, in practice by choosing $m \ll \text{rank}(Q)$ 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).

**Algorithm 1: Non-negative Quadratic Pursuit**

**Parameters:** $m$: cardinality threshold, $\epsilon$: stopping threshold

**Input:** $Q \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$ when $f(\vec{\gamma}) = \frac{1}{2} \vec{\gamma}^\top Q \vec{\gamma} + c^\top \vec{\gamma}$

**Output:** An approximate solution $\vec{\gamma}$

**Initialization:** $\vec{\gamma} = 0$, $I = \{}$, $S = \{1, ..., n\}$, $t = 1$

repeat
  $j = \text{arg min}_{j \in S} q_{jj}^T \vec{\gamma} + c_j \quad \text{s.t.} \quad q_{jj}^T \vec{\gamma} + c_j < 0$

  if $j = \emptyset$ then Convergence.

  $I := I \cap j$;
  $\vec{q}_{II} := \text{a vector based on selecting rows } I \text{ and column } j \text{ of matrix } Q$.
  $\vec{c}_I := \text{a subvector of } c \text{ based on selecting entries } I \text{ of vector } \vec{c}$.

  if $t > 1$ then
    $v := \text{Solve for } v \{Lv = \vec{q}_{II}\}$;
    $L := \begin{bmatrix} L & 0 \\ v^\top & \sqrt{q_{jj} - v^\top v} \end{bmatrix}$
  else
    $L = q_{jj}$
  end

  $\vec{\gamma}_I^t := \text{Solve for } x \{LL^\top x = \vec{c}_I\}$;

  if $\exists j \in \mathbb{N}; (\gamma_j^t < 0)$ then
    $\vec{\gamma}_I^t := \text{the nearest zero-crossing to } \vec{\gamma}_I^{(t-1)} \text{ via a line search.}$
    $S := S - \{\text{zeros entries in } \vec{\gamma}_I^t\}$
  end

  $S := S - j$
  $t = t + 1$
until ($S = \{}$) $\lor$ ($||\vec{\gamma}||_0 = m$) $\lor$ ($\frac{1}{2} \vec{\gamma}^T Q \vec{\gamma} + c^T \vec{\gamma} < \epsilon$);
Convergence.
1.1 The Convergence of NQP

NQP does not guarantee the global optimum as it is a greedy selection of rows/columns of matrix \(Q\) 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 \(I\).

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 \(f(\vec{\gamma}^{(t)}(t-1))\). It finds a non-negative \(\vec{\gamma}^{(t)}_{\text{new}}\) between the line connecting \(\vec{\gamma}^{(t-1)}(t)\) to \(\vec{\gamma}^t(t)\), and since \(f(\vec{\gamma})\) is convex, \(f(\vec{\gamma}^{(t)}_{\text{new}}) < f(\vec{\gamma}^{(t-1)}(t))\)

Consequently, each of the steps guarantees a monotonic decrease in the value of \(f(\vec{\gamma})\), therefore if \(\|\vec{\gamma}^{(t+1)}\|_0 > \|\vec{\gamma}^t\|_0 \implies f(\vec{\gamma}^{(t+1)}) < f(\vec{\gamma}^{(t)})\). Also the algorithm structure guarantees that in any iteration \(t, I_t \neq I_v \forall i < t\) 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 \(I\) is bounded. Concluding from the aboves, the NQP algorithm converges in a limited number of iterations.

1.2 The Computational Complexity of NQP

We can calculate computational complexity of NQP by considering its individual steps. Iteration \(t\) contains computing \(Q\vec{\gamma} + \vec{c}\) \((nt + t\) 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 \(\vec{\gamma}^t_l\) \((2t\) back-substitutions resulting in \(2t^2\) operation), and checking negativity of entries of \(\vec{\gamma}^t_l\) 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

\[T_{NQP} = (n + 4)m^2 + 2mn + m^3\]

Considering that in practice \(m \ll n\), the algorithm’s computational complexity is \(O(nm^2)\).

References

