Applied Mathematics and Computation 235 (2014) 80–86

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

An incremental decomposition method for unconstrained optimization

Luca Bravi *, Marco Sciandrone

Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Via Santa Marta 3, 50139 Florence, Italy

ARTICLE INFO

Keywords:
Large-scale unconstrained optimization
Decomposition
Gradient incremental methods

ABSTRACT

In this work we consider the problem of minimizing a sum of continuously differentiable functions. The vector of variables is partitioned into two blocks, and we assume that the objective function is convex with respect to a block-component. Problems with this structure arise, for instance, in machine learning. In order to advantageously exploit the structure of the objective function and to take into account that the number of terms of the objective function may be huge, we propose a decomposition algorithm combined with a gradient incremental strategy. Global convergence of the proposed algorithm is proved. The results of computational experiments performed on large-scale real problems show the effectiveness of the proposed approach with respect to existing algorithms.

1. Introduction

Let us consider the problem

$$\min_{x\in R^{n_x}, y\in R^{n_y}} f(x,y) = \sum_{i=1}^{m} f_i(x,y), \tag{1}$$

where $f : R^n \to R$ is a bounded below, continuously differentiable function, $n = n_x + n_y$, and $f(\cdot, y) : R^{n_x} \to R$ is a convex function for any $y \in R^{n_y}$. Furthermore, we assume that the number m of terms composing the objective function may be huge, so that the computation of the whole objective function and the whole gradient may be very expensive.

Problem (1) arises, for instance, in machine learning (see, e.g., [6,9]), and takes the form

$$\min_{x\in R^{n_x}, y\in R^{n_y}} f(x,y) = \frac{1}{2}\|A(y)x - b\|^2 + \frac{1}{2}\tau\left(\|x\|^2 + \|y\|^2\right), \tag{2}$$

where A is a $m \times n_x$ matrix depending on the subvector $y \in R^{n_y}$ of variables, $b \in R^m$, $\tau > 0$ is a regularizing parameter. Following the approach proposed in [5] for handling missing data, a formulation of the form (2) could be adopted even in connection with linear regression models with uncertainty in input data.

We may observe that, once fixed y, the resulting subproblem in the block component x is a linear least square problem, which can be efficiently solved even if the number m is very large (possibly by means of an incremental-based strategy). Therefore, the adoption of a decomposition technique for solving problem (2) appears suitable.

* Corresponding author. Tel.: +39 055 4796460; fax: +39 055 4796363.
E-mail addresses: l.bravi@unifi.it (L. Bravi), marco.sciandrone@unifi.it (M. Sciandrone).

http://dx.doi.org/10.1016/j.amc.2014.02.088
0096-3003/

A nonlinear Gauss–Seidel decomposition algorithm [2] applied to (1) leads to an iteration of the form

$$x^{k+1} \in \arg\min_{x \in R^{n_x}} f(x, y^k),$$
$$y^{k+1} \in \arg\min_{y \in R^{n_y}} f(x^{k+1}, y).$$

Global convergence of the Gauss–Seidel algorithm applied to (1) follows from the results stated in [10]. However, the updating of the block-component y requires to solve a nonconvex global optimization problem, and this may be not trivial.

In order to take into account this latter issue, a globally convergent modified version of the Gauss–Seidel algorithm proposed in [10] could be adopted defining the following iteration

$$\begin{aligned} x^{k+1} &\in \arg\min_{x \in R^{n_x}} f(x, y^k),\\ y^{k+1} &= y^k - \alpha^k \nabla_y f(x^{k+1}, y^k), \end{aligned} \tag{3}$$

where the stepsize α^k is computed by means of an Armijo-type line search. This strategy requires the computation of the whole objective function and of the gradient, and this may be not convenient since we are assuming that the number of terms composing the objective function is huge.

In this paper we focus on the study of decomposition methods in order to advantageously exploit the structure of the objective function of (1), in connection with an incremental strategy to take into account that the number m of terms of the objective function may be huge. We propose a decomposition algorithm combined with a gradient incremental strategy, we prove global convergence properties, and we report the results of computational experiments performed on large-scale real test problems.

2. An incremental decomposition algorithm

In order to take into account the fact that the number m of terms of the objective function of (1) may be huge, a first possibility could be that of adopting one of the incremental gradient methods that has been widely studied in the literature (see, e.g., [3,8,11,13]). A recent survey on incremental methods can be found in [4]. The incremental gradient method proposed in [3] can be described as follows

$$\begin{aligned} x^{k+1} &= x^k - \alpha^k \sum_{i=1}^{m} \nabla_x f_i(x^{k,i}, y^{k,i}),\\ y^{k+1} &= y^k - \alpha^k \sum_{i=1}^{m} \nabla_y f_i(x^{k,i}, y^{k,i}), \end{aligned} \tag{4}$$

where $x^{k,0} = x^k, y^{k,0} = y^k$ and

$$x^{k,i} = x^{k,i-1} - \alpha^k \nabla_x f_i(x^{k,i-1}, y^{k,i-1}), \quad i = 1, \ldots, m.$$

$$y^{k,i} = y^{k,i-1} - \alpha^k \nabla_y f_i(x^{k,i-1}, y^{k,i-1}), \quad i = 1, \ldots, m.$$

Global convergence results have been established in [3] under suitable assumptions on the stepsize α^k (see later).

Here, in order to advantageously exploit the convexity of the objective function with respect to the block-component x, we propose a decomposition algorithm based on a gradient incremental strategy. In particular, the iteration of the proposed method takes the form

$$y^{k+1} = y^k - \alpha^k \sum_{i=1}^{m} \nabla_y f_i(x^k, y^{k,i}),$$
$$x^{k+1} \in \arg\min_{x \in R^{n_x}} f(x, y^{k+1}).$$

The algorithm is formally described below.

Algorithm 1. INcremental Decomposition (IND) Algorithm

```
x^0 ∈ R^{n_x}, y^0 ∈ R^{n_y}, k = 0, a sequence {α^k} of positive scalars.
while the stopping criterion is not satisfied do
    y^{k,0} = y^k;
    for i = 1, ..., m do
        y^{k,i} = y^{k,i-1} − α^k ∇f_i(x^k, y^{k,i-1});
    end
    y^{k+1} = y^{k,m};
    x^{k+1} ∈ arg min_{x∈R^{n_x}} f(x, y^{k+1});
    k = k + 1;
end
```