CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

RESEARCH REPORT

ISSN 1213-2365

Sequential Coordinate-wise Algorithm for Non-negative Least Squares Problem

Working document of the EU project COSPAL IST-004176

Vojtěch Franc, Mirko Navara, Václav Hlaváč

{xfrancv,navara,hlavac}@cmp.felk.cvut.cz

CTU–CMP–2005–06

February 22, 2005

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/franc/Franc-TR-2005-06.ps

Authors were supported by the projects IST-004176 COSPAL, CONEX GZ 45.535 and GAČR 102/00/1679.

Research Reports of CMP, Czech Technical University in Prague, No. 6, 2005

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Abstract

This report contributes to the solution of non-negative least squares problem (NLS). The NLS problem is a substantial part of a learning procedure of associative networks. First, stopping conditions suitable for iterative numerical algorithms solving the NLS problem are derived. The conditions allow to control the solution found in terms of optimized objective function. Second, a novel sequential coordinate-wise algorithm is proposed. The algorithm is easy to implement and showed promising performance on synthetical experiments conducted.

1 Introduction

A solution of the non-negative least squares (NLS) problem is required for learning of the associative networks [2, 3]. The NLS problem becomes challenging if a large amount of data are to be processed which makes standard optimization methods infeasible. The oblique-projected Landweber method was proposed to deal with large NLS problems [3]. The oblique-projected Landweber method is a gradient based iterative algorithm which produces a sequence of solutions converging to the optimal one. This report proposes two contributions to the solution the NLS problem: (i) stopping conditions for iterative algorithms which allow to control the precision of found solution in terms of optimized objective function and (ii) a novel sequential coordinate-wise algorithm which is easy to implement and has promising performance on synthetical data.

This work is related to the COgnitive Systems using Perception-Action Learning (COSPAL) project. In the concerete, the associative networks are intended by the group from the Linköping University to model low level signals of the designed system.

The report is organized as follows. The NLS problem to be solved is defined in Section 2. The stopping conditions suitable for iterative algorithms solving the NLS problem are derived in Section 3. A novel sequential coordinate-wise algorithm for the optimization of the NLS problem is proposed in Section 4. A comparison to the oblique-projected Landweber method is given in Section 5. Section 6 describes an experiment comparing the proposed sequential coordinate-wise algorithm to the projected Landweber method. Conclusions are given in Section 7.

Notation used:
Upper-case bold letters denote matrices. Vectors are implicitly columns. Vectors are denoted by lower-case bold italic letters. For instance, $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$ is a matrix made of n column vectors $\boldsymbol{a}_i$, $i \in \mathcal{I}$, where $\mathcal{I} = \{1, \ldots, n\}$ is a set of indices. The non-bold letters are used to denote indices of vectors and matrices. For instance, $\boldsymbol{x} = [x_1, \ldots, x_n]^T$ is a column vector with n entries (coordinates). The notation $[\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_i$ stands for the i-th entry of the vector defined by the term $\mathbf{H}\boldsymbol{x} + \boldsymbol{f}$. The term $\boldsymbol{x} \geq \mathbf{0}$ is a shortcut for a set of inequalities $x_i \geq 0$, $\forall i \in \mathcal{I}$. The expression $\langle \boldsymbol{x}, \boldsymbol{f} \rangle$ stands for the dot (scalar) product between vectors $\boldsymbol{x}$ and $\boldsymbol{f}$.

2 Non-negative least squares problem

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a matrix and $\boldsymbol{b} \in \mathbb{R}^m$ a column vector. The non-negative least squares (NLS) problem is defined as

$$\boldsymbol{x}^* \quad = \quad \operatorname*{argmin}_{\boldsymbol{x} \geq \mathbf{0}} \frac{1}{2} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \,. \tag{1}$$

Without loss of generality, we may assume that all columns $\boldsymbol{a}_i$, $i \in \mathcal{I} = \{1, \ldots, n\}$ of the matrix $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$ are non-zero. A particular instance of the NLS problem (1) arises when all the entries of $\mathbf{A}$ are non-negative. This case matches the problem of learning of associative networks. In this formulation, we are searching for an optimum within an unbounded positive cone in $\mathbb{R}^n$. It is important to restict the search to a bounded set by finding also an upper estimate of the optimal solution $\boldsymbol{x}^*$. In our case

$\boldsymbol{x}^* \leq \boldsymbol{x}^o = [x_1^o, \ldots, x_n^o]^T$, where

$$x_i^o = \max\left(0, \frac{\langle \boldsymbol{a}_i, \boldsymbol{b}\rangle}{\langle \boldsymbol{a}_i, \boldsymbol{a}_i\rangle}\right) , \quad \forall i \in \mathcal{I} .$$

This condition is a result of [3, Theorem 7], where the maximum with 0 has been omitted in [3, formula (41)]; however, the original proof works after this correction. (Possibly a finer estimate could be found.) By $\boldsymbol{e} \in \mathbb{R}^n$ we denote the vector all coordinates of which are 1. We have an upper bound of the sum of entries of $\boldsymbol{x}^*$:

$$\langle \boldsymbol{x}^*, \boldsymbol{e}\rangle = \sum_{i=1}^{n} x_i^* \leq \sum_{i=1}^{n} x_i^o = \langle \boldsymbol{x}^o, \boldsymbol{e}\rangle . \tag{2}$$

Inequality (2) will be important for a derivation of stopping conditions of an iterative algorithm introduced below.

It can be seen that the NLS problem (1) is a special instance of a more general quadratic programming (QP) task with non-negativity constrains. The quadratic objective function is

$$F(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{f}\rangle . \tag{3}$$

The quadratic programming (QP) task with the non-negativity constraints reads

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x}\geq\mathbf{0}} F(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{x}\geq\mathbf{0}} \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{f}\rangle . \tag{4}$$

The solution of the QP task (4) coincides with the solution of the NLS problem (1) if the matrix $\mathbf{H} = \mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n\times n}$ and the vector $\boldsymbol{f} = -\boldsymbol{b}\mathbf{A} \in \mathbb{R}^n$.

The form of task (4) cannot be arbitrary; due to the formulation of the original task (1), $\mathbf{H}$ and $\boldsymbol{f}$ satisfy some special properties:

1. $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ is symmetric and positive semidefinite.
2. $\mathbf{H}_{k,k} = \langle \boldsymbol{a}_k, \boldsymbol{a}_k\rangle > 0$ for all k.
3. The task may have multiple solutions if 0 is an eigenvalue of $\mathbf{H}$; however, the positive solutions are bounded.

Corollary 2.1 *Let $r \in \mathbb{R}$. Then $\{\boldsymbol{x} \in \mathbb{R}^n : F(\boldsymbol{x}) \leq r\}$ is a compact convex set, i.e., it is closed, bounded, and convex.*

The rest of this report deals with this special form of task (4).

3 Stopping conditions

The QP task (4) can be solved by iterative algorithms which produce a sequence of solutions $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, …, $\boldsymbol{x}^{(t)}$ converging to the optimal solution $\boldsymbol{x}^*$. There is a need to stop the algorithm when the current solution $\boldsymbol{x}^{(t)}$ is sufficiently close to the optimal $\boldsymbol{x}^*$. Two possible stopping conditions will be introduced. First, the stopping conditions based on the Karush-Kuhn-Tucker (KKT) conditions will be described in Section 3.1. Second, the stopping conditions based on lower and upper bounds of the optimal value $F(\boldsymbol{x}^*)$ will be derived in Section 3.2.

3.1 Karush-Kuhn-Tucker conditions

The objective function (3) is convex as the matrix $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ is symmetric and positive semidefinite. The constraints $\boldsymbol{x} \geq \mathbf{0}$ define also a convex feasible set. As both the objective function and the feasible set are convex, the QP task (4) is convex as well. In the case of the convex optimization task, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for the optimal solution [1]. The KKT conditions for the QP task (4) have a particularly simple form introduced below.

The Lagrange function for the task (4) reads

$$L(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{f}\rangle - \langle \boldsymbol{x}, \boldsymbol{\mu}\rangle \,, \tag{5}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ are Lagrange multipliers (or dual variables). The conditions of the QP task (4) are defined as

$$\begin{aligned} \frac{\partial L(\boldsymbol{x}, \boldsymbol{\mu})}{\partial \boldsymbol{x}} \quad = \quad \mathbf{H}\boldsymbol{x} + \boldsymbol{f} - \boldsymbol{\mu} \quad &= \quad \mathbf{0} \,, \\ \boldsymbol{x} \quad &\geq \quad \mathbf{0} \,, \\ \boldsymbol{\mu} \quad &\geq \quad \mathbf{0} \,, \\ \langle \boldsymbol{x}, \boldsymbol{\mu}\rangle \quad &= \quad 0 \,. \end{aligned} \tag{6}$$

Any vector $\boldsymbol{x}$ which satisfies the KKT conditions (6) is the optimal solution of the QP task (4) and vice versa.

Let $\mathcal{X}^* \subset \mathbb{R}^n$ denote the set of vectors which satisfy the KKT conditions (6), i.e., any $\boldsymbol{x}^* \in \mathcal{X}^*$ is the solution of the task (4) for some $\boldsymbol{\mu}$. Notice that the set $\mathcal{X}^*$ is convex and it contains just one vector if the matrix $\mathbf{H}$ is positive definite. Reasonable stopping conditions for an iterative algorithm can be derived by introducing a relaxed version of the KKT conditions. The ε-KKT conditions are defined as a set of linear inequalities

$$\begin{aligned} \boldsymbol{x} \quad &\geq \quad \mathbf{0} \,, \\ [\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_i \quad &\geq \quad -\varepsilon \,, \qquad \text{for} \quad i \in \mathcal{I} = \{1, \ldots, n\} \,, \\ [\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_i \quad &\leq \quad \varepsilon \,, \qquad \text{for} \quad i \in \mathcal{I}_\emptyset = \{i \in \mathcal{I} \colon x_i > 0\} \,, \end{aligned} \tag{7}$$

where $\varepsilon > 0$ is a constant which defines a precision of the solution. Let $\mathcal{X}^\varepsilon \subset \mathbb{R}^n$ be a set of vectors which satisfy the conditions (7). It is easy to show that $\mathcal{X}^* \subseteq \mathcal{X}^\varepsilon$ holds generally and $\mathcal{X}^* = \mathcal{X}^\varepsilon$ holds for $\varepsilon = 0$.

The ε-KKT conditions are easy to evaluate and they can be used as an indicator that the current solution is close to the optimal one. It is not immediately seen, however, how the solution satisfying the ε-KKT conditions corresponds to the optimal $\boldsymbol{x}^*$ in terms of the optimized function $F(\boldsymbol{x})$. This drawback is removed after introducing a lower bound $LB(\boldsymbol{x})$ of the optimal value $F(\boldsymbol{x}^*)$ which shall be derived below.

3.2 Bounds of the optimal solution

In this section, we exclude the (possible) trivial solution $\boldsymbol{x}^* = 0$. If the optimum is obtained at 0, we easily find it by a test of the inputs, or after the first step (starting from 0 as the initial estimate, we obtain it as the next approximation and a fixed point).

Let the vector $\nabla F(\boldsymbol{x}^*)$ be the gradient of the function $F(\boldsymbol{x})$ evaluated at the vector $\boldsymbol{x}^*$. It follows from the convexity of the function $F(\boldsymbol{x})$ that

$$\begin{aligned} F(\boldsymbol{x}^*) + \langle (\boldsymbol{x} - \boldsymbol{x}^*), \nabla F(\boldsymbol{x}^*)\rangle \quad &\leq \quad F(\boldsymbol{x}) \,, \\ \frac{1}{2}\langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x}^*\rangle + \langle \boldsymbol{x}^*, \boldsymbol{f}\rangle + \langle (\boldsymbol{x} - \boldsymbol{x}^*), (\mathbf{H}\boldsymbol{x}^* + \boldsymbol{f})\rangle \quad &\leq \quad \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{f}\rangle \,, \end{aligned}$$

which can be further rearranged to

$$\langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x} + \boldsymbol{f}\rangle - \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle \leq \frac{1}{2}\langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x}^*\rangle + \langle \boldsymbol{x}^*, \boldsymbol{f}\rangle \,. \tag{8}$$

Since the entries of the optimal vector $\boldsymbol{x}^*$ are non-negative, the following inequality holds

$$\langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x} + \boldsymbol{f}\rangle \geq \langle \boldsymbol{x}^*, \boldsymbol{e}\rangle \min_{i \in \mathcal{I}} [\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_i \,. \tag{9}$$

Inequalities (8) and (9) give a lower bound

$$\underbrace{\langle \boldsymbol{x}^*, \boldsymbol{e}\rangle \min_{i \in \mathcal{I}} [\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_i - \frac{1}{2}\langle \boldsymbol{x}, \mathbf{H}\boldsymbol{x}\rangle}_{LB(\boldsymbol{x})} \leq \underbrace{\frac{1}{2}\langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x}^*\rangle + \langle \boldsymbol{x}^*, \boldsymbol{f}\rangle}_{F(\boldsymbol{x}^*)} \,. \tag{10}$$

Equality is obtained for the optimal solution vector $\boldsymbol{x}^*$, i.e., $LB(\boldsymbol{x}^*) = F(\boldsymbol{x}^*)$ holds true which follows from the equalities

$$\min_{i\in\mathcal{I}}[\mathbf{H}\boldsymbol{x}^* + \boldsymbol{f}]_i = 0 \quad \text{and} \quad \langle \boldsymbol{x}^*, \mathbf{H}\boldsymbol{x}^*\rangle + \langle \boldsymbol{x}^*, \boldsymbol{f}\rangle = 0\,,$$

derived directly from the KKT conditions (6). (The former equality is based on the fact that there is at least one $i \in \mathcal{I}$ such that $[\mathbf{H}\boldsymbol{x}^* + \boldsymbol{f}]_i = 0$. Otherwise, $\boldsymbol{x}^* = 0$; this case can be easily recognized and it was excluded by our assumption.)

The lower bound (10) is valid for an arbitrary optimization task (4). The bound depends on a generally unknown term $\langle \boldsymbol{x}^*, \boldsymbol{e}\rangle$. However, an upper bound of $\langle \boldsymbol{x}^*, \boldsymbol{e}\rangle$ can be derived for a special instance of the task (4) which was mentioned in Section 2. Provided the term $\langle \boldsymbol{x}^*, \boldsymbol{e}\rangle$ is known (or its upper bound is known) then the lower bound $LB(\boldsymbol{x})$ can be evaluated and used as a stopping condition of an iterative algorithm. A reasonable stopping condition reads

$$F(\boldsymbol{x}) - F(\boldsymbol{x}^*) \leq \delta\,, \tag{11}$$

where $\delta > 0$ is a constant which limits the distance between vectors $\boldsymbol{x}$ and $\boldsymbol{x}^*$ in terms of the optimized criterion. The stopping condition (11) is satisfied if the inequality

$$F(\boldsymbol{x}) - LB(\boldsymbol{x}) \leq \delta\,, \tag{12}$$

holds which could be evaluated provided the lower bound (10) is known.

4 Sequential coordinate-wise algorithm

This section describes a novel (according to the authors' knowledge) sequential coordinate-wise algorithm for optimization of the task (4). The algorithm produces a sequence of solutions $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}$ which converges to the optimal $\boldsymbol{x}^*$. The idea is to optimize in each iteration with respect to a single coordinate while the remaining coordinates are fixed. The optimization with respect to a single coordinate has an analytical solution, thus it can be computed efficiently.

Let $x_k \in \mathbb{R}$ be the k-th coordinate of the vector $\boldsymbol{x} = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$. The objective function $F(\boldsymbol{x})$ can be equivalently rewritten as follows

$$\begin{aligned}
F(\boldsymbol{x}) &= \frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} x_i x_j H_{i,j} + \sum_{i\in\mathcal{I}} x_i f_i \\
&= \frac{1}{2}x_k^2 H_{k,k} + x_k f_k + x_k \sum_{i\in\mathcal{I}\setminus\{k\}} x_i H_{i,k} + \sum_{i\in\mathcal{I}\setminus\{k\}} x_i f_i + \frac{1}{2}\sum_{i\in\mathcal{I}\setminus\{k\}}\sum_{j\in\mathcal{I}\setminus\{k\}} x_i x_j H_{i,j} \\
&= \frac{1}{2}x_k^2\alpha + x_k\beta + \gamma\,,
\end{aligned}$$

where

$$\begin{aligned}
\alpha &= H_{k,k}\,, \\
\beta &= f_k + \sum_{i\in\mathcal{I}\setminus\{k\}} x_i H_{i,k} = [\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_k - H_{k,k}x_k\,, \\
\gamma &= \sum_{i\in\mathcal{I}\setminus\{k\}} x_i f_i + \frac{1}{2}\sum_{i\in\mathcal{I}\setminus\{k\}}\sum_{j\in\mathcal{I}\setminus\{k\}} x_i x_j H_{i,j}\,.
\end{aligned}$$

The optimization of $F(\boldsymbol{x})$ with respect to a selected x_k has an analytical solution

$$\begin{aligned}
x_k^* &= \operatorname*{argmin}_{x_k\geq 0} F([x_1, \ldots, x_k, \ldots, x_n]^T) \\
&= \operatorname*{argmin}_{x_k\geq 0} \frac{1}{2}x_k^2\alpha + x_k\beta + \gamma \\
&= \max\left(0, -\frac{\beta}{\alpha}\right) \\
&= \max\left(0, x_k - \frac{[\mathbf{H}\boldsymbol{x} + \boldsymbol{f}]_k}{H_{k,k}}\right)\,.
\end{aligned}$$

The iterative algorithm derived in the sequel updates a single variable x_k in each iteration, i.e.,

$$x_i^{(t+1)} = x_i^{(t)}\,, \forall i \in \mathcal{I} \setminus \{k\} \qquad \text{and} \qquad x_k^{(t+1)} \neq x_k^{(t)}\,. \tag{13}$$

The formula for the update requires the gradient $\boldsymbol{\mu}^{(t)} = \mathbf{H}\boldsymbol{x}^{(t)} + \boldsymbol{f}$. We recommend to update the vector $\boldsymbol{\mu}^{(t)}$ in each iteration instead of computing it from a scratch. Thanks to (13) the update can be written as

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} + \left(x_k^{(t+1)} - x_k^{(t)}\right)\boldsymbol{h}_k\,, \tag{14}$$

where $\boldsymbol{h}_k$ is the kth column of the matrix $\mathbf{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n]$. (In fact, the original formula for β has the same order of complexity, because we need only one coordinate of the gradient. However, the latter formula allows to compute the whole gradient which is needed for stopping conditions.) The proposed iterative algorithm to solve the task (4) is the following:

Algorithm 1: Sequential Coordinate-wise Algorithm

1. Initialization. Set $\boldsymbol{x}^{(0)} = \mathbf{0}$ and $\boldsymbol{\mu}^{(0)} = \boldsymbol{f}$.
2. Repeat until the stopping condition is satisfied:

   For $k = 1$ to n

$$x_k^{(t+1)} = \max\left(0, x_k^{(t)} - \frac{\mu_k^{(t)}}{H_{k,k}}\right) \quad \text{and} \quad x_i^{(t+1)} = x_i^{(t)}\,, \forall i \in \mathcal{I} \setminus \{k\}\,,$$
$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} + \left(x_k^{(t+1)} - x_k^{(t)}\right)\boldsymbol{h}_k\,.$$

   (The latter formula is skipped if $x_k^{(t+1)} = x_k^{(t)}$ which is the case quite often.)

Algorithm 1 requires $O(n)$ computations for each update from $\boldsymbol{x}^{(t)}$ to $\boldsymbol{x}^{(t+1)}$. The gradient vector $\boldsymbol{\mu}^{(t)}$ is known in each iteration which can be employed for evaluation of the stopping conditions. The stopping conditions are evaluated after all n coordinates were updated. Section 3 describes two different stopping conditions which can be used to halt the algorithm. It is obvious that in Algorithm 1 the objective function $F(\boldsymbol{x}^{(t)})$ decreases or remains unchanged, however, we have not found a proof of its convergence yet. We have at least the following observation:

Proposition 4.1 *All fixed points of Algorithm 1 are optimal solutions of task* (4).

PROOF: Suppose that $\boldsymbol{x}^{(t)}$ is a fixed point of Algorithm 1, i.e., $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}$. This means that for each $k \in \mathcal{I}$ one of the following conditions holds:

1. $\mu_k^{(t)} = 0$,
2. $\mu_k^{(t)} > 0$ and $x_k^{(t)} = 0$.

Thus the KKT conditions are satisfied for $\boldsymbol{x}^{(t)}, \boldsymbol{\mu}^{(t)}$. ■

The assumptions of Proposition 4.1 can be weakened:

Corollary 4.2 *If in Algorithm 1 we obtain* $F(\boldsymbol{x}^{(t+1)}) = F(\boldsymbol{x}^{(t)})$ *then* $\boldsymbol{x}^{(t)}$ *is the optimal solution of task* (4).

PROOF: Whenever the conditions from the proof of Proposition 4.1 are violated, the criterion strictly decreases. Thus it remains unchanged only if we are at a fixed point of the algorithm. ■

5 Comparison to oblique-projected Landweber method

The oblique-projected Landweber method modifed to solve the NLS problem is described in [3]. The method is based on a gradient descent minimization with solution projected to the feasible set. The algorithm for optimization of the task (4) based on the oblique-projected Landweber method is the following:

Algorithm 2: Oblique-projected Landweber Algorithm

1. Initialization. Set $\boldsymbol{x}^{(0)} = \mathbf{0}$ and $\boldsymbol{d} = \mathbf{H}\boldsymbol{e}$.
2. Repeat until the stopping condition is satisfied:

$$\begin{aligned} \boldsymbol{\mu}^{(t)} &= \mathbf{H}\boldsymbol{x}^{(t)} + \boldsymbol{f}\,, \\ x_i^{(t+1)} &= \max\left(0, x_i^{(t)} - \frac{\mu_i^{(t)}}{d_i}\right)\,, \quad \forall i \in \mathcal{I}\,. \end{aligned}$$

An updated vector $\boldsymbol{x}^{(t+1)}$ is computed from the previous solution $\boldsymbol{x}^{(t)}$ by moving in the direction opposite to the gradient $\boldsymbol{\mu}^{(t)} = \mathbf{H}\boldsymbol{x}^{(t)} + \boldsymbol{f}$. The coordinates of $\boldsymbol{\mu}^{(t)}$ are scaled by the vector $\boldsymbol{d}$. Comparison between the sequential coordinate-wise Algorithm 1 (SCA) and the oblique-projected Landweber Algorithm 2 (LA) shows two differences:

(i) The SCA recomputes the gradient after each update of a single variable unlike the LA which recomputes the gradient after all n variables are updated.

(ii) The SCA updates individual variables sequentially and each update is optimal with respect to the corresponding variable. The LA updates all variables simultaneously without optimizing the size of the step along the gradient.

Both algorithms require $O(n)$ operations for an update of a single variable. In cotrast to this, any modification of Newton's method requires a matrix inversion with complexity $O(n^3)$. This can hardly be balanced by a reduction of the number of steps.

6 Experiments

This section describes an experiment carried out on a synthetical data. The problem selected is to train an associative network with channel based representation of input and output signals. For more information about the associative networks we refer to [2, 3]. The training data are sampled from a sine function with added Gaussian noise, i.e., $f(x) = \sin(\pi x) + N(0, 0.2)$, where $N(0, 0.2)$ stands for Gaussian distribution with zero mean and standard deviation equal to 0.2. The number of 4000 training samples is generated. The input layer contained 1000 channels and the output layer 10 channels with a cosine kernel. This setting results into 10 training problems of the form (4) with the number of $n = 1000$ variables.

The proposed sequential coordinate-wise Algorithm 1 (SCA) is compared to the oblique-projected Landweber Algorithm 2 (LA). Matlab implementation was used for all the experiments. The data matrix contains only positive entries which allows to evaluate the lower bound on $F(\boldsymbol{x}^*)$ and to use the stopping condition (12). The stopping condition was

$$F(\boldsymbol{x}^{(t)}) - F(\boldsymbol{x}^*) \leq F(\boldsymbol{x}^{(t)}) - F(\boldsymbol{x}^*) \leq 10^{-6}\,. \tag{15}$$

The aim was to measure the convergence speed of the tested algorithms. The speed is measured in terms of (i) the number of updates required for convergence and (ii) an estimate of the required CPU time on common PC with 1.80GHz processor [1].

[1] It is worth mentioning that the CPU time estimate is biased to the benefit of the LA algorithm because of the following implementation reasons. The critical part of the LA algorithm can be implemented by a single matrix multiplication unlike the SCA algorithm which requires using a loop in Matlab language. This influences the measure of the computational time because the matrix multiplication in Matlab is much more effective than loops.

The sample data and their approximation by the trained associative network can be seen in Figure 1(a). Figure 1(b) shows the sparseness of 10 tasks used for testing. The sparseness is measured by the number of non-zero variables of the solution vector for the corresponding QP task.

The comparison of the convergence speed can be seen in Figure 2. The number of updates per single variable can be seen in Figure 2(a). Figure 2(b) shows an estimate of the the require CPU time. These values are measured for all 10 problems separately. The SCA turned out to be in average three time faster compared to the LA.

The convergence of both the algorithms on the 5th problem is displayed in Figure 3. The figure shows the upper and the lower bound on the optimal value of the objective function with respect to the number of updates. The upper bound corresponds to the value of the objective function for current solution.

7 Conclusions and future work

This report describes two contributions to the problem of solving the non-negative least squares (NLS) problem. First, stopping conditions suitable for iterative algorithms solving the NLS problem were proposed. The stopping conditions allow to control the precision of the solution found in terms of the optimized objective function. Second, a sequential coordinate-wise algorithm to solve the NLS problem was proposed. The algorithm is easy to implement and showed promising performance. The proposed algorithm outperformed the oblique-projected Landweber method which has been used to solve the NLS problem. The methods were benchmarked on a synthetical data.

The future research should address the issue of proving convergence of the proposed sequential coordinate-wise algorithm. A more thorough experimental evaluation of the proposed algorithm and comparison to other methods is also needed.

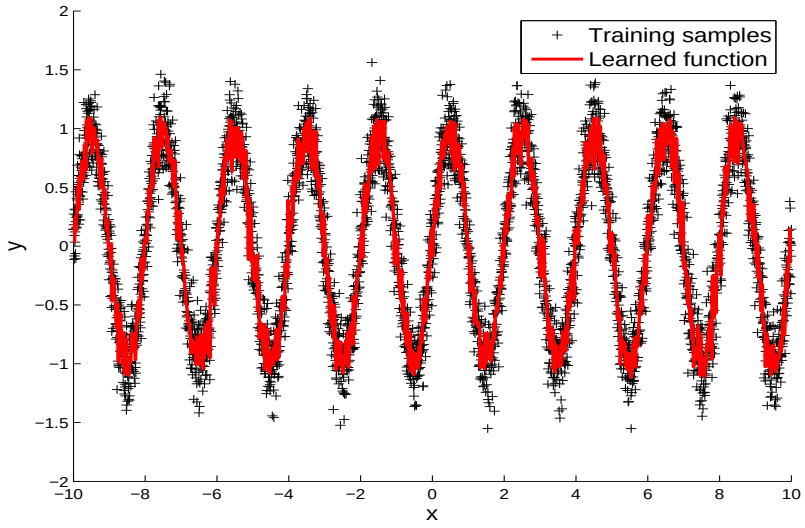
References

[1] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, New York, USA, 2nd edition, 1990.

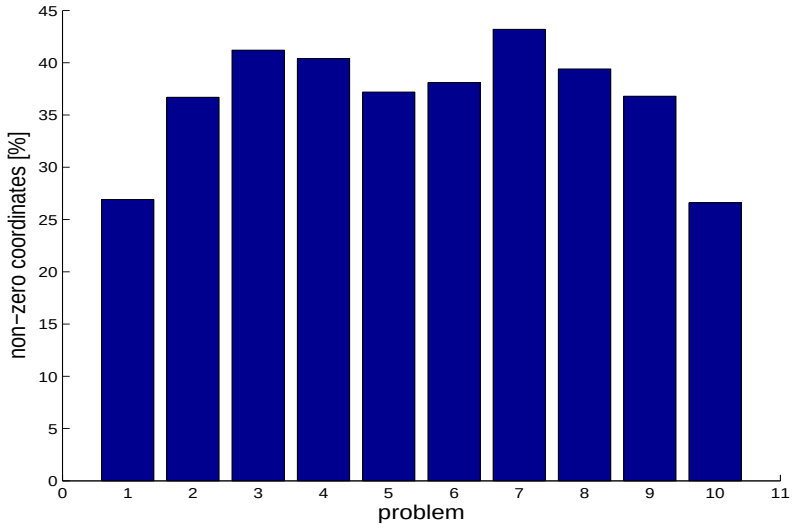
[2] G. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings of Algebraic Frames for the Perception-Action cycle (AFPAC)*, Kiel, Germany, September 2000.

[3] B. Johansson, T. Elfving, V. Kozlov, T. Censor, and G. Granlund. The application of an oblique-projected Landweber method to a model of supervised learning. Technical Report LiTH-ISY-R-2623, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, September 2004.

Figure 1: Figure (a) shows the sample data and their approximation by the associative network. Figure (b) shows the number of non-zero variables of the solution vectors of 10 quadratic programming problems.

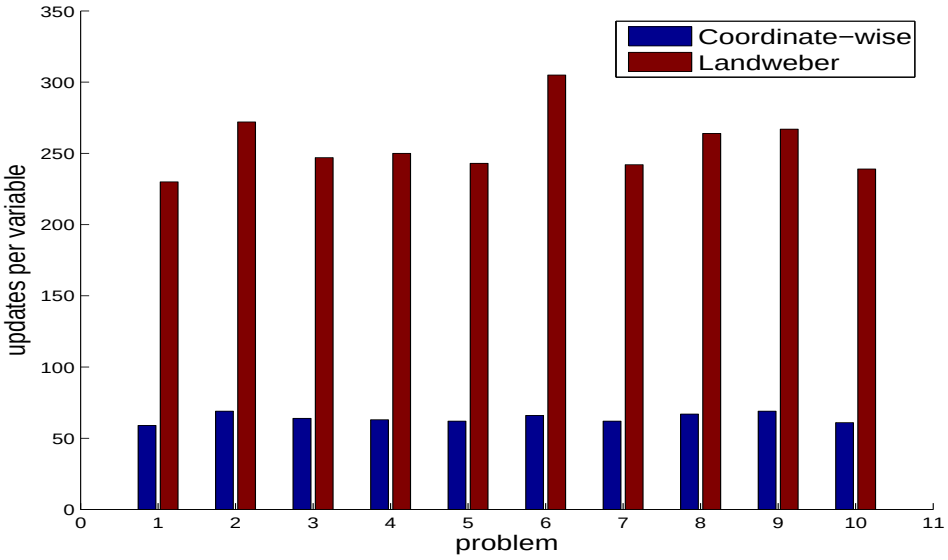


(a)

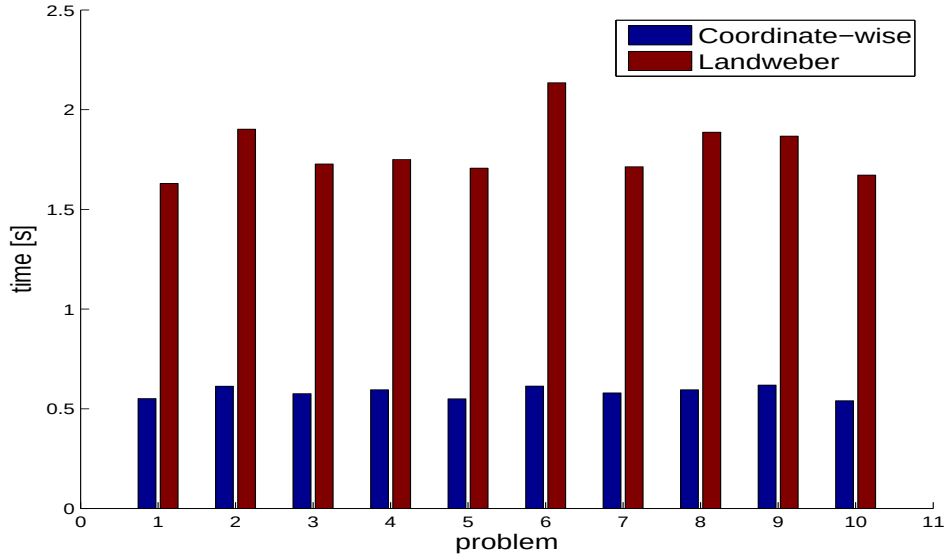


(b)

Figure 2: Figure (a) shows the number of updates per single variable required for convergence. Figure (b) shows an estimate of the required CPU time. The measures are displayed for 10 different quadratic programming tasks separately.



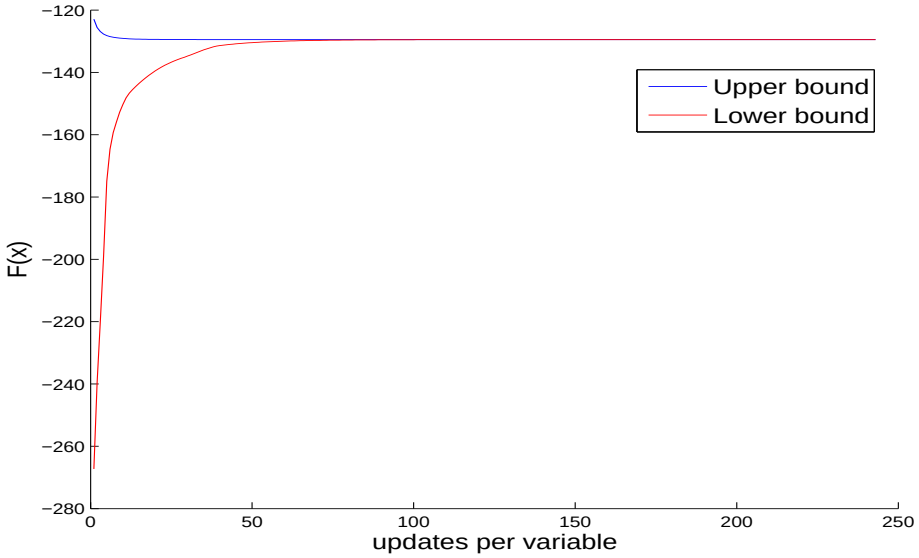
(a)



(b)

Figure 3: The figure shows the convergence of the sequential coordinate-wise algorithm and the oblique-projected Landweber method on the 5th training problem. The plot of the upper bound and the lower bound of the optimal value is displayed with respect to the number of updates.

Oblique-projected Landweber Method



Sequential Coordinate-wise Algorithm