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

Vojtěch Franc, Václav Hlaváč, and Mirko Navara

Center for Machine Perception, Department of Cybernetics,
Faculty of Electrical Engineering, Czech Technical University,
{xfrancv,hlavac,navara}@cmp.felk.cvut.cz,
http://cmp.felk.cvut.cz

Abstract. This paper contributes to the solution of the non-negative least squares problem (NNLS). The NNLS problem constitutes a substantial part of many computer vision methods and methods in other fields, too. We propose a novel sequential coordinate-wise algorithm which is easy to implement and it is able to cope with large scale problems. We also derive stopping conditions which allow to control the distance of the solution found to the optimal one in terms of the optimized objective function. The proposed algorithm showed promising performance in comparison to the projected Landweber method.

1 Introduction

A common approach of fitting model parameters to data is formalized as the least squares problem. There are situations in which additional constraints forcing the fitted parameters to be non-negative are useful. This leads to the non-negative least squares problem (NNLS). The non-negativity constraints are beneficial for the problems in which the negative values of the fitted parameters do not correspond to the physical reality, e.g., the problems dealing with pixel values in image modeling. The non-negativity constraints can also be used to introduce regularization for ill-posed problems. Examples of using NNLS in computer vision include, for instance, object recognition with unknown lighting conditions [1], image restoration [2] or tracking [3]. Learning of associative neural networks [5, 6] is another task which can be expressed as the NNLS problem. This work has been motivated by the project COgnitive Systems using Perception-Action Learning (COSPAL http:\\www.cospal.org) in which associative networks play a substantial role in modeling low-level signals of the designed robotic system.

The NNLS problem becomes challenging if a large amount of data is to be processed, which makes standard optimization methods infeasible, e.g., the method by Lawson and Hanson [7]. The projected Landweber method was proposed to deal with large NNLS problems [6]. The projected Landweber method is a gradient-based iterative algorithm which produces a sequence of solutions converging to the optimal one. This paper proposes two contributions to the solution of the NNLS problem: (i) stopping conditions for iterative algorithms

A. Gagalowicz and W. Philips (Eds.): CAIP 2005, LNCS, pp. 407-414. Springer-Verlag Berlin 2005.

which allow to control the precision of the found solution in terms of the optimized objective function and (ii) a novel sequential coordinate-wise algorithm which is easy to implement and has promising performance on synthetical data.

The paper is organized as follows. The NNLS problem is defined in Section 2. The stopping conditions suitable for iterative algorithms solving the NNLS problem are derived in Section 3. A novel sequential coordinate-wise algorithm which solves the NNLS problem is proposed in Section 4. Section 5 describes an experiment comparing the proposed sequential coordinate-wise algorithm to the projected Landweber method. Conclusions are given in Section 6.

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 entries. The non-bolded 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 ith 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 (inner) product of 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 (NNLS) problem is defined as

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

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 NNLS problem (1) arises when all entries of $\mathbf{A}$ are non-negative. This case matches the problem of learning of associative networks. In this formulation, we are searching for the optimum within an unbounded positive cone in $\mathbb{R}^n$. It is important to restrict 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 [6, Theorem 7], where the maximum with 0 has been omitted in [6, formula (41)]; however, the original proof works after this correction. By $\boldsymbol{e} \in \mathbb{R}^n$ we denote the vector with all coordinates equal to 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 stopping conditions of an iterative algorithm introduced below.

It can be seen that the NNLS 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 QP task with the non-negativity constraints reads

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

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

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.

The rest of this paper 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 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 a convex optimization task, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for the optimal solution (see [4]). The KKT conditions for the QP task (4) have a particularly simple form introduced below.

The Lagrange function for 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). We obtain conditions

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

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

Let $\mathcal{X}^* \subset \mathbb{R}^n$ denote the set of vectors which satisfy (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

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

where $\varepsilon > 0$ is a constant defining the precision of the solution. Let $\mathcal{X}^\varepsilon \subset \mathbb{R}^n$ be the set of vectors which satisfy conditions (7). It is easy to show that $\mathcal{X}^* \subseteq \mathcal{X}^\varepsilon$ holds in general 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 $F(\boldsymbol{x}^*)$ derived in the sequel.

3.2 Bounds of the optimal solution

In this section, we exclude the (possible) trivial solution $\boldsymbol{x}^* = \mathbf{0}$. If the optimum is obtained at $\mathbf{0}$, we find it easily by a test of the inputs or after the first step (starting from $\mathbf{0}$ 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 evaluated at $\boldsymbol{x}^*$. It follows from the convexity of the function F that

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

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 in (10) 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 has been 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, the upper bound of $\langle \boldsymbol{x}^*, \boldsymbol{e} \rangle$ can be derived for a special instance of task (4) which was specified in Section 2. Provided the term $\langle \boldsymbol{x}^*, \boldsymbol{e} \rangle$ (or its upper bound) is known 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). Without the positivity constraint, our method coincides with the Gauss-Seidel method which is known to converge if $\mathbf{H}$ is positive definite. 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 $\boldsymbol{x} = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ and $\mathcal{I}_k = \mathcal{I} \setminus \{k\}$. The objective function $F(\boldsymbol{x})$ can be equivalently rewritten as

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

where

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

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

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

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

$$x_i^{(t+1)} = x_i^{(t)} \,, \forall i \in \mathcal{I}_k \,. \tag{13}$$

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 the 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 task (4) is the following:

Algorithm 1: Sequential Coordinate-wise Algorithm for NNLS (abbrev. SCA)

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

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

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

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 the 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 the objective function $F(\boldsymbol{x}^{(t)})$ decreases or remains unchanged in Algorithm 1, however, we have not found a proof of its convergence yet. We have the following observation at least:

Proposition 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 $k \in \mathcal{I}$ either $\mu_k^{(t)} = 0$ or ($\mu_k^{(t)} > 0$ and $x_k^{(t)} = 0$) hold. Thus the KKT conditions are satisfied for $\boldsymbol{x}^{(t)}, \boldsymbol{\mu}^{(t)}$. ■

5 Experiments

This section outlines an experiment carried out on synthetical data. The problem selected is to train an associative network with channel-based representation of input and output signals. We refer to [5, 6] for more information about associative networks. The adopted setting results into 10 training problems of the form (4) with the number of $n = 2500$ variables.

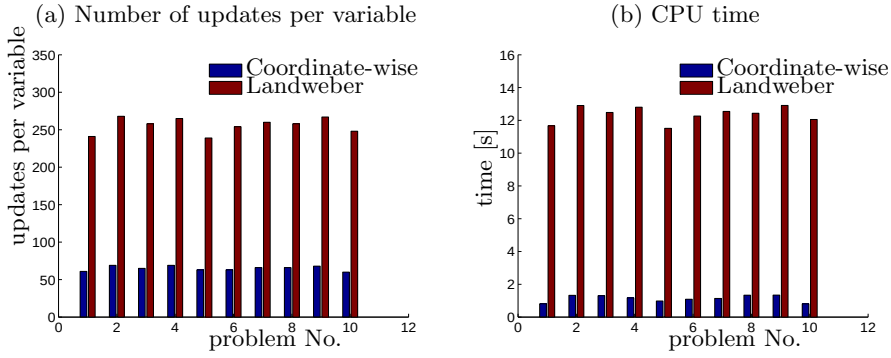
The proposed sequential coordinate-wise Algorithm 1 (SCA) is compared to the projected Landweber Algorithm [6] (LA). The Matlab implementation was used in all experiments. The data matrix contains only positive entries, which allows to evaluate the lower bound on $F(\boldsymbol{x}^*)$ and to use the stopping condition (12). The stopping condition $F(\boldsymbol{x}^{(t)}) - F(\boldsymbol{x}^*) \leq 10^{-6}$ was used. We measured the speed of convergence in terms of (i) the number of updates required for convergence and (ii) an estimate of the required CPU time on the common PC with Intel Pentium IV 2.80GHz processor.

The comparison of the convergence speed can be seen in terms of the number of iterations and the required CPU time can be seen in Figure 1. These values are measured for all 10 problems separately. The SCA turned out to be on average more than ten times faster compared to the LA.

6 Conclusions

This paper describes two contributions to the problem of solving the non-negative least squares (NNLS) problem. First, stopping conditions suitable for iterative algorithms solving the NNLS 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 NNLS problem was proposed. The algorithm is easy to implement and showed promising performance. The proposed algorithm outperformed the projected Landweber method which has been used to solve the NNLS problem. The methods were benchmarked on synthetical data.

Fig. 1. Comparison between the projected Landweber method and the sequential coordinate-wise algorithm on 10 different NNLS problems having 2500 variables each.



Acknowledgments

The authors would like to thank Gösta Granlund, Michael Felsberg, Per-Erik Forssén, and Erik Jonsson for introducing to the problem and for fruitful discussions. The authors were supported by the projects IST-004176 COSPAL, CONEX GZ 45.535. The first author was also supported by MSM 6840770013. The second author was also supported by INTAS 04-77-7347 (PRINCESS).

References

1. R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 25(2):218–233, 2003.
2. M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, Great Britain, 1998.
3. J.E. Boyd and J. Meloche. Evaluation of statistical and multiple-hypothesis tracking for video surveillance. *Machine Vision and Applications*, 13(5-6):244–351, 2003.
4. R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, USA, 2nd edition, 1990.
5. 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.
6. 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.
7. C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.