Linear Algebra and its Applications 469 (2015) 419–439

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Accelerating the alternating projection algorithm for the case of affine subspaces using supporting hyperplanes

C.H. Jeffrey Pang

Department of Mathematics, National University of Singapore, Block S17 08-11, 10 Lower Kent Ridge Road, Singapore 119076, Singapore

ARTICLE INFO

Article history:
Received 16 July 2014
Accepted 28 November 2014
Available online 11 December 2014
Submitted by H. Bart

MSC:
11D04
90C59
47J25
47A46
47A50
52A20
41A50

Keywords:
Best approximation problem
Alternating projections
Supporting hyperplanes

ABSTRACT

The von Neumann–Halperin method of alternating projections converges strongly to the projection of a given point onto the intersection of finitely many closed affine subspaces. We propose acceleration schemes making use of two ideas: Firstly, each projection onto an affine subspace identifies a hyperplane of codimension 1 containing the intersection, and secondly, it is easy to project onto a finite intersection of such hyperplanes. We give conditions for which our accelerations converge strongly. Finally, we perform numerical experiments to show that these accelerations perform well for a matrix model updating problem.

E-mail address: matpchj@nus.edu.sg.

http://dx.doi.org/10.1016/j.laa.2014.11.035

1. Introduction

Let X be a (real) Hilbert space, and let $M_1, M_2, \ldots, M_k$ be a finite number of closed linear subspaces with $M := \bigcap_{l=1}^k M_l$. For any closed subspace N of X, let P_N denote the orthogonal projection onto N. The von Neumann–Halperin method of alternating projections, or MAP for short, is an iterative algorithm for determining the best approximation $P_M x$, the projection of x onto M. We recall their theorem on the strong convergence of the MAP below.

Theorem 1.1. *(See von Neumann [20] for $k = 2$, Halperin [12] for $k \geq 2$.) Let $M_1, M_2, \ldots, M_k$ be closed subspaces in the Hilbert space X and let $M := \bigcap_{l=1}^k M_l$. Then*

$$\lim_{n\to\infty} \left\| (P_{M_k} P_{M_{k-1}} \cdots P_{M_1})^n x - P_M x \right\| = 0 \quad \text{for all } x \in X. \tag{1.1}$$

In the case where $k = 2$, this result was rediscovered numerous times.

The method of alternating projections, as suggested in the formula (1.1), guarantees convergence to the projection $P_M x$, but the convergence is slow in practice. Various acceleration schemes have been studied in [11,10,4]. An identity for the convergence of the method of alternating projections in the case of linear subspaces is presented in [21]. Other recent works on accelerating alternating projections not already mentioned in [18] include [13,6]. In fact [13] treats the particular case of accelerating alternating projections for the case of finitely many linear subspaces, just like the case treated in this paper. They use a novel way of minimizing an appropriate quadratic function.

We remark that the Boyle–Dykstra Theorem [5] generalizes the strong convergence to the projection in Theorem 1.1 to Dykstra's algorithm [8], where the M_l do not have to be linear subspaces.

When the sets M_l are not linear subspaces, a simple example using a halfspace and a line in $\mathbb{R}^2$ shows that the method of alternating projections may not converge to the projection $P_M x$. Nevertheless, the method of alternating projections is still useful for the SIP (Set Intersection Problem). When $M_1, M_2, \ldots, M_k$ is a finite number of closed (not necessarily convex) subsets of a Hilbert space X, the SIP is the problem of finding a point in $M := \bigcap_{l=1}^k M_l$, i.e.,

$$\text{(SIP):} \quad \text{Find } x \in M := \bigcap_{l=1}^{k} M_l, \quad \text{where } M \neq \emptyset. \tag{1.2}$$

An acceleration of the method of alternating projections for the case where each M_l were closed convex sets (but not necessarily subspaces) was studied in [18] and improved in [17]. The idea there, named as the SHQP strategy (Supporting Halfspaces and Quadratic Programming) was to store each of the halfspace produced by the projection process, and use quadratic programming to project onto an intersection of a reasonable number of these halfspaces.