# Alternating Projections

Stephen Boyd and Jon Dattorro
EE392o, Stanford University

Autumn, 2003

## 1 Alternating projection algorithm

*Alternating projections* is a very simple algorithm for computing a point in the intersection of some convex sets, using a sequence of projections onto the sets. Like a gradient or subgradient method, alternating projections can be slow, but the method can be useful when we have some efficient method, such as an analytical formula, for carrying out the projections. In these notes, we use only the Euclidean norm, Euclidean distance, and Euclidean projection.

Suppose $C$ and $D$ are closed convex sets in $\mathbf{R}^n$, and let $P_C$ and $P_D$ denote projection on $C$ and $D$, respectively. The algorithm starts with any $x_0 \in C$, and then alternately projects onto $C$ and $D$:

$$y_k = P_D(x_k), \qquad x_{k+1} = P_C(y_k), \qquad k = 0, 1, 2, \ldots$$

This generates a sequence of points $x_k \in C$ and $y_k \in D$.

A basic result, due to Cheney and Goldstein [CG59] is the following. If $C \cap D \neq \emptyset$, then the sequences $x_k$ and $y_k$ both converge to a point $x^* \in C \cap D$. Roughly speaking, alternating projections finds a point in the intersection of the sets, provided they intersect. Note that we're not claiming the algorithm produces a point in $C \cap D$ in a finite number of steps. We are claiming that the sequence $x_k$ (which lies in $C$) satisfies $\mathbf{dist}(x_k, D) \to 0$, and likewise for $y_k$. A simple example is illustrated in figure 1.

Alternating projections is also useful when the sets do not intersect. In this case we can prove the following. Assume the distance between $C$ and $D$ is achieved (*i.e.*, there exist points in $C$ and $D$ whose distance is $\mathbf{dist}(C, D)$). Then $x_k \to x^* \in C$, and $y_k \to y^* \in D$, where $\|x^* - y^*\|_2 = \mathbf{dist}(C, D)$. In other words, alternating projections yields a pair of points in $C$ and $D$ that have minimum distance. In this case, alternating projections also yields (in the limit) a hyperplane that separates $C$ and $D$. A simple example is illustrated in figure 2.

There are many variations on and extensions of the basic alternating projections algorithm. For example, we can find a point in the intersection of $k > 2$ convex sets, by projecting onto $C_1$, then $C_2$, ..., then $C_k$, and then repeating the cycle of $k$ projections. (This is called the sequential or cyclic projection algorithm, instead of alternating projection.) We'll describe several other extensions below.

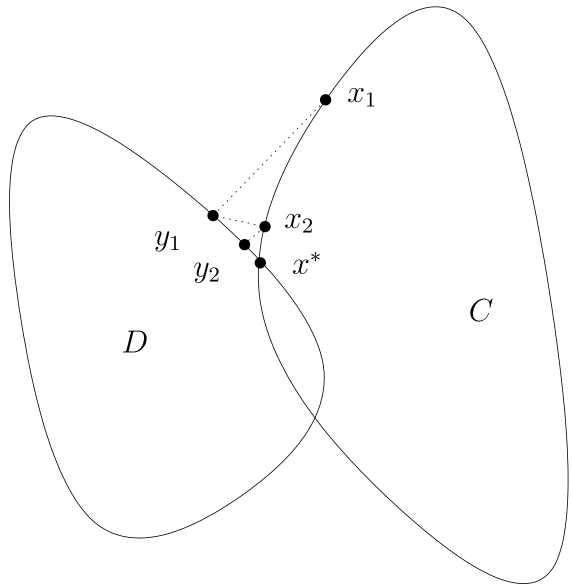


**Figure 1:** First few iterations of alternating projection algorithm. Both sequences are converging to the point $x^* \in C \cap D$.

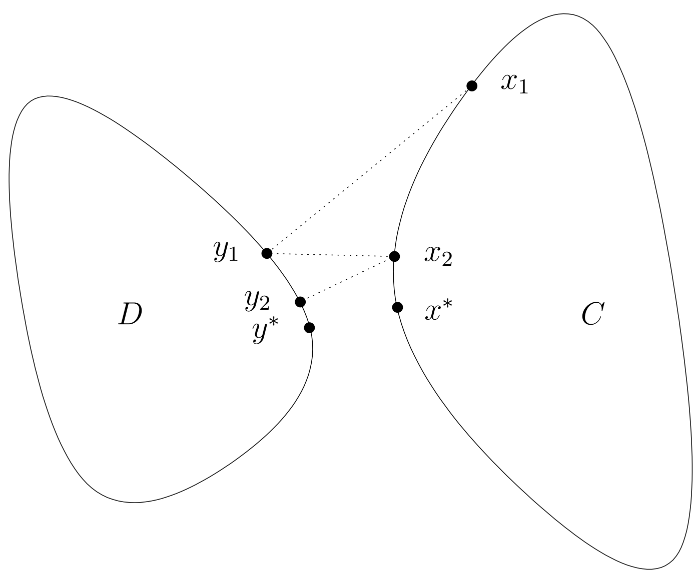


**Figure 2:** First few iterations of alternating projection algorithm, for a case in which $C \cap D = \emptyset$. The sequence $x_k$ is converging to $x^* \in C$, and the sequence $y_k$ is converging to $y^* \in D$, where $\|x^* - y^*\|_2 = \mathbf{dist}(C, D)$.

## 2 Convergence proof

We'll prove convergence of the alternating projections method for the case $C \cap D \neq \emptyset$. The proof of convergence for the case when $C \cap D = \emptyset$ is similar (and can be found in Cheney and Goldstein, for example). See also Bauschke and Borwein [BB96], Bregman [Bre67], and Akgul [Akg84] for related material.

Let $\bar{x}$ be any point in the intersection $C \cap D$. We claim that each projection brings the point closer to $\bar{x}$. To see this, we first observe that since $y_k$ is the projection of $x_k$ onto $D$, we have

$$D \subseteq \{z \mid (x_k - y_k)^T (z - y_k) \leq 0\}.$$

In other words, the halfspace passing through $y_k$, with outward normal $x_k - y_k$, contains $D$. This follows from the optimality conditions for Euclidean projection, and is very easy to show directly in any case: if any point of $D$ were on the other side of the hyperplane, a small step from $y_k$ towards the point would give a point in $D$ that is closer to $x_k$ than $y_k$, which is impossible.

Now we note that

$$\begin{array}{rcl} \|x_k - \bar{x}\|^2 & = & \|x_k - y_k + y_k - \bar{x}\|^2 \\ & = & \|x_k - y_k\|^2 + \|y_k - \bar{x}\|^2 + 2(x_k - y_k)^T (y_k - \bar{x}) \\ & \geq & \|x_k - y_k\|^2 + \|y_k - \bar{x}\|^2 \end{array}$$

using our observation above. Thus we have

$$\|y_k - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|y_k - x_k\|^2. \qquad (1)$$

This shows that $y_k$ is closer to $\bar{x}$ than $x_k$ is. In a similar way we can show that

$$\|x_{k+1} - \bar{x}\|^2 \leq \|y_k - \bar{x}\|^2 - \|x_{k+1} - y_k\|^2, \qquad (2)$$

*i.e.*, $x_{k+1}$ is closer to $\bar{x}$ than $y_k$ is.

We can make several conclusions. First, all the iterates are no farther from $\bar{x}$ than $x_0$:

$$\|x_k - \bar{x}\| \leq \|x_0 - \bar{x}\|, \qquad \|y_k - \bar{x}\| \leq \|x_0 - \bar{x}\|, \qquad k = 1, 2, \ldots$$

In particular, we conclude that the sequences $x_k$ and $y_k$ are bounded. Therefore the sequence $x_k$ has an accumulation point $x^*$. Since $C$ is closed, and $x_k \in C$, we have $x^* \in C$. We are going to show that $x^* \in D$, and that the sequences $x_k$ and $y_k$ both converge to $x^*$.

From (1) and (2), we find that the sequence

$$\|x_0 - \bar{x}\|, \ \|y_0 - \bar{x}\|, \ \|x_1 - \bar{x}\|, \ \|y_1 - \bar{x}\|, \ \ldots$$

is decreasing, and so converges. We then conclude from (1) and (2) that $\|y_k - x_k\|$ and $\|x_{k+1} - y_k\|$ must converge to zero.

A subsequence of $x_k$ converges to $x^*$. From

$$\mathbf{dist}(x_k, D) = \mathbf{dist}(x_k, y_k) \to 0$$

and closedness of $D$, we conclude that $x^* \in D$. Thus, $x^* \in C \cap D$.

Since $x^*$ is in the intersection, we can take $\bar{x} = x^*$ above (since $\bar{x}$ was any point in the intersection) to find that the distance of both $x_k$ and $y_k$ to $x^*$ is decreasing. Since a subsequence converges to zero, we conclude that $\|x_k - x^*\|$ and $\|y_k - x^*\|$ both converge to zero.

# 3 Extensions and variations

One simple variation starts with $x_0 \in C$ and $y_0 \in C$. We then find the average, $z_0 = (x_0 + y_0)/2$, and set $x_1 = P_C(z_0)$, $y_1 = P_D(z_0)$. We then repeat. In this algorithm, we average the current pair of points (one in $C$, one in $D$), and then project onto $C$ and $D$ respectively.

In fact, this is the same as applying the alternating projection algorithm in the product space, using the sets

$$C \times D, \qquad \{(u, v) \in \mathbf{R}^n \times \mathbf{R}^n \mid u = v\}.$$

It is also possible to use *under projection* or *over projection*. We set, for example, $y_k = \theta P_D(x_k) + (1-\theta)x_k$, where $\theta \in (0, 1)$ for under projection, and $\theta \in (1, 2)$ for over projection. In under projection, we step only the fraction $\theta$ from the current point to $D$; in over projection we move farther than the projection. Over projection is sometimes used to find a point in the intersection in a finite number of steps (assuming the intersection has nonempty interior). In both cases, the important part is that each step brings the point closer to $\bar{x}$, for any point $\bar{x} \in C \cap D$.

When sequential projection is done on more than two sets, the order does not have to be cyclic. We can project the current point onto *any* of the sets the point is not in, provided we infinitely often project onto any set the current point is not in.

# 4 Example: SDP feasibility

As an example we consider the problem of finding $X \in \mathbf{S}^n$ that satisfies

$$X \succeq 0, \qquad \mathbf{Tr}(A_i X) = b_i, \quad i = 1, \ldots, m,$$

where $A_i \in \mathbf{S}^n$. Here we take $C$ to be the positive semidefinite cone $\mathbf{S}^n_+$, and we take $D$ to be the affine set in $\mathbf{S}^n$ defined by the linear equalities. The Euclidean norm here is the Frobenius norm.

The projection of iterate $Y_k$ onto $C$ can be found from the eigenvalue decomposition $Y_k = \sum_{i=1}^n \lambda_i q_i q_i^T$ (see [BV03, §8.1.1]):

$$P_C(Y_k) = \sum_{i=1}^n \max\{0, \lambda_i\} q_i q_i^T.$$

The projection of iterate $X_k$ onto the affine set is also easy to work out:

$$P_D(X_k) = X_k - \sum_{i=1}^m u_i A_i \tag{3}$$

where $u_i$ are found from the normal equations,

$$Gu = (\mathbf{Tr}(A_1 X_k) - b_1, \ldots, \mathbf{Tr}(A_m X_k) - b_m), \qquad G_{ij} = \mathbf{Tr}(A_i A_j). \tag{4}$$

Alternating projections converges to a point in the intersection, if it is nonempty; otherwise it converges to the positive semidefinite matrix, and the symmetric matrix satisfying the linear equalities, that are of minimum Frobenius distance.

Now we consider a numerical example, which is a *positive semidefinite matrix completion problem* [BV03, exer. 4.47]. We are given a matrix in $\mathbf{S}^n$ with some of its entries (including all of its diagonal entries) fixed, and the others to be found. The goal is to find values for the other entries so that the (completed) matrix is positive semidefinite. For this problem, orthogonal projection onto the equality constraints is extremely easy: we simply take the matrix $X$, and set its fixed entries back to the fixed values given. Thus, we alternate between eigenvalue decomposition and truncation, and re-setting the fixed entries back to their required values.

As a specific example we consider

$$X = \begin{bmatrix} 4 & 3 & ? & 2 \\ 3 & 4 & 3 & ? \\ ? & 3 & 4 & 3 \\ 2 & ? & 3 & 4 \end{bmatrix},$$

where the question marks denote the entries to be determined. We initialize with $Y_0 = X$, taking the unknown entries as 0.

To track convergence of the algorithm, we plot $d_k = \|X_k - Y_{k-1}\|_F$, which is the squareroot of the sum of the squares of the negative eigenvalues of $Y_k$, and also the distance between $Y_k$ and the positive semidefinite cone. We also plot $\tilde{d}_k = \|Y_k - X_k\|_F$, which is the squareroot of the sum of the squares of the adjustments made to the fixed entries of $X_k$, and also the distance between $X_k$ and the affine set defined by the linear equalities. These are plotted in figure 3. In this example, both unknown entries converge quickly to the same value, 1.5858 (rounded), which yields a positive semidefinite completion of the original matrix. The plots show that convergence is linear.

# 5 Example: Relaxation method for linear inequalities

As another simple application, suppose we want to find a point in the polyhedron

$$\mathcal{P} = \{x \mid a_i^T x \le b_i,\ i = 1, \ldots, m\},$$

assuming it is nonempty. (This can of course be done using linear programming.) We'll use alternating projections onto the $m$ halfspaces $a_i^T x \le b_i$ to do this.

By scaling we can assume, without loss of generality, that $\|a_i\| = 1$. The projection of a point $z$ onto the halfspace $\mathcal{H}_i$ defined by $a_i^T x \le b_i$ is then given by

$$P_i(z) = \begin{cases} z & a_i^T z \le b_i \\ z - (a_i^T z - b_i) a_i & a_i^T z > b_i. \end{cases}$$

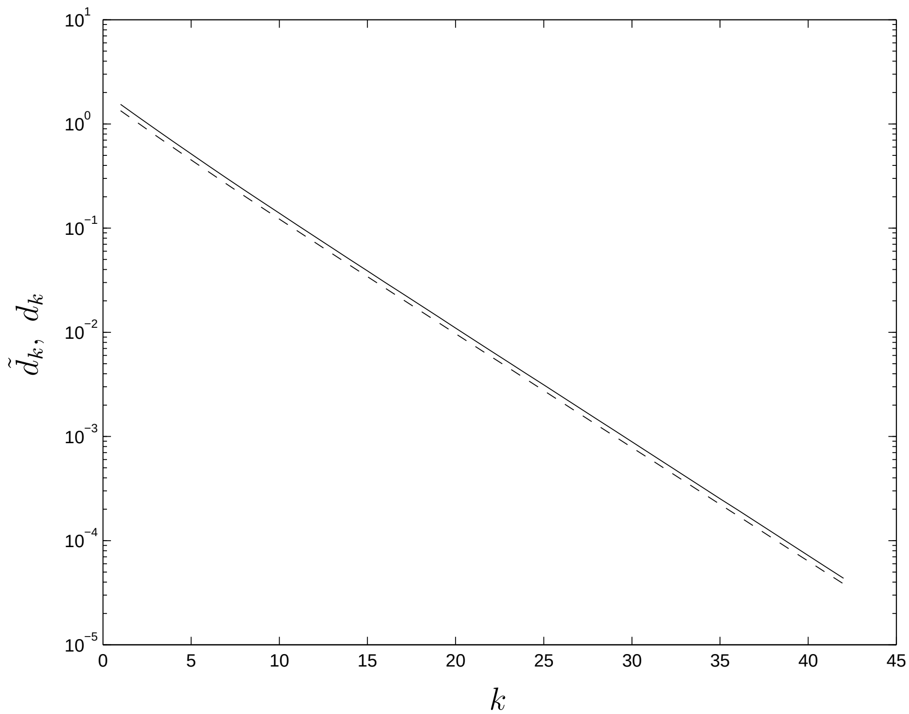


**Figure 3:** Convergence of alterating projections for a matrix completion problem. $d_k$ (solid line) gives the distance from $Y_{k-1}$ (which satisfies the equality constraints) to the positive semidefinite cone; $\tilde{d}_k$ (dashed) gives the distance from $X_k$ (which is positive semdefinite) to the affine set defined by the equality constraints.

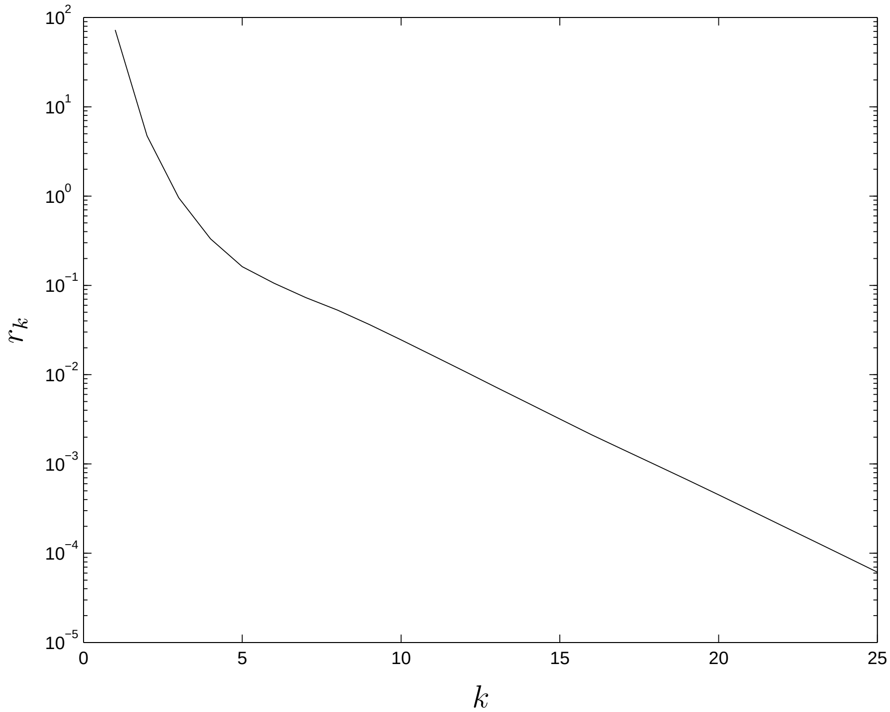


**Figure 4:** Maximum constraint violation versus iteration for relaxation method for solving linear inequalities, for a problem with $n = 100$ variables and $m = 1000$ inequalities.

By cycling through these projections, we generate a sequence that converges to a point in $\mathcal{P}$. This very simple algorithm for solving a set of linear inequalities is called the *relaxation method for linear inequalities* (see [Agm54, Ere65]). It was used, for example, to find a separating hyperplane between two sets of points. In this context it was dubbed the *perceptron algorithm* (see, *e.g.*, [WW96]).

To demonstrate convergence, we find a feasible point in a polyhedron in $\mathbf{R}^{100}$ defined by $m = 1000$ randomly chosen linear inequalities, $a_i^T x \leq b_i$, $i = 1, \ldots, m$, where $\|a_i\| = 1$. A starting point not in the polyhedron is selected. To show convergence, we plot the maximum constraint violation,

$$r_k = \max_{i=1}^{m} \max\{0, a_i^T x_k - b_i\},$$

versus iteration number, in figure 4.

# 6 Example: Row and column sum bounds

We consider the problem of finding a matrix $X \in \mathbf{R}^{m \times n}$ whose row sums and columns sums must lie in specified ranges, *i.e.*,

$$\sum_{j=1}^{n} X_{ij} \in [\underline{r}_i, \overline{r}_i], \quad i = 1, \ldots, m, \qquad \sum_{i=1}^{m} X_{ij} \in [\underline{c}_j, \overline{c}_j], \quad j = 1, \ldots, n.$$

Here $\underline{r}_i$ and $\overline{r}_i$ are given lower and upper bounds for the sum of the $i$th row, and $\underline{c}_j$ and $\overline{c}_j$ are given lower and upper bounds for the sum of the $j$th column. We can, of course, solve this feasibility problem using linear programming.

Alternating projections for this problem is very simple. To project a matrix onto the row sum range set, we can project each row separately. To project a vector $u$ onto the set $\{z \mid \alpha \leq \mathbf{1}^T z \leq \beta\}$ is easy: the projection is

$$P(z) = \begin{cases} z, & \alpha \leq \mathbf{1}^T z \leq \beta \\ z - ((\mathbf{1}^T z - \beta)/n)\mathbf{1}, & \mathbf{1}^T z > \beta \\ z - ((\alpha - \mathbf{1}^T z)/n)\mathbf{1}, & \mathbf{1}^T z < \alpha. \end{cases}$$

Alternating projections proceeds by alternately projecting the rows onto their required ranges (which can be done in parallel), and projecting the columns onto their required ranges (which also can be done in parallel).

# References

[Agm54] S. Agmon. The relaxation method for linear inequalities. *Canadian J. Math.*, 6:382–392, 1954.

[Akg84] M. Akgül. *Topics in Relaxation and Ellipsoidal Methods*, volume 97 of *Research Notes in Mathematics*. Pitman, 1984.

[BB96] H. Bauschke and J. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, 1996.

[Bre67] L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Math. and Math. Physics*, 7(3):200–217, 1967.

[BV03] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

[CG59] W. Cheney and A. Goldstein. Proximity maps for convex sets. *Proceedings of the AMS*, 10:448–450, 1959.

[Ere65] I. Eremin. The relaxation method of solving systems of inequalities with convex functions on the left sides. *Soviet Math. Dokl.*, 6:219–222, 1965.

[WW96] B. Widrow and E. Wallach. *Adaptive Inverse Control*. Prentice-Hall, 1996.