Alternating Projections

Original Author: Henry Pfister

1 Alternating Projection Algorithm

The alternating projection algorithm is a method for computing a point in the intersection of convex sets, using a sequence of projections onto these sets. It readily applies to projections onto subspaces because a subspace is a convex set. In general, this algorithm can be slow, yet it can be useful, especially when there exists an efficient method for carrying out the projections.

1.1 What is the effect of alternating between two orthogonal projections?

Suppose P_U and P_W are orthogonal projections onto closed subspaces U and W of a Hilbert space V. For an arbitrary $\underline{v}_0 \in V$, what is the behavior of the alternating projection

$$\underline{v}_{n+1} = \begin{cases} P_U \underline{v}_n & \text{if } n \text{ is even} \\ P_W \underline{v}_n & \text{if } n \text{ is odd.} \end{cases} \tag{1}$$

Since $P_U \underline{v} = \underline{v}$ (resp. $P_W \underline{v} = \underline{v}$) if and only if $\underline{v} \in U$ (resp. $\underline{v} \in W$), it is easy to see that any vector $\underline{v} \in U \cap W$ is a fixed point of this recursion. Letting $P_{U\cap W}$ denote the orthogonal projection onto $U \cap W$, one might guess that $\underline{v}_n$ converges to $P_{U\cap W}\underline{v}_0$ and indeed it does.

1.2 Can one use alternating projection to solve a system of linear equations?

Let $A \in \mathbb{R}^{m\times n}$ and $\underline{b} \in \mathbb{R}^m$ define a set of m linear equations in n variables with at least one solution. The goal is to use alternating projections to find a solution $\underline{x}^*$ such that $A\underline{x}^* = \underline{b}$. If $\underline{b} = \underline{0}$, then the set of solutions is a subspace equal to the null space of A,

$$\mathcal{N}(A) = \{\underline{x} \in \mathbb{R}^n \mid A\underline{x} = \underline{0}\} = \bigcap_{i=1}^{m} \left\{ \underline{x} \in \mathbb{R}^n \mid \textstyle\sum_{j=0}^{n} a_{i,j} x_j = 0 \right\}.$$

In this case, the result follows easily because $\mathcal{N}(A)$ equals the intersection of m subspaces of dimension $n-1$. What happens when $\underline{b} \neq \underline{0}$ or when no such solution exists?

1.3 Can alternating projection bound the value of a convex optimization?

Let $A \subseteq V$ be a closed convex set of a Hilbert space V. The projection of $\underline{v} \in V$ onto A is defined by

$$P_A(\underline{v}) \triangleq \arg\min_{\underline{u}\in A} \|\underline{u} - \underline{v}\|,$$

where the existence and uniqueness of the minimizer has been verified in the course notes. The term projection is overloaded here because this operation includes standard orthogonal projection (to a closed subspace) as a special case. Similar to orthogonal projections, alternating between projections onto convex sets provides a simple way to find a a point in their intersection.

For convex functions $f_i : V \to \mathbb{R}$ where $i = 0, 1, \ldots, m$, consider the convex optimization

$$\min_{\underline{x} \in V} f_0(\underline{x}) \text{ subject to } f_i(\underline{x}) \le b_i \quad i = 1, \ldots, m.$$

If $\underline{x} \in V$ satisfies the constraints and $f_0(\underline{x}) \le b_0$, then there is an $\underline{x} \in W$ where

$$W \triangleq \bigcap_{i=0}^{m} \{\underline{v} \in V \mid f_i(\underline{v}) \le b_i\}.$$

To test this hypothesis, one can apply the alternating projection algorithm to try and find a point in W. If the iteration converges, then the convex optimization has value at most b_0. Otherwise, the algorithm cycles and $W = \emptyset$.

1.4 Can one use alternating projections to train a linear classifier?

Let $A \in \mathbb{R}^{m \times n}$ and $\underline{b} \in \{-1, 1\}^m$ define a pattern classification problem where the i-th row of A is a sample vector with class label b_i. For a sample vector $\underline{v} \in \mathbb{R}^n$, a linear classifier with weight vector $\underline{x}$ uses the decision rule

$$\sum_{i=1}^{n} v_i x_i \gtrless 0.$$

If there is a weight vector that correctly classifies all training samples, then the training set is called *linearly separable*. In this case, the set of weight vectors (up to a scale factor) that separates the two classes is given by

$$\mathcal{W} = \left\{\underline{x} \in \mathbb{R}^n \mid b_i \textstyle\sum_{j=0}^{n} a_{ij} x_j \ge 1\right\} = \bigcap_{i=1}^{m} \left\{\underline{x} \in \mathbb{R}^n \mid b_i \textstyle\sum_{j=0}^{n} a_{i,j} x_j \ge 1\right\}.$$

Notice that $\mathcal{W}$ equals the intersection of m half-spaces. One can apply alternating projection onto these half-spaces to find a weight vector in $\mathcal{W}$. The performance can be quite reasonable even if the training set is not exactly separable.

2 A Closer look at Alternating Projections

Alternating projection is a method of finding a point in the intersection of multiple convex sets by sequentially projecting onto each of the sets. If the sets are all affine shifts of subspaces, then the process converges to the the orthogonal projection of the initial vector onto the intersection of the sets. For more complex sets, the algorithm is only guaranteed to produce a vector that lies in the intersection. But, this vector may not be the closest to the initial vector. There is, however, a simple generalization of the algorithm by Dykstra that computes the orthogonal projection onto the intersection of general convex sets.

It is worth noting that, while the idea of alternating projection provides algorithms that are simple and easy to understand, it often fails to provide the most computationally efficient solution.

2.1 Proof of Convergence for Two Subspaces

Theorem 2.1. *The sequence $\underline{v}_n$ converges to $P_{U\cap W}(\underline{v}_0)$, its projection onto $U \cap W$.*

Proof (for the case where $(U\cap W)^{\perp}$ is finite dimensional). Let U and W be two closed subspaces of a Hilbert space V. For any $\underline{v}_0 \in V$, let the sequence $\underline{v}_1, \underline{v}_2, \ldots$ be defined by (1). Then, we have $\underline{v}_i \in \mathrm{span}(U, W)$ for $i \geq 1$. Thus, it suffices to assume that $V = \mathrm{span}(U, W)$. Using the unique decomposition

$$\underline{v}_0 = P_{U\cap W}\underline{v}_0 + (I - P_{U\cap W})\,\underline{v}_0,$$

we observe that

$$\underline{v}_1 = P_U\underline{v}_0 = P_U P_{U\cap W}\underline{v}_0 + P_U\,(I - P_{U\cap W})\,\underline{v}_0 = P_{U\cap W}\underline{v}_0 + (I - P_{U\cap W})\,P_U\underline{v}_0$$

because $P_U P_{U\cap W} = P_{U\cap W} P_U$. Similarly, we have

$$\underline{v}_2 = P_W\underline{v}_1 = P_W P_{U\cap W}\underline{v}_0 + P_W\,(I - P_{U\cap W})\,P_U\underline{v}_0 = P_{U\cap W}\underline{v}_0 + (I - P_{U\cap W})\,P_W\underline{v}_1$$

because $P_W P_{U\cap W} = P_{U\cap W} P_W$. Defining the error as

$$\underline{z}_n \triangleq (I - P_{U\cap W})\,\underline{v}_n = \underline{v}_n - P_{U\cap W}\underline{v}_0,$$

we see that

$$\begin{aligned}
\underline{z}_1 &= P_U\,(I - P_{U\cap W})\,\underline{v}_0 = P_U\underline{z}_0 = (I - P_{U\cap W})\,P_U\underline{v}_0 \\
\underline{z}_2 &= P_W\,(I - P_{U\cap W})\,\underline{v}_1 = P_W\underline{z}_1 = (I - P_{U\cap W})\,P_W\underline{v}_1.
\end{aligned}$$

This sequence continues by induction and shows both that $\underline{z}_n \in (U\cap W)^{\perp}$ for all n and that

$$\underline{z}_{n+1} = \begin{cases} P_U\underline{z}_n & \text{if } n \text{ is even} \\ P_W\underline{z}_n & \text{if } n \text{ is odd.} \end{cases}$$

satisfies the same recursion as $\underline{v}_n$ starting from $\underline{z}_0 = (I - P_{U\cap W})\,\underline{v}_0$.

To show that $\underline{v}_n \to P_{U\cap W}\underline{v}_0$, it is sufficient (based on the previous decomposition) to show that $\underline{z}_n \to \underline{0}$. Since the recursion implies that $\left\|\underline{z}_{n+1}\right\| \leq \|\underline{z}_n\|$ is decreasing, we know that $\|\underline{z}_n\| \to d$ for some $d \geq 0$. If $(U\cap W)^{\perp}$ is finite dimensional, then any closed subset of $(U\cap W)^{\perp}$ with bounded norm is compact and, as such, there must be a subsequence $\underline{z}_{n_i}$ that converges. Let $\underline{z}_\infty$ denote the limit of this subsequence and notice that $\underline{z}_n \in (U\cap W)^{\perp}$ for all n implies $\underline{z}_\infty \in (U\cap W)^{\perp}$ because $(U\cap W)^{\perp}$ is closed. Using this subsequence, the continuity of the norm implies that $\lim_{i\to\infty}\left\|\underline{z}_{n_i}\right\| = \|\underline{z}_\infty\| = d$ and the continuity of the recursion implies that $P_W P_U\underline{z}_\infty = \underline{z}_\infty$. But, the last statement can only hold if $\underline{z}_\infty \in U\cap W$. Therefore, $\underline{z}_\infty \in U\cap W$ and $\underline{z}_\infty \in (U\cap W)$ together imply that $\underline{z}_\infty = \underline{0}$. Hence, we see that $d = 0$ and $\underline{z}_n \to \underline{0}$. □

This proof can be extended in a straightforward manner to the case where a finite number of orthogonal projections are applied sequentially. A more technical proof, which avoids the assumption that $(U\cap W)^{\perp}$ is finite dimensional, is presented in Appendix B.

Theorem 2.2. *Let $W_1, \ldots, W_m$ be closed subspaces of a Hilbert space and define $W_0 = \cap_{i=1}^m W_i$. Then, for any $\underline{v}_0 \in V$, the recursion*

$$\underline{v}_{n+1} = P_{W_{(n \bmod m)+1}} \underline{v}_n$$

generates a sequence $\underline{v}_n$ that converges to the orthogonal projection $P_{W_0}\underline{v}_0$.

Exercise 1. *Let U and W be subspaces of $\mathbb{R}^5$ that are spanned, respectively, by the columns of the matrices A and B (shown below). Write a function `altproj(A,B,v0,n)` that performs $2n$ steps of alternating projection onto U and W starting from $\underline{v}_0$. This function should return the final vector $\underline{v}_{2n}$ and a vector of error values $g_{2k} = \|\underline{v}_{2k} - P_{U\cap W}(\underline{v}_0)\|_\infty$ for $k = 1, 2, \ldots, n$. Use this function to estimate the orthogonal projection of $\underline{v}_0$ (shown below) onto $U \cap W$. How large should n be chosen so that the projection is correct to 4 decimal places (i.e., $g_{2n} \leq 0.0001$)?*

$$A = \begin{bmatrix} 3 & 2 & 3 \\ 1 & 5 & 7 \\ 3 & 11 & 13 \\ 1 & 17 & 19 \\ 5 & 23 & 29 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 1 & 2.5 \\ 2 & 0 & 6 \\ 2 & 1 & 12 \\ 2 & 0 & 18 \\ 6 & -3 & 26 \end{bmatrix} \qquad \underline{v}_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

To find the intersection of U and W, we note that the following code snippets returns matrices whose columns span $U \cap W$:

```
% Matlab
basis_UintW = [A B]*null([A -B],'r');

# Python
import numpy as np
from scipy.linalg import svd
def null_space(A, rcond=None):
    u, s, vh = svd(A, full_matrices=True)
    M, N = u.shape[0], vh.shape[1]
    if rcond is None:
        rcond = np.finfo(s.dtype).eps * max(M, N)
    tol = np.amax(s) * rcond
    num = np.sum(s > tol, dtype=int)
    Q = vh[num:,:].T.conj()
    return Q

basis_UintW = np.hstack([A, B]) @ null_space(np.hstack([A, -B]))
```

3 Kaczmarz's Algorithm

Kaczmarz's algorithm is a method of solving a system of linear equations based on iteratively projecting a candidate vector onto each of the linear equality constraints. For a matrix $A \in \mathbb{R}^{m \times n}$

and vector $\underline{b} \in \mathbb{R}^m$, the algorithm starts from $\underline{v}_0 = \underline{0}$ and recursively defines $\underline{v}_{i+1}$ to be the projection of $\underline{v}_i$ onto the set

$$W_i = \left\{ \underline{v} \in \mathbb{R}^n \,\middle|\, \sum_{k=1}^{n} a_{\sigma(i),k} v_k = b_{\sigma(i)} \right\},$$

where $\sigma(i) = (i \bmod m) + 1$. Using (6), we can write this explicitly as

$$\underline{v}_{i+1} = \underline{v}_i - \frac{\langle \underline{v}_i | \underline{a}_{\sigma(i)} \rangle - b_{\sigma(i)}}{\|\underline{a}_{\sigma(i)}\|^2} \underline{a}_{\sigma(i)}, \tag{2}$$

where $\underline{a}_j$ is the j-th row of the matrix A.

Theorem 3.1. *If the linear system is consistent (e.g., there exists $\underline{x}^* \in \mathbb{R}^n$ such that $A\underline{x}^* = \underline{b}$), then the sequence defined by* (2) *converges to the minimum norm solution of the linear system.*

Proof. To see this, we will analyze the algorithm in a shifted coordinate system. Let $\underline{x}_i = \underline{v}_i - \underline{x}^*$ so that $\underline{v}_i = \underline{x}_i + \underline{x}^*$. Then, the update computes

$$\underline{x}_{i+1} = \underline{v}_{i+1} - \underline{x}^* = (\underline{x}^* + \underline{x}_i) - \frac{\langle \underline{x}^* + \underline{x}_i | \underline{a}_{\sigma(i)} \rangle - b_{\sigma(i)}}{\|\underline{a}_{\sigma(i)}\|^2} \underline{a}_{\sigma(i)} - \underline{x}^* = \underline{x}_i - \frac{\langle \underline{x}_i | \underline{a}_{\sigma(i)} \rangle}{\|\underline{a}_{\sigma(i)}\|^2} \underline{a}_{\sigma(i)},$$

which equals the orthogonal projection of $\underline{x}_i$ onto the subspace given by $\left\{ \underline{x} \in \mathbb{C}^n \,\middle|\, \langle \underline{x}_i | \underline{a}_{\sigma(i)} \rangle = 0 \right\}$. The initialization $\underline{v}_0 = \underline{0}$ implies that $\underline{x}_0 = -\underline{x}^*$ and, from Theorem 2.2, we know that the sequence $\underline{x}_i$ must converge to

$$P_{\{\underline{x}: A\underline{x}=0\}}(-\underline{x}^*) + \underline{x}^*.$$

But, applying (5), we see that

$$P_{\{\underline{x}: A\underline{x}=0\}}(-\underline{x}^*) + \underline{x}^* = P_{\{\underline{x}: A\underline{x}=0\}+\underline{x}^*}(-\underline{x}^* + \underline{x}^*) + \underline{x}^* - \underline{x}^* = P_{\{\underline{x}: A\underline{x}=\underline{b}\}}(\underline{0}).$$

Therefore, Kaczmarz's algorithm converges to $P_{\{\underline{x}: A\underline{x}=\underline{b}\}}(\underline{0})$, which is the minimum norm solution of $A\underline{x} = \underline{b}$. $\square$

Remark 3.2. *Recently, a number of researchers have analyzed the convergence of Kaczmarz's algorithm for the case where, for each i, $\sigma(i)$ is chosen to be a uniform random integer in $\{1, 2, \ldots, m\}$ [1]. Also, while Kaczmarz's algorithm does not converge if the linear system is inconsistent, there is an extended version that converges to the least-squares solution in this case [2].*

Exercise 2. *Write a function `kaczmarz(A,b,I)` that performs the Kaczmarz algorithm for matrix A and right-hand side $\underline{b}$ using `I` full passes through the rows (e.g., one full pass equals m steps). It should return a matrix `X` with `I` columns corresponding to the vector after each full pass and a vector containing the error $g_k = \|A\underline{v}_{km} - \underline{b}\|_\infty$ for $k = 1, 2, \ldots, \mathtt{I}$. Use this function to estimate the minimum-norm solution of linear system $A\underline{x} = \underline{b}$ for*

$$A = \begin{bmatrix} 2 & 5 & 11 & 17 & 23 \\ 3 & 7 & 13 & 19 & 29 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} 228 \\ 277 \end{bmatrix}.$$

For $\mathtt{I} = 500$, plot the error g_k on a log scale for $k = 1, 2, \ldots, \mathtt{I}$.

Exercise 3. *Repeat the experiment with* `I` $= 100$ *for a random system where* A *is a* 500×1000 *standard Gaussian matrix,* $\underline{b}$ *is a* 500×1 *vector defined by* $\underline{b} = A\underline{x}$ *where* $\underline{x}$ *is a* 1000×1 *standard Gaussian vector. Compare the iterative solution with the true minimum-norm solution* $\hat{x} = A^H(AA^H)^{-1}\underline{b}$.

```
# Python
from numpy.random import randn
A = randn(500, 1000)
b = A @ randn(1000)
```

4 Bounding the Value of a Convex Optimization

The value of a convex optimization problem can also be bounded by determining whether or not the intersection of a collection of convex sets is empty or not. The alternating projection algorithm can be used to find a point in the intersection of all the sets but it is not guaranteed to find the closest point in the intersection. Let $C_1, C_2, \ldots, C_m$ be closed convex subsets of a Hilbert space V. Then, starting from any $\underline{x}_0 \in V$, the alternating projection algorithm computes

$$\underline{x}_{i+1} = (1-s)\underline{x}_i + s\, P_{C_{\sigma(i)}}(\underline{x}_i), \tag{3}$$

where $\sigma(i) = (i \bmod m) + 1$ and $s \in (0, 1]$ is step-size parameter.

Theorem 4.1 (Bregman)**.** *For some* $\underline{x} \in \cap_{i=1}^{m} C_i$*, the sequence generated by the above iteration with* $s = 1$ *satisfies*

$$\langle \underline{x}_i - \underline{x} | \underline{u} \rangle \to 0$$

for all $\underline{u} \in V$*. This type of convergence is known as weak convergence. If* V *is finite-dimensional, then weak convergence implies (strong) convergence and* $\underline{x}_i \to \underline{x}$*.*

Consider the linear program

$$\min \underline{c}^T \underline{x} \text{ subject to } A\underline{x} \geq \underline{b},\ \underline{x} \geq 0, \tag{4}$$

where $\underline{c} \in \mathbb{R}^n$, $\underline{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $\underline{b} \in \mathbb{R}^m$. For a concrete example, we will choose

$$\underline{c} = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 0 & 2 \\ -7 & 4 & -6 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}.$$

Let p^* denote the optimum value of this program. Then, $p^* \leq 0$ is satisfied if and only if there is a non-negative vector $\underline{x} = (x_1, x_2, x_3)^T$ satisfying

$$\begin{aligned} 2x_1 - x_2 + x_3 &\geq -1 \\ x_1 + 2x_3 &\geq 2 \\ -7x_1 + 4x_2 - 6x_3 &\geq 1 \\ -3x_1 + x_2 - 2x_3 &\geq 0, \end{aligned}$$

where the last inequality restricts the value of the program to be at most 0. One can find the optimum value `p` and an optimizer `x` with the commands:

```
# Python
from scipy.optimize import linprog
res = linprog(c, Aub=-A, bub=-b, bounds=[(0, None)] * c.size, method='inter
x, p = res.x, res.fun
```

Exercise 4. *Write a function* `x=lp_altproj(A,b,I,s)` *that uses* (3) *(starting from $\underline{x}_0 = \underline{0}$) to implement alternating projections onto half spaces (see* (7)*). The program should use* `I` *passes through the entire set of inequality constraints (with step size s) to find a non-negative vector $\underline{x}$ that satisfies $A\underline{x} \geq \underline{b}$. It should output the final vector $\underline{x}_{mI}$ and a vector containing the maximum feasibilty gap $g_k = \max_j \left[\underline{b} - A\underline{x}_{km}\right]_j$ for $k = 1, 2, \ldots,$* `I`*.*

Apply this program with $s = 1$ to the above set of 4 inequalities in 3 variables. Warning: don't forget to also project onto the half spaces defined by the non-negativity constraints $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Use this result to find a vector (e.g., by rounding) that satisfies all the inequalities. The goal of this problem is to satisfy $\underline{x} \geq \underline{0}$ and $A\underline{x} \geq \underline{b}$. How many full passes are required so that g_k is at most 0.0001?

Remark 4.2. *For $j = 1, 2, \ldots, n$, the projection of $\underline{x}$ onto the constraint $x_j \geq 0$ is given by $\underline{\tilde{x}}$ where $\tilde{x}_j = \max(0, x_j)$ and $\tilde{x}_k = x_k$ for $k \neq j$. One can handle these constraints either by appending them to A and $\underline{b}$ or by building this operation in after each projection step. For maximum flexibility, we ask you to include the projection minimum value (0 in this case) as an optional argument* `xmin` *with the value* `None` *indicating that no projection is required. Both approaches are guaranteed to converge though they may have different convergence rates.*

Exercise 5. *Use the function* `x=lp_altproj(A,b,I,1)` *to find a non-negative vector $\underline{x}$ that satisfies $A\underline{x} \geq \underline{b}$ for the "random" convex optimization problem defined by:*

```
# Python
import numpy as np
from numpy.random import randn
np.random.seed(0)
c = randn(1000)
A = np.vstack([-np.ones((1,1000)),randn(500,1000)])
b = np.concatenate([[-1000], A[1:] @ rand(1000)])
```

Then, modify A and $\underline{b}$ (by adding one row and one element) so that your function can be used to prove that the value of the convex optimization problem, in (4), is at most -1000. Try using `I` $= 1000$ passes through all 501 inequality constraints.

This type of iteration typically terminates with an "almost feasible" $\underline{x}$. To find a strictly feasible point, for some small $\epsilon > 0$ (e.g., try $\epsilon = 10^{-6}$), try running the same algorithm with the argument $\underline{b} + \epsilon$ and projecting onto strictly positive set $\{\underline{x} \in \mathbb{R}^{1000} \,|\, x_i \geq \epsilon,\, i \in [1000]\}$. Then, the resulting $\underline{x}$ will satisfy all constraints. The code below checks if all constrainst are satisfied.

```
# Python
import numpy as np
np.all(x>0)
np.all((A@x-b)>0)
```

5 Orthogonal Projection Onto the Intersection of Convex Sets

The alternating projection algorithm in Section 4 finds a point in the intersection of all the sets but it is not guaranteed to find the closest point in the intersection. Fortunately, there is a modification by Dykstra that rectifies this problem [3].

Let $C_1, C_2, \ldots, C_m$ be closed convex subsets of a Hilbert space V. Then, Dykstra's Algorithm computes the projection $P_{\cap_{i=1}^m C_i}(\underline{v}_0)$ via the iteration

$$\begin{aligned} \underline{v}_{i+1} &= P_{C_{\sigma(i)}}\left(\underline{v}_i - \underline{w}_{\sigma(i)}\right) \\ \underline{w}_{\sigma(i)} &= \underline{v}_{i+1} - \left(\underline{v}_i - \underline{w}_{\sigma(i)}\right), \end{aligned}$$

where $\underline{w}_1, \ldots, \underline{w}_m$ are initialized to $\underline{0}$.

Theorem 5.1 ([3])**.** *The sequence generated by the above iteration satisfies*

$$\lim_{i\to\infty} \underline{v}_i = P_{\cap_{i=1}^m C_i}(\underline{v}_0).$$

Exercise 6. *(optional) Let $V = \mathbb{R}^2$ and consider the orthogonal projection of $\underline{u} = (1, -2)$ onto the intersection of*

$$\begin{aligned} C_1 &= \{\underline{v} \in V \mid v_2 \geq 0\} \\ C_2 &= \left\{ \underline{v} \in V \,\middle|\, v_1^2 + \left(v_2 - \tfrac{\sqrt{3}}{2}\right)^2 \leq 1 \right\}. \end{aligned}$$

Draw a picture illustrating the alternating projections (without Dykstra's modification) defined by: $P_{C_2}(P_{C_1}(\underline{u}))$ and $P_{C_1}(P_{C_2}(\underline{u}))$. Does either give the desired result $P_{C_1\cap C_2}(\underline{u})$? Now, try Dykstra's algorithm for 10 iterations using both projection orders (i.e., C_1 first or C_2 first). Do these approach $P_{C_1\cap C_2}(\underline{u})$?

6 Conclusion

The goal of this note is to highlight the utility of alternating projection for understanding and solving problems. While it may not provide the most computationally efficient solution, it does lead to simple and geometrically interpretable algorithms that can be easily adapted to many problems.

A Projections onto Standard Sets

Let A be a closed convex subset of a Hilbert space V. Then, for all $\underline{v}, \underline{v}_0 \in V$, the projection onto V satisfies

$$\begin{aligned} P_{A+\underline{v}_0}(\underline{v} + \underline{v}_0) &= \arg\min_{\underline{u}\in A+\underline{v}_0} \|\underline{u} - \underline{v} - \underline{v}_0\| \\ &= \underline{v}_0 + \arg\min_{\underline{u}'\in A} \|(\underline{u}' + \underline{v}_0) - \underline{v} - \underline{v}_0\| \\ &= \underline{v}_0 + \arg\min_{\underline{u}'\in A} \|\underline{u}' - \underline{v}\| \\ &= \underline{v}_0 + P_A(\underline{v}). \end{aligned} \tag{5}$$

In words, this means that translating the set A and the vector $\underline{v}$ by the same vector $\underline{v}_0$ results in an output that is also translated by $\underline{v}_0$. This also leads to the following trick. If a projection is easy when the set is centered, then one can: (i) translate the problem so that the set is centered, (ii) project onto the centered set, and (iii) translate back.

A.1 Subspaces of Dimension 1, Linear Equalities, and Half Spaces

Using the best approximation theorem, it is easy to verify that the orthogonal projection of $\underline{v} \in V$ onto a one-dimensional subspace $W = \text{span}(\underline{w})$ is given by

$$P_W(\underline{v}) = \frac{\langle \underline{v}|\underline{w}\rangle}{\|\underline{w}\|^2}\underline{w}.$$

A closed subspace U with co-dimension one (e.g., if V has dimension n, then this is a subspace of dimension $n-1$) is a subset of V that satisfies a single linear equality of the form $\langle \underline{v}|\underline{w}\rangle = 0$. Thus, U can be seen as the orthogonal complement of a one-dimensional subspace (e.g., $U = W^{\perp}$) and we can write

$$P_U(\underline{v}) = P_{W^\perp}(\underline{v}) = \underline{v} - \frac{\langle \underline{v}|\underline{w}\rangle}{\|\underline{w}\|^2}\underline{w}.$$

Similarly, a linear equality such as $\langle \underline{v}|\underline{w}\rangle = c$ defines a shifted subspace $U + \underline{v}_0$ (where $\underline{v}_0$ is any vector in V satisfying $\langle \underline{v}_0|\underline{w}\rangle = c$) with co-dimension one because

$$\langle \underline{v}|\underline{w}\rangle = \langle \underline{u} + \underline{v}_0|\underline{w}\rangle = \langle \underline{u}|\underline{w}\rangle + \langle \underline{v}_0|\underline{w}\rangle = 0 + c = c.$$

Thus, we can project onto $U + \underline{v}_0$ by translating, projecting, and then translating back. This gives

$$P_{U+\underline{v}_0}(\underline{v}) = \left((\underline{v} - \underline{v}_0) - \frac{\langle \underline{v} - \underline{v}_0|\underline{w}\rangle}{\|\underline{w}\|^2}\underline{w}\right) + \underline{v}_0 = \underline{v} - \frac{\langle \underline{v}|\underline{w}\rangle - c}{\|\underline{w}\|^2}\underline{w}, \tag{6}$$

which does not depend on the choice of $\underline{v}_0$.

Finally, let H be the subset of $\underline{v} \in V$ satisfying the linear inequality $\langle \underline{v}|\underline{w}\rangle \geq c$. Then, H is a closed convex set known as a *half space*. For any $\underline{v} \in H$, we have $P_H(\underline{v}) = \underline{v}$ and, for any $\underline{v} \notin H$, we have $P_H(\underline{v}) = P_{U+\underline{v}_0}(\underline{v})$ because the closest point must achieve the inequality with equality. Putting these together, for any $\underline{v} \in H$, we find that

$$P_H(\underline{v}) = \begin{cases} \underline{v} & \text{if } \langle \underline{v}|\underline{w}\rangle \geq c \\ \underline{v} - \frac{\langle \underline{v}|\underline{w}\rangle - c}{\|\underline{w}\|^2}\underline{w} & \text{if } \langle \underline{v}|\underline{w}\rangle < c. \end{cases} \tag{7}$$

A.2 The Unit Ball

In section, we consider orthogonal projections onto convex bodies similar to the unit ball. Using (5), we now know that it is sufficient to consider convex bodies centered at $\underline{0}$. For a Hilbert space V over $\mathbb{R}$, the unit ball is defined to be

$$B \triangleq \{\underline{w} \in V \mid \|\underline{w}\| \leq 1\}.$$

By drawing a picture, it is easy to see that

$$P_B(\underline{v}) = \begin{cases} \underline{v} & \text{if } \|\underline{v}\| \le 1 \\ \frac{\underline{v}}{\|\underline{v}\|} & \text{if } \|\underline{v}\| > 1. \end{cases}$$

For $\|\underline{v}\| \le 1$, the statement is trivial. For $\|\underline{v}\| > 1$, it follows from the generalized orthogonality principle for projections onto convex sets and

$$\begin{aligned} \left\langle \underline{v} - \frac{\underline{v}}{\|\underline{v}\|} \,\middle|\, \underline{w} - \frac{\underline{v}}{\|\underline{v}\|} \right\rangle &= \langle \underline{v} \,|\, \underline{w} \rangle - \frac{1}{\|\underline{v}\|} \langle \underline{v} \,|\, \underline{w} \rangle - \|\underline{v}\| + 1 \\ &= \left(1 - \frac{1}{\|\underline{v}\|}\right) \langle \underline{v} \,|\, \underline{w} \rangle - \|\underline{v}\| + 1 \\ &\le \left(1 - \frac{1}{\|\underline{v}\|}\right) \|\underline{v}\| \, \|\underline{w}\| - \|\underline{v}\| + 1 \\ &\le 0 \end{aligned}$$

for all $\underline{w} \in B$, where the inequalities rely on $1 - 1/\|\underline{v}\| \ge 0$, $\langle \underline{v} \,|\, \underline{w} \rangle \le \|\underline{v}\| \, \|\underline{w}\|$, and $\|\underline{w}\| \le 1$.

For the scaled and translated unit ball, $aB + \underline{v}_0$, the formula becomes

$$P_{aB+\underline{v}_0}(\underline{v}) = \begin{cases} \underline{v} & \text{if } \|\underline{v} - \underline{v}_0\| \le a \\ \frac{a(\underline{v}-\underline{v}_0)}{\|\underline{v}-\underline{v}_0\|} + \underline{v}_0 & \text{if } \|\underline{v} - \underline{v}_0\| > a. \end{cases}$$

B General Proof of Subspace Alternating Projection Theorem

Earlier in this document, we presented an intuitive proof sketch of the alternating projection theorem under the assumption that $(U \cap W)^\perp$ is finite dimensional. Here, we present a shorter but more technical proof that does not require this assumption [4]. Both proofs can be extended in a straightforward manner to the case where a finite number of orthogonal projections are applied sequentially.

Proof of Theorem 2.1. For even n, Lemma B.1 shows that

$$\left\| (P_W P_U)^{n/2} (I - P_W P_U) \underline{v}_0 \right\| = \left\| \underline{v}_n - \underline{v}_{n+2} \right\|^2 \to 0$$

as $n \to \infty$ for all $\underline{v}_0 \in V$. This implies that $(P_W P_U)^{n/2} \underline{w} \to 0$ for all $\underline{w} \in \mathcal{R}(I - P_W P_U)$. Next, we observe that

$$\begin{aligned} \mathcal{R}(I - P_W P_U) &= \mathcal{N}\big((I - P_W P_U)^H\big)^\perp \\ &= \mathcal{N}(I - P_U P_W)^\perp \\ &= (U \cap W)^\perp, \end{aligned}$$

where the 3rd step holds because "$P_U P_W \underline{v} = \underline{v}$ if and only if $\underline{v} \in U \cap W$" implies that "$\underline{v} \in \mathcal{N}(I - P_U P_W)$ if and only if $\underline{v} \in U \cap W$". Applying this result separately to the two terms in $\underline{v}_0 = P_{U \cap W} \underline{v}_0 + (I - P_{U \cap W}) \underline{v}_0$, we see that the first term is preserved while the second term is driven to zero. Thus, we find that $\underline{v}_n \to P_{U \cap W} \underline{v}_0$ along the even n subsequence. Of course, convergence along any subsequence follows by noting that P_U is continuous. $\square$

Lemma B.1 (Kakutani)**.** *For all $n \geq 0$, the upper bound*

$$\left\|\underline{v}_{n+2} - \underline{v}_n\right\|^2 \leq 2\left(\left\|\underline{v}_n\right\|^2 - \left\|\underline{v}_{n+2}\right\|^2\right)$$

implies that $\left\|\underline{v}_{n+2} - \underline{v}_n\right\|^2 \to 0$ *as* $n \to \infty$*.*

Proof. We start by assuming n is even and writing

$$\begin{aligned}
\left\|\underline{v}_{n+2} - \underline{v}_n\right\|^2 &= \|P_W P_U \underline{v}_n - P_U \underline{v}_n + P_U \underline{v}_n - \underline{v}_n\| \\
&\overset{(a)}{\leq} \left(\|P_W P_U \underline{v}_n - P_U \underline{v}_n\| + \|P_U \underline{v}_n - \underline{v}_n\|\right)^2 \\
&\overset{(b)}{\leq} 2\left(\|P_W P_U \underline{v}_n - P_U \underline{v}_n\|^2 + \|P_U \underline{v}_n - \underline{v}_n\|^2\right) \\
&\overset{(c)}{=} 2\left(\|P_U \underline{v}_n\|^2 - \|P_W P_U \underline{v}_n\|^2 + \|\underline{v}_n\|^2 - \|P_U \underline{v}_n\|^2\right) \\
&\leq 2\left(\left\|\underline{v}_n\right\|^2 - \left\|\underline{v}_{n+2}\right\|^2\right),
\end{aligned}$$

where (a) follows from the triangle inequality, (b) holds because $(a+b)^2 \leq 2(a^2+b^2)$, and (c) follows from

$$\|P_U \underline{v}_n - \underline{v}_n\|^2 = \|\underline{v}_n\|^2 - \|P_U \underline{v}_n\|^2 .$$

The same argument works when n is odd by switching P_U and P_W. To see the convergence to 0, we note that $\|\underline{v}_n\|^2 \leq \left\|\underline{v}_{n+1}\right\|^2$ implies that $\|\underline{v}_n\|^2$ converges to a limit. Thus, $\|\underline{v}_n\|^2 - \left\|\underline{v}_{n+2}\right\|^2$ converges to 0. □

References

[1] T. Strohmer and R. Vershynin, "A randomized kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2009.

[2] S. Petra and C. Popa, "Single projection kaczmarz extended algorithms," *Numerical Algorithms*, vol. 73, no. 3, pp. 791–806, 2016.

[3] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in hilbert spaces," in *Advances in order restricted statistical inference*, pp. 28–47, Springer, 1986.

[4] A. Netyanun and D. C. Solmon, "Iterated products of projections in hilbert space," *The American Mathematical Monthly*, vol. 113, no. 7, pp. 644–648, 2006.