Supplement to the Briefing on Min-Norm-Point Algorithm, Process Explanation

Abstract: this part demonstrates and explains the ideas behind the algorithm mentioned in "The Minimum-Norm-Point Algorithm Submodular Function Minimization and Linear Programming" by Satoru Fujishige, Takumi Hayashi and Shigueo Isotani, published in September 2006.

Introduction

The algorithm is described below.

2.1. Description of the minimum-norm-point algorithm

Consider the n-dimensional Euclidean space $\mathbf{R}^n$. Suppose that we are given a finite set P of points p_i $(i \in I)$ in $\mathbf{R}^n$. The problem is to find the minimum-norm point x^* in the convex hull $\hat{P}$ of points p_i $(i \in I)$.

Wolfe's algorithm [15] is given as follows.

The Minimum-Norm-Point Algorithm

Input: A finite set P of points p_i $(i \in I)$ in $\mathbf{R}^n$.
Output: The minimum-norm point x^* in the convex hull $\hat{P}$ of the points p_i $(i \in I)$.

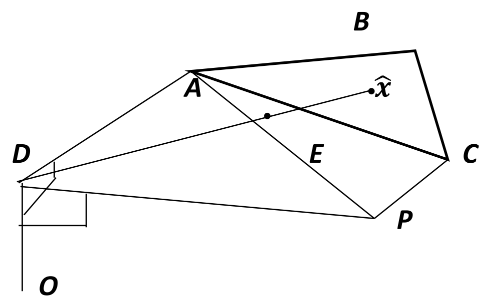
Step 1: Choose any point p in P and put $S := \{p\}$ and $\hat{x} := p$.

Step 2: Find a point $\hat{p}$ in P that minimizes the linear function $\langle \hat{x}, p \rangle = \sum_{k=1}^{n} \hat{x}(k)p(k)$ in $p \in P$. Put $S := S \cup \{\hat{p}\}$.
If $\langle \hat{x}, \hat{p} \rangle = \langle \hat{x}, \hat{x} \rangle$, then return $x^* = \hat{x}$ and halt;
else go to Step 3.

Step 3: Find the minimum-norm point y in the affine hull of points in S.
If y lies in the relative interior of the convex hull of S, then put $\hat{x} := y$ and go to Step 2.

Step 4: Let z be the point that is the nearest to y among the intersection of the convex hull of S and the line segment $[y, \hat{x}]$ between y and $\hat{x}$. Also let $S' \subset S$ be the unique subset of S such that z lies in the relative interior of the convex hull of S'. Put $S := S'$ and $\hat{x} := z$. Go to Step 3.
(End)

Illustration of the algorithm: Suppose current affine convex hull formed by *S* is $\triangle ABC$ and $\hat{x}$ is currently inside this triangle (at the beginning *S* should be point $\hat{x}$), and another point *P* that minimizes $\langle \hat{x}, P \rangle$ is found and added to *S*:



Then *ABCP* forms an affine triangular pyramid and *D* is the minimum-norm point in the affine space of *{A,B,C,P}* (the way to find *D* is introduced before as taking projection of a vertex onto the plane). If *D* is inside the convex hull of *S* (i.e. quadrilateral *ABCP*) then put $\hat{x}$ as *D* and continue to find another vertex that minimizes its inner-product with *D*. Otherwise move $\hat{x}$ towards *P* until it is about to leave *ABCP* (e.g. at point *E* on $\triangle ACP$). Designate the new $\hat{x}$ as *E* and find the smallest convex hull (which will be shown is unique) that contains *E*, which is $\triangle ACP$ and update *S* as *{A,P}*, after which carry on with step 3.

We shall prove the validity of the algorithm based on the following points:

1. Similar to finding the minimum-norm point in base polytope *B*, optimal solution to submodular function minimization can be derived by finding the minimum-norm point in the convex hull *B′* formed by these vertices of the base polytope $[y_1, ..., y_n]^T$ whose entries can be ordered as $(y_{S_1})_{k=1}^{n}$ where:

$$y_{S_1} = F(\{S_1\}), \sum_{j=1}^{k} y_{S_j} = F\left(\{S_j\}_{j=1}^{k}\right) \text{ for } 2 \leq k \leq n$$

   where *F* is the submodular function that defines the base polytope *B*. This is important as the coding implementation of the algorithm finds the minimum-norm point in *B′* instead of *B.*

2. The norm of $\hat{x}$, if it is not minimal, will continually descend.
3. After finite steps $\hat{x}$ will repeatedly become the minimum-norm point of some *convex boundary subspace* (we will soon define this) of *P′*.
4. Since the convex boundary subspaces of *P′* are finite and the minimum-norm point of *P′* is always on the boundary (i.e., is always the minimum-norm point of some convex boundary affine subspace), $\hat{x}$ will always descend and become the minimum-norm point of *P′* (and thus of the entire base polytope) in finite steps.

For point 1, we have shown that:

$$min\, F = \min_{x\in\{0,1\}^n} \max_{y\in B} \langle x, y\rangle = \max_{y\in B} \min_{x\in\{0,1\}^n} \langle x, y\rangle = y'\left(S^-(y')\right)$$

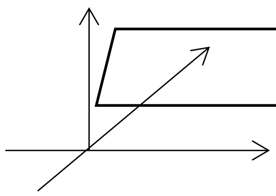
where *y′* is the minimum-norm point in *B*. Replacing *B* with *B′* for the proof of the above formula (shown in the previous document) also works.

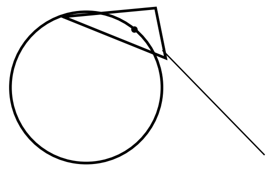
Point 4 also comes quite naturally after we give the definition of *convex boundary affine subspace*, and thus we will define it and proceed to show points 2 and 3.

Definitions for This Note and Proofs

An *affine hull* defined by points $\{x_k\}_{k=1}^{m}$ is the point set *$\{x_j+y\}$*, where *j* can be arbitrarily chosen and *y* is in the linear space spanned by $\{x_k - x_j\}_{k\neq j}$.

A *boundary* of a convex hull is its boundary **restricted to its affine hull**. For example, in the figure below, the boundary of the square is not the square itself, but its sides, although in the 3-d space the entire square is a boundary. Furthermore, a mininum-norm point needs not be on the boundary, as shown below by the sphere.





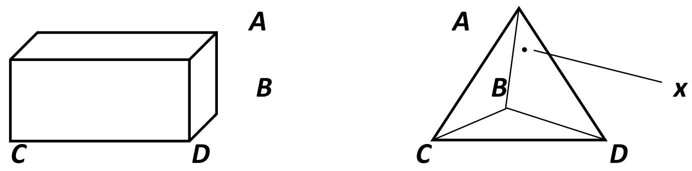
Imagine this is a sphere

Intersecting a triangle in its relative interior

The *dimension* of an affine hull (or a convex hull) with vertices $\{x_k\}_{k=1}^{m}$ is the dimension of the linear space spanned by $\{x_k - x_j\}_{k\neq j}$ for some (or each) *j*. The vertices are of *full rank* if the dimension of the affine hull formed by them is *m-1*. We may denote the convex hull formed by vertices *V* of full rank as *conv(V)*.

An affine hull with the lowest dimension is a single point, which has dimension 0.

The *convex boundary subspace* is the convex hull formed by some (but not all) vertices that are contained in the boundary. For example, in the cube below, $\triangle ABC$ is not a convex boundary subspace but $\triangle ABD$ is.



When the vertices are of full rank (i.e. there are *n+1* vertices in an n-dimensional space), then for any point *x* in the convex hull formed by these vertices, there is a unique smallest set of vertices that forms a convex hull that contains *x* (i.e., *x* is in its relative interior). For *x* in the relative interior of convex hull, the set is that of all the vertices; otherwise when *x* is on the convex boundary subspace, the set is all the vertices of this subspace, for Example *{A,B,D}* in the triangular pyramid above.

In algebraic terms, for vertices of full rank $\{x_k\}_{k=1}^{m}$ and a point *x* in the convex (or affine) hull, *x* has a unique expression $x = \sum_{k=1}^{m} a_k x_k$, then the smallest set of vertices are those with non-zero coordinates, and particularly, if *x* is a vertex then the set is *{x}* itself.

Also, a convex boundary subspace of an affine hull is at least one dimension lower than the hull itself, e.g. a convex boundary subspace of a triangular pyramid can be a triangle, which has three line segments as its convex boundary subspaces, each of which has two points as its convex boundary subspace.

With these definitions, we can prove points 2 and 3.

Proof of point 2: $\hat{x}$ always moves in a direction that reduces its norm

For a point in a polytope, it *not reaching the minimal norm* is equivalent to there *existing a vertex P such that* $\langle x, P\rangle < \langle x, x\rangle$. As illustrated below, if a vector *P* has lower inner product with *x* than *x* itself, then when *x* moves to *P* linearly, its norm must decrease at first (though it may first decrease then increase).

Contrarily if *x* has minimal inner product with itself, then all other vertices in the base polytope equal to *x+k*, with $k \neq 0$ and $\langle x, k\rangle \geq 0$, which makes *x* the minimum-norm-point in *P*.

If $\hat{x}$ has not reached minimal norm, then step 2 integrates a $\hat{p}$ into *S* and guarantees that the minimal-norm point in the affine hull formed by the new *S* has lower norm than $\hat{x}$. Then whether $\hat{x}$ becomes this minimum-norm point, when it is inside the convex hull formed by *S*, according to step 3, or otherwise $\hat{x}$ moves to it until it hits the boundary of *S*, according to step 4, $\hat{x}$ has a lower norm. ☐

As discussed previously, the direction that minimizes the inner-product with *x* is the Lovasz extension of -*x*, i.e. towards a vector $[y_1, \cdots, y_n]^T$ in the base polytope with

$$y_{S_1} = F(\{S_1\}), y_{S_2} = F(\{S_1, S_2\}) - F(\{S_1\}), \cdots$$

Proof of step 3: $\hat{x}$ always jumps to the minimum-norm point of a convex boundary subspace after finite steps

At the beginning $\hat{x}$ is a vertex, and the smallest convex hull (or convex boundary subspace) that contains it is $S = \{\hat{x}\}$. Thus $\hat{x}$ is the minimum-norm point in a convex boundary subspace of polytope *P'*, and *S* is of full rank. We will show by induction that these two properties are maintained.

If after some step, $\hat{x}$ has become the minimum-norm point in a convex boundary subspace $conv(S) \ni \hat{x}$, where *S* is the set as described in the algorithm, contains *m* vertices and is of full rank, then $\hat{x}$ may have been updated from step 3 or step 4.

If $\hat{x}$ is updated from step 4, then it will go through step 3. *S* is the smallest set of vertices that contains $\hat{x}$. $\hat{x}$ is the minimum-norm point of *conv(S)* and is in the relative interior of *S*, which means that $\hat{x}$ is also the minimum-norm point in the affine hull of *S*. The coming step 3 will reset $\hat{x}$ to itself and goes to step 2.

If *x* is updated from step 3, it is also in the relative interior if *S* and will also go through step 2. If $\hat{x}$ has achieved the minimal norm in *P'*, then the coming step 2 will terminate the algorithm, or else step 2 will include another vertex $\hat{p}$. Again, since $\hat{x}$ is the minimum-norm point of *conv(S)* and is in the relative interior of *S*, it is also the minimum-norm point of the affine hull of *S*, and the minimum-norm point of the affine hull of $S \cup \{\hat{p}\}$ will not be on the affine hull of *S*, and the new *S*, which becomes $S \cup \{\hat{p}\}$, is still of full rank. Hence all additions (and of course subtractions) of vertices to and from *S* keeps its full rank.

After *S* expands, it has *m+1* vertices and *m* dimensions. $\hat{x}$ will either:

1. Become the minimum-norm point in the affine hull of this new *S* which is also in the convex hull of *S*. Then we are done: $\hat{x}$ is already the minimum-norm point of a convex boundary subspace of *P'*. Or:
2. Move towards the minimum-norm point in the affine hull of *S* until it hits the boundary (i.e. some convex boundary subspace) of *conv(S)*. Then by step 4, *S* is updated to its unique smallest subset of vertices that forms a convex hull that contains *x*, which exists by our note following the definition of convex boundary subspace.

   From another note following this definition, the new *S* has a lower dimension than *m*, and since it is of full rank, it contains at most *m* vertices. In the next steps, $\hat{x}$ will either become the minimum-norm point of *conv(S)* or move to a convex boundary subspace, causing *S* to update to a smaller set.

As we can see, if $\hat{x}$ has not reached the minimum-norm in *conv(S)*, the size of *S* will continually shrink. This process will stop before *S* contains only a single point $\{\hat{x}\}$. Then *S* is a convex boundary subspace and $\hat{x}$ is its minimum-norm point. ☐

We have hence shown point 3 and also the validity of the minimum-norm-point algorithm.