predicates

Steven E. Pav*

December 17, 2001

1 Circumcenter

Given a sequence of points, $\{P_i\}_{i=0}^{i=n}$ in $\mathbb{R}^m$, we would like to determine the center of the smallest sphere with all the points on its surface. There may be degeneracies, in the form of

$$\text{affinedim}(\{P_i\}_{i=0}^{i=n}) < n \text{ or } n < m.$$

In the first case it may be that there is no sphere with all the points on its surface; we would like to be able to detect this situation.

If cc is at the center of a sphere with all the points on its surface, then for any pair of points cc is equidistant from them. So we should have the equation $(P_i - P_0) \cdot cc = (P_i - P_0) \cdot \frac{P_i + P_0}{2}$, an equation reflecting that cc is in the hyperplane equidistant from P_i, P_0. Also cc should be in the subspace spanned by the set of points, so there are constants α_i such that $cc = P_0 + \sum_{i=1}^{n} \alpha_i (P_i - P_0)$.[1] We let $\mathbf{x} = cc - P_0$. We revisit the hyperplane equation of above:

$$(P_i - P_0)^\top cc = (P_i - P_0)^\top \mathbf{x} + P_i^\top P_0 - P_0^\top P_0 = (P_i - P_0)^\top \frac{P_i + P_0}{2} = \frac{P_i^\top P_i - P_0^\top P_0}{2}$$

and so

$$(P_i - P_0)^\top \mathbf{x} = \frac{P_i^\top P_i - 2P_i^\top P_0 + P_0^\top P_0}{2} = \frac{(P_i - P_0)^\top (P_i - P_0)}{2}.$$

If we let $\mathbf{v_i} = P_i - P_0$, α be the column of α_i, and $\mathbf{A}$ be the matrix with the $\mathbf{v_i}$ as columns, then $\mathbf{x} = \mathbf{A}\alpha$, and the hyperplane equation is $\mathbf{v_i}^\top \mathbf{x} = \frac{\mathbf{v_i}^\top \mathbf{v_i}}{2}$, for each i. Letting $\mathbf{b}$ be the column of $\frac{\mathbf{v_i}^\top \mathbf{v_i}}{2}$ values, then

$$\mathbf{A}^\top \mathbf{A}\alpha = \mathbf{A}^\top \mathbf{x} = \mathbf{b}.$$

Since $\mathbf{x}$ is the vector from P_0 to the circumcenter, the radius of the circumsphere is $|\mathbf{x}| = \sqrt{\mathbf{x}^\top \mathbf{x}}$. If $n < m$, we have the least squares problem:

$$\min \frac{1}{2}\mathbf{x}^\top \mathbf{x} \text{ s.t. } \mathbf{A}^\top \mathbf{x} = \mathbf{b}$$

The Lagrangian is

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \lambda) &= \frac{1}{2}\mathbf{x}^\top \mathbf{x} - \lambda^\top \left(\mathbf{A}^\top \mathbf{x} - \mathbf{b} \right) \\ \nabla_x \mathcal{L}(\mathbf{x}, \lambda) &= \mathbf{x} - \mathbf{A}\lambda. \end{aligned}$$

*Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213. Supported in part by National Science Foundation

[1] Note that cc does not have to be inside the convex hull of the points, so we have no nice assumptions about the α_i.

To satisfy the KKT conditions, we set $\nabla_x \mathcal{L}(\mathbf{x}, \lambda) = 0$ and $\mathbf{A}^\top \mathbf{x} = \mathbf{b}$. This gives $\mathbf{x} = \mathbf{A}\lambda$, $\mathbf{A}^\top \mathbf{A}\lambda = \mathbf{b}$. The solution is then $\mathbf{x} = \mathbf{A}\left(\mathbf{A}^\top \mathbf{A}\right)^{-1}\mathbf{b}$.

When $m = n$, $\mathbf{A}$ is square and $\mathbf{x} = \mathbf{A}^{-\top}\mathbf{b}$.

2 Insphere Test by Distance to Circumcenter

Now, given a point p, we would like to determine if it is inside the circumball of the points $\{P_i\}_{i=0}^{i=n}$. Additionally, if the points are not affinely independent, we would like our test to tell us so.

Let $cc, \mathbf{x}, \mathbf{A}, \mathbf{v_i}, \mathbf{b}$ be as in Section 1. Consider the equation $\mathbf{x}^\top\mathbf{x} + \beta = |p - cc|^2 = |p - (P_0 + \mathbf{x})|^2$. Then

$$\beta = \begin{cases} < 0 & \text{if } p \text{ is IN the circumball,} \\ 0 & \text{if } p \text{ is ON the circumball,} \\ > 0 & \text{if } p \text{ is OUT of the circumball.} \end{cases}$$

But the equation reduces to $(p - P_0)^\top \mathbf{x} + \beta' = \frac{(p-P_0)^\top (p-P_0)}{2}$, where $\beta' = \frac{\beta}{2}$ has the same sign as β. Letting $\mathbf{w}$ be $p - P_0$, the system of equations can be expressed as

$$\begin{bmatrix} \mathbf{A}^\top & 0 \\ \mathbf{w}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \beta' \end{bmatrix} = \begin{bmatrix} \mathbf{v_1}^\top & 0 \\ \mathbf{v_2}^\top & 0 \\ \vdots & \vdots \\ \mathbf{v_n}^\top & 0 \\ \mathbf{w}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \beta' \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\mathbf{v_1}^\top \mathbf{v_1} \\ \frac{1}{2}\mathbf{v_2}^\top \mathbf{v_2} \\ \vdots \\ \frac{1}{2}\mathbf{v_n}^\top \mathbf{v_n} \\ \frac{1}{2}\mathbf{w}^\top \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \frac{1}{2}\mathbf{w}^\top \mathbf{w} \end{bmatrix}. \tag{1}$$

By Cramer's rule the solution to this equation is

$$\beta' = \frac{\det \begin{bmatrix} \mathbf{A}^\top & \mathbf{b} \\ \mathbf{w}^\top & \frac{1}{2}\mathbf{w}^\top \mathbf{w} \end{bmatrix}}{\det \begin{bmatrix} \mathbf{A}^\top & 0 \\ \mathbf{w}^\top & 1 \end{bmatrix}} = \frac{\det \begin{bmatrix} \mathbf{A}^\top & \mathbf{b} \\ \mathbf{w}^\top & \frac{1}{2}\mathbf{w}^\top \mathbf{w} \end{bmatrix}}{\det \mathbf{A}^\top} \tag{2}$$

By definition, the sequence of corners is correctly oriented if the denominator is positive, and thus maintaining orientation would appear to require one less determinant calculation. However, the denominator is a minor in the calculation of the numerator, so we calculate it anyway, and orientation is not required. Note that if we only care about the sign of β', and not its magnitude, we needn't actually perform the division. Also note that the denominator may be zero, in which case $\mathbf{A}$ is not of full rank, *i.e.*, the points are not affinely independent. However, the point set may or may not have a circumsphere in this case, and the zero does not distinguish between the two cases.

2.1 Dimensional Embedding

The above analysis suffices for the case of $n + 1$ points in $\mathbb{R}^n$. However, we may wish to check encroachment of the circumball of an n-simplex in $\mathbb{R}^m$ with $n < m$. How do we get the remaining $m - n$ equations? We recognize, as in Section 1, that $\mathbf{x} = \mathbf{A}\alpha$, for some n-vector, α. Then

$$\begin{bmatrix} \mathbf{x} \\ \beta' \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta' \end{bmatrix},$$

and equation 1 becomes

$$\begin{bmatrix} \mathbf{A}^\top & 0 \\ \mathbf{w}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta' \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top\mathbf{A} & 0 \\ \mathbf{w}^\top\mathbf{A} & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta' \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \frac{1}{2}\mathbf{w}^\top\mathbf{w} \end{bmatrix}. \tag{3}$$

Again Cramer's rule gives

$$\beta' = \frac{\det \begin{bmatrix} \mathbf{A}^\top\mathbf{A} & \mathbf{b} \\ \mathbf{w}^\top\mathbf{A} & \frac{1}{2}\mathbf{w}^\top\mathbf{w} \end{bmatrix}}{\det \begin{bmatrix} \mathbf{A}^\top\mathbf{A} & 0 \\ \mathbf{w}^\top\mathbf{A} & 1 \end{bmatrix}} = \frac{\det \begin{bmatrix} \mathbf{A}^\top\mathbf{A} & \mathbf{b} \\ \mathbf{w}^\top\mathbf{A} & \frac{1}{2}\mathbf{w}^\top\mathbf{w} \end{bmatrix}}{\det \mathbf{A}^\top\mathbf{A}} \tag{4}$$

Again the denominator is a minor in the calculation of the numerator, so we get it at no additional cost. However, we note that $\mathbf{A}^\top\mathbf{A}$ is positive definite if $\mathbf{A}$ is nonsingular, and zero otherwise. So it won't change the sign of the test, but may reveal degeneracies, as noted above.

There remains the question of whether this form is more susceptible to degeneracy or innaccuracy than some other form of insphere; this suspicion is based on the general behaviour of such normal form equations.

3 Insphere Test by Projecting to a Parabola

We may perform an insphere test by projecting points up to the parabola in the next higher dimension and using a plane-side test. We assume that the sequence of points $\{P_i\}_{i=0}^{i=n}$ are affinely independent in $\mathbb{R}^n$. We project the points to the parabola in $\mathbb{R}^{n+1}$ via $\psi(P) = (P, \frac{P^\top P}{2})$. Then the encroachment test is related to the sign of the determinant of $\{\psi(P_i)\}_0^n \cup \{\psi(p)\}$, which is the volume (up to orientation) of the $(n+1)$-simplex of these $n+2$ corners.

To solve the problem of orientation, we use the fact that the centroid (center of gravity) of the $n+1$ points does encroach the circumball. Let g be the centroid. Then we should calculate

$$\beta = \frac{\begin{bmatrix} \psi(P_1) - \psi(P_0)^\top \\ \psi(P_2) - \psi(P_0)^\top \\ \vdots \\ \psi(P_n) - \psi(P_0)^\top \\ \psi(p) - \psi(P_0)^\top \end{bmatrix}}{\begin{bmatrix} \psi(P_1) - \psi(P_0)^\top \\ \psi(P_2) - \psi(P_0)^\top \\ \vdots \\ \psi(P_n) - \psi(P_0)^\top \\ \psi(g) - \psi(P_0)^\top \end{bmatrix}} = \frac{\begin{bmatrix} \psi(P_0)^\top & 1 \\ \psi(P_1)^\top & 1 \\ \vdots & \vdots \\ \psi(P_n)^\top & 1 \\ \psi(p)^\top & 1 \end{bmatrix}}{\begin{bmatrix} \psi(P_0)^\top & 1 \\ \psi(P_1)^\top & 1 \\ \vdots & \vdots \\ \psi(P_n)^\top & 1 \\ \psi(g)^\top & 1 \end{bmatrix}}.$$

The computation of the denominator can be performed at little additional cost. An alternative form first performs a translation to make P_0 the origin of the coordinate system in $\mathbb{R}^n$, then applies ψ. This results in a bunch of corners of a simplex in $\mathbb{R}^{n+1}$, one of which is the origin. So we can

calculate

$$\beta' = \frac{\begin{bmatrix} \psi(P_1 - P_0)^\top \\ \psi(P_2 - P_0)^\top \\ \vdots \\ \psi(P_n - P_0)^\top \\ \psi(p - P_0)^\top \end{bmatrix}}{\begin{bmatrix} \psi(P_1 - P_0)^\top \\ \psi(P_2 - P_0)^\top \\ \vdots \\ \psi(P_n - P_0)^\top \\ \psi(g - P_0)^\top \end{bmatrix}}.$$

But letting $\mathbf{v_i}$ be $P_i - P_0$, as in Section 1, we have $\psi(\mathbf{v_i}) = (\mathbf{v_i}, \frac{\mathbf{v_i}^\top \mathbf{v_i}}{2})$, so the numerator is exactly the same as in equation 2. Given that the parabola form is usually implemented with orientation assumptions, the two forms would be identical, with the exception that generalizing the parabola form to the dimension-embedded case is not at all clear.

4 Implementation

How will we actually implement these tests? We consider the cases:

- **n-d in n-d** We use equation 2, and deal with orientation by returning the minor which is the denominator.
- **1-d in n-d** We use equation 4, but there are some simplifications; since $\mathbf{A}$ will be a column vector, the denominator will be positive, so it won't affect the sign of β', and needn't be computed. Moreover the equation reduces to

$$\beta'' = \det \begin{bmatrix} \mathbf{v_1}^\top \mathbf{v_1} & \frac{1}{2}\mathbf{v_1}^\top \mathbf{v_1} \\ \mathbf{w}^\top \mathbf{v_1} & \frac{1}{2}\mathbf{w}^\top \mathbf{w} \end{bmatrix} = \frac{1}{2}(\mathbf{v_1}^\top \mathbf{v_1}\mathbf{w}^\top \mathbf{w} - \mathbf{w}^\top \mathbf{v_1}\mathbf{v_1}^\top \mathbf{v_1}) = \frac{\mathbf{v_1}^\top \mathbf{v_1}}{2}\mathbf{w}^\top(\mathbf{w} - \mathbf{v_1}).$$

  Again, $\mathbf{v_1}^\top \mathbf{v_1}$ is positive, so the test reduces to the sign of $\mathbf{w}^\top(\mathbf{w} - \mathbf{v_1})$, which can be interpreted geometrically as the angle subtended by the segments from p to the two endpoints of the diameter: when this angle is obtuse, the point is internal to the sphere and the dot product is negative. It is thrilling to see that this checks. Of course, this assumes that $\mathbf{v_1}$ is nonzero.
- **n-d in m-d** We use the normal equation form of equation 4. Are there nice reductions as in the 1-d case? Is there some kind of Thale's theorem for solid angles?

4.1 code

The source code predicates follow

```
val insphere1to2 =
fn [ ax, ay, bx, by, px, py ] =>
let
  val vx = px - ax
```

```
  val vy = py - ay
  val wx = px - bx
  val wy = py - by
  val insphere = ( vx * wx ) + ( vy * wy )
end

val insphere2to2 =
fn [ ax, ay, bx, by, cx, cy, px, py ] =>
let
  val brx = bx - ax
  val bry = by - ay
  val crx = cx - ax
  val cry = cy - ay
  val prx = px - ax
  val pry = py - ay

  val brSqrd = ( sq brx ) + ( sq bry )
  val crSqrd = ( sq crx ) + ( sq cry )
  val prSqrd = ( sq prx ) + ( sq pry )

  val xMinor = ( bry * crSqrd ) - ( cry * brSqrd )
  val yMinor = ( brx * crSqrd ) - ( crx * brSqrd )
  val denom = ( brx * cry ) - ( crx * bry )

  val numer = prx * xMinor - pry * yMinor + prSqrd * denom
  val insphere = numer * denom
end

val insphere1to3 =
fn [ ax, ay, az, bx, by, bz, px, py, pz ] =>
let
  val vx = px - ax
  val vy = py - ay
  val vz = pz - az
  val wx = px - bx
  val wy = py - by
  val wz = pz - bz
  val insphere = ( vx * wx ) + ( vy * wy ) + ( vz * wz )
end

val insphere2to3 =
fn [ ax, ay, az, bx, by, bz, cx, cy, cz, px, py, pz ] =>
let
  val brx = bx - ax
  val bry = by - ay
  val brz = bz - az
  val crx = cx - ax
  val cry = cy - ay
  val crz = cz - az
  val prx = px - ax
  val pry = py - ay
  val prz = pz - az

  val v1v1 = ( sq brx ) + ( sq bry ) + ( sq brz )
  val v2v2 = ( sq crx ) + ( sq cry ) + ( sq crz )
  val ww = ( sq prx ) + ( sq pry ) + ( sq prz )
  val v1v2 = ( brx * crx ) + ( bry * cry ) + ( brz * crz )
  val wv1 = ( brx * prx ) + ( bry * pry ) + ( brz * prz )
  val wv2 = ( prx * crx ) + ( pry * cry ) + ( prz * crz )
```

```
  val xMinor = ( v1v2 * wv2 ) - ( wv1 * v2v2 )
  val yMinor = ( v1v1 * wv2 ) - ( wv1 * v1v2 )

  val denom = ( v1v1 * v2v2 ) - ( v1v2 * v1v2 )
  val numer = ( v1v1 * xMinor ) - ( v2v2 * yMinor ) + ( ww * denom )
  val insphere = numer * denom
end

val insphere3to3 =
fn [ ax, ay, az, bx, by, bz, cx, cy, cz, dx, dy, dz, px, py, pz ] =>
let
  val brx = bx - ax
  val bry = by - ay
  val brz = bz - az
  val crx = cx - ax
  val cry = cy - ay
  val crz = cz - az
  val drx = dx - ax
  val dry = dy - ay
  val drz = dz - az
  val prx = px - ax
  val pry = py - ay
  val prz = pz - az

  val brSqrd = ( sq brx ) + ( sq bry ) + ( sq brz )
  val crSqrd = ( sq crx ) + ( sq cry ) + ( sq crz )
  val drSqrd = ( sq drx ) + ( sq dry ) + ( sq drz )
  val prSqrd = ( sq prx ) + ( sq pry ) + ( sq prz )

  val bxmin  = ( cry * drz ) - ( dry * crz )
  val bymin  = ( crx * drz ) - ( drx * crz )
  val bzmin  = ( crx * dry ) - ( drx * cry )

  val cxmin  = ( bry * drz ) - ( dry * brz )
  val cymin  = ( brx * drz ) - ( drx * brz )
  val czmin  = ( brx * dry ) - ( drx * bry )

  val dxmin  = ( bry * crz ) - ( cry * brz )
  val dymin  = ( brx * crz ) - ( crx * brz )
  val dzmin  = ( brx * cry ) - ( crx * bry )

  val xMinor = ( bxmin * brSqrd ) - ( cxmin * crSqrd ) + ( dxmin * drSqrd )
  val yMinor = ( bymin * brSqrd ) - ( cymin * crSqrd ) + ( dymin * drSqrd )
  val zMinor = ( bzmin * brSqrd ) - ( czmin * crSqrd ) + ( dzmin * drSqrd )
  val denom = ( brx * bxmin ) - ( crx * cxmin ) + ( drx * dxmin )

  val numer = ( pry * yMinor - prx * xMinor ) + ( prSqrd * denom - prz * zMinor )
  val insphere = numer * denom
end
```