

Geometría Analítica I

LECTURA 10

Ayudante: Guilmer González

Día 23 de noviembre, 2015

El día de hoy veremos:

0. Comentarios sobre los trabajos últimos.
1. Sobre la distancia de un punto a un polígono.

1 Distancia de un punto a un segmento

El segmento es parte de una recta. Sabemos como calcular la distancia de un punto que no se encuentra sobre la recta a esta, básicamente usando proyecciones. Para el caso de segmentos, debemos “restringir” el problema o adaptarlo a esta modalidad.

Si sobre el segmento $\overline{P_1P_2}$, consideramos la recta que pasa por los puntos, podemos proyectar el punto sobre la recta, llamemos al pie de la proyección P^* . Sin embargo, en algunas situaciones, es la distancia del punto Q a alguno de los dos puntos la más cercana. Si usamos la propiedad de que el segmento $\overline{QP^*}$ es perpendicular al segmento $\overline{P_1P_2}$, podemos observar regiones formadas por rectas perpendiculares al segmento.

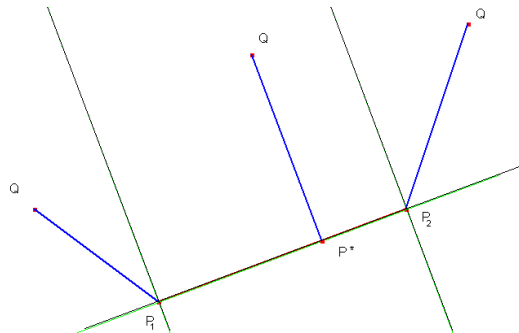


Figura 1: Distancia de un punto Q a un segmento $\overline{P_1P_2}$.

Como puede observarse, si el punto Q está en la región contenida por las dos recta perpendiculares, la distancia del punto al segmento es $d(Q, \overline{P_1P_2}) = \|Q - P^*\|$. Si Q está afuera, entonces

$$d(Q, \overline{P_1P_2}) = \min\{\|Q - P_1\|, \|Q - P_2\|\}$$

La cuestión en este problema es hacer un algoritmo que de manera automática nos diga cómo calcular esa distancia. La idea es saber si la proyección P^* está dentro del segmento o fuera.

Algoritmo, pseudo-código

- 1) Construya la recta que pasa por esos dos puntos. $P(t) = P_1 + t(P_2 - P_1)$.
- 2) Proyecte el punto P sobre la recta, $P^*(t^*) = P_1 + t^*(P_2 - P_1)$.
- 3) Si el punto P^* está sobre el segmento,

$$d(Q, \overline{P_1P_2}) = \|Q - P^*\|$$

- 4) Si el punto P^* está fuera del segmento,

$$d(Q, \overline{P_1P_2}) = \min\{\|Q - P_1\|, \|Q - P_2\|\}$$

Ahora, vamos a hacer los pasos de manera analítica.

Se parametriza la recta en la forma:

$$P(t) = P_1 + t(P_2 - P_1)$$

fijemos un punto de referencia entre los segmentos, digamos P_1 . Sobre este punto formamos el vector $\overrightarrow{P_1Q} = \overrightarrow{OQ} - \overrightarrow{OP_1}$. Proyectemos este vector sobre el vector dirección de la recta $\overrightarrow{P_1P_2}$, ese vector parte de P_1 a P^* , el pie de la proyección:

$$\overrightarrow{P_1P^*} = \frac{(\overrightarrow{P_1P_2}) \cdot (\overrightarrow{P_1Q})}{(\overrightarrow{P_1P_2}) \cdot (\overrightarrow{P_1P_2})} \overrightarrow{P_1P_2}$$

observe que

$$t^* = \frac{(\overrightarrow{P_1P_2})(\overrightarrow{P_1Q})}{(\overrightarrow{P_1P_2}) \cdot (\overrightarrow{P_1P_2})}$$

es el parámetro para $P^* = P(t^*)$. Si $0 \leq t^* \leq 1$, P^* está dentro del segmento, si $t^* < 0$, P^* a la izquierda de P_1 , por consiguiente, P está en la franja donde los puntos están más cercanos a P_1 que a P_2 . Si $t^* > 1$, P está en la franja donde los puntos están más cercanos a P_2 que a P_1 . En Matlab, esto se escribe como

```
% function dist=dpto2seg(p,a,b)
%
% Esta funcion calcula la distancia de un punto a un segmento
%
% Modo de uso:
%           dist=dpto2seg(p,a,b)
%
% * Calculamos la proyeccion de p sobre el segmento
%   solamente nos interesa su parametrizacion
%
%
% donde p=(x_p,y_p) es el punto desde el que se desea calcular
% y a=(x_a,y_a), b=(x_b,y_b), los vertices del segmento.
%
% Ultima revision: 21 de octubre de 2006.
%
% Copyright(c) 2006 Seminario EjeMedial
%           Facultad de Ciencias - UNAM

u = b-a; % calculamos el vector direccion del segmento
alpha = (p-a)'*(u)/norm(u)^2;

if (alpha<0),
    dist=norm(p-a); % El punto esta a la izquierda del segmento
elseif(alpha>1),
    dist=norm(p-b); % El punto esta a la derecha del segmento
```

```

else
    dist=norm(a + alpha* u - p); % Calculamos la distancia del punto al pie
end                                % de la proyeccion
%
% * Last card
%
```

2 Distancia de un punto a un polígono

Ahora generalizamos la idea, y consideremos que se tiene por figura $\mathcal{F}_1 = \{Q\}$, y a la figura $\mathcal{F}_2 = \Gamma$ un polígono. Veamos como resolver $d(Q, \mathcal{F})$.

Para enfocar ideas, consideremos que nuestro polígono Γ está formado por dos segmentos (por 3 puntos).

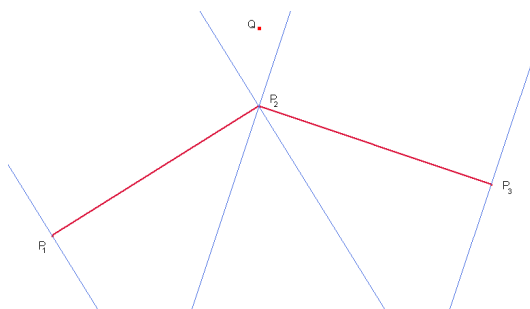


Figura 2: Distancia de un punto Q a un polígono de dos segmentos.

Si observamos la figura anterior, este problema está ligado con el anterior, en el sentido de que la figura está formada por segmentos, luego es fácil ver que se tienen 6 regiones de estudio, a partir de las líneas verticales sobre cada segmento. Lo importante aquí, es identificar que la figura es un polígono, no puntos aislados, así, la distancia de Q a $\Gamma = \Gamma(P_1, P_2, P_3)$ será aquella que sea la menor a uno de los dos segmentos. Esto es, el problema se puede expresar en la forma

$$\begin{aligned}
 d(Q, \Gamma(P_1, P_2, P_3)) &= \min_{P^* \in \Gamma(P_1, P_2, P_3)} d(Q, P^*) \\
 &= \min \{d(Q, \overline{P_1P_2}), d(Q, \overline{P_2P_3})\}
 \end{aligned}$$

bajo esta observación (un polígono es la unión de segmentos), podemos generalizar esa idea a cualquier polígono Γ formado por m segmentos

$$\begin{aligned} d(Q, \Gamma(P_1, P_2, \dots, P_{m+1})) &= \min_{P^* \in \Gamma(P_1, P_2, \dots, P_{m+1})} d(Q, P^*) \\ &= \min \{d(Q, \overline{P_1 P_2}), \dots, d(Q, \overline{P_m P_{m+1}})\} \end{aligned}$$

En Matlab, podemos hacer una rutina para calcular la distancia entre un punto, y un polígono reusando el código que hemos elaborado para resolver el subproblema punto-segmento, este se escribe como

```
function [dist,vdist]=dpto2poly(p,Poly)
%function [dist,d]=dpto2poly(p,Poly)
%
% Esta funcion calcula la distancia de un punto a un poligono
%
% Modo de uso
%   [dist,d]=dpto2poly(p,Poly)
%
% donde p=(x_p,y_p) es el punto desde el que se desea calcular
% y Poly = [ ...
%           x_1  y_1
%           x_2  y_2
%           ...
%           x_m  y_m ]
%
% el una matrix de m x 2 que contiene los vertices del poligono.
%
% Ultima revision: 21 de octubre de 2006.
%
% Copyright(c) 2006 Seminario EjeMedial
%                   Facultad de Ciencias - UNAM
%
[m,n]=size(Poly); % Tomamos el tamaño del vector
```

```

p1=Poly(1,:);
for i=2:m,
    p2 = Poly(i,:);
    vdist(i-1)=dpto2seg(p,p1,p2); % Calculamos la distancia punto segmento
    p1=p2;
end
%
% * Calculamos el minimo sobre todas las distancias
%
    dist = min(vdist);
%
% * Last card
%
```