7. Interpolación y aproximación

Vídeo de Interpolación y aproximación de superficies

Supongamos que tenemos una nube de $(m+1)\cdot(n+1)$ datos en el espacio, $\{a_{0,0},\dots,a_{m,n}\}$, por los cuales queremos que pase por una superficie, $c(u,v)$, para valores determinados de dos parámetros u,v,

$$a_{i,j} = c(u_i, v_j), \qquad i = 0,\dots,m \qquad j = 0,\dots,n. \tag{26}$$

Intentaremos ajustarlos con una parametrización producto de una superficie de bigrado (m,n). En vez de atacar el problema como un sistema lineal de $(m+1)\cdot(n+1)$ incógnitas, los vértices de la malla de control, haremos uso del producto de polinomios para escribirlo como $B_U C B_V^t = A$,

$$A = \begin{pmatrix} a_{0,0} & \cdots & a_{0,n} \\ \vdots & \ddots & \vdots \\ a_{m,0} & \cdots & a_{m,n} \end{pmatrix}, \qquad B_U = \begin{pmatrix} B_0^m(u_0) & \cdots & B_m^m(u_0) \\ \vdots & \ddots & \vdots \\ B_0^m(u_m) & \cdots & B_m^m(u_m) \end{pmatrix}$$
$$C = \begin{pmatrix} c_{0,0} & \cdots & c_{0,n} \\ \vdots & \ddots & \vdots \\ c_{m,0} & \cdots & c_{m,n} \end{pmatrix}, \qquad B_V = \begin{pmatrix} B_0^n(v_0) & \cdots & B_n^n(v_0) \\ \vdots & \ddots & \vdots \\ B_0^n(v_n) & \cdots & B_n^n(v_n) \end{pmatrix}, \tag{27}$$

donde las matrices A, C definen tres sistemas lineales, uno por cada coordenada de los puntos.

Las coordenadas de los vértices de la malla de control se obtienen formalmente invirtiendo las matrices, $C = B_U^{-1} A (B_V^t)^{-1}$. Aunque es más sencillo reducir el problema al ya estudiado de las curvas interpolantes, definiendo matrices auxiliares $\tilde{C} := B_U C$, que serán la incógnita del sistema $\tilde{C} B_V^t = A$.

Por tanto, hemos reducido la solución del problema a dos pasos. Primero resolvemos los sistemas $\tilde{C} B_V^t = A$ y con la solución obtenida, $\tilde{C}$, planteamos $B_U C = \tilde{C}$, cuya solución es la malla de control de la superficie interpolante.

Nótese que con esta argucia hemos reducido el problema inicial, un sistema lineal de $(m+1)\cdot(n+1)$ ecuaciones a dos sistemas, de $(m+1)$ y $(n+1)$ ecuaciones, respectivamente, con el consiguiente ahorro de operaciones. Ejemplo.

Este planteamiento puede emplearse también para aproximar un conjunto de datos, sustituyendo las curvas interpolantes por curvas aproximantes.

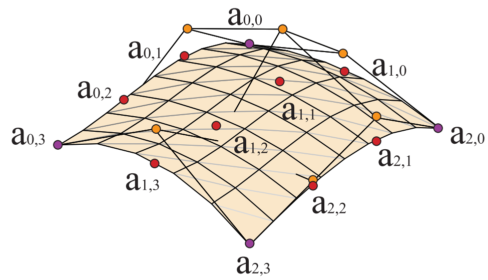

Figura 19: Superficie interpolante de bigrado (2,3)

Otra opción, mucho más frecuente, es interpolar mediante superficies B-spline bicúbicas. El planteamiento es esencialmente el mismo, sólo hay que sustituir en nuestro sistema los polinomios de Bernstein por funciones B-spline.

Supongamos que nuestros datos deben verificar $a_{i,j} = c(u_i, v_j)$, para $i = 0, \ldots, M$, $j = 0, \ldots, N$. Podemos fijar u_i e interpolar mediante un *spline* cúbico los puntos de la fila correspondiente, $\{a_{i,0}, \ldots, a_{i,N}\}$. Obtenemos $M + 1$ polígonos B-spline, uno por cada fila, que forman una matriz de puntos $\{\tilde{d}_{0,0}, \ldots, \tilde{d}_{M,N+2}\}$. Interpolamos con *splines* cúbicos cada columna $\{\tilde{d}_{0,j}, \ldots, \tilde{d}_{M,j}\}$ y obtenemos una familia de $N + 3$ polígonos B-spline, $\{d_{0,0}, \ldots, d_{M+2,N+2}\}$, que constituyen la malla B-spline de la superficie interpolante. Las sucesiones de nudos son las propias $\{u_0, \ldots, u_M\}$, $\{v_0, \ldots, v_N\}$, con los nudos inicial y final repetidos tres veces. Obviamente, hace falta imponer condiciones en los extremos para que el problema de interpolación tenga solución única. Ejemplo.

También podríamos haber resuelto el problema interpolando primero las columnas de datos y después las filas de vértices obtenidos.

En general, no es de esperar que nuestros datos estén organizados en el espacio de acuerdo a un patrón establecido, menos aún una malla aproximadamente rectangular. Lo usual es que debamos asignar a cada dato a_i unos parámetros (u_i, v_i) que no estén situados sobre una malla rectangular.

Consideremos el problema de aproximación, en el cual queremos obtener una superficie de bigrado (m, n) que aproxime un conjunto de datos,

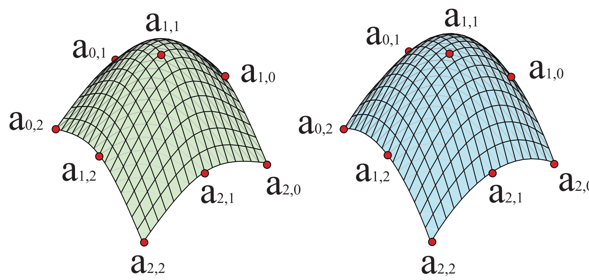

Figura 20: Superficie spline cúbica interpolante de una malla de 3×3 puntos con condiciones naturales y tangentes de Bessel

$\{a_0, \ldots, a_M\}$, tales que $a_i = c(u_i, v_i)$. El sistema lineal del problema,

$$\begin{pmatrix} B_0^m(u_0)B_0^n(v_0) & \cdots & B_m^m(u_0)B_n^n(v_0) \\ \vdots & \ddots & \vdots \\ B_0^m(u_M)B_0^n(v_M) & \cdots & B_m^m(u_M)B_n^n(v_M) \end{pmatrix} \begin{pmatrix} c_{0,0} \\ \vdots \\ c_{m,n} \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_M \end{pmatrix}, \quad (28)$$

tiene $(m+1)\cdot(n+1)$ incógnitas para $M+1$ ecuaciones, así que para un número alto de datos, el sistema está sobredeterminado.

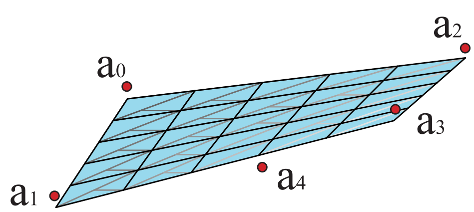

Figura 21: Superficie spline de bigrado (1,1) aproximante de una malla de 5 puntos

Recurrimos, pues, a la aproximación por mínimos cuadrados. Si escribimos nuestro sistema como $BC = A$, lo podemos transformar, como hicimos en el problema equivalente para curvas, en un problema de $(m+1)\cdot(n+1)$ ecuaciones e incógnitas,

$$B^t BC = B^t A, \quad (29)$$

que, en principio, es abordable, aunque para un número elevado de datos el sistema está mal condicionado y no es soluble numéricamente.

Como ya sucedía en el caso de las curvas, no es común tener los parámetros (u_i, v_i) de los datos, sino que el diseñador tendrá que imponerlos de alguna manera eficiente. Esta es una tarea ardua y complicada.

Una opción sería proyectar los puntos sobre un plano, por ejemplo, el XY y leer las coordenadas (x_i, y_i) de los datos como parámetros, transformándolos si es preciso mediante un cambio de escala para que estén contenidos en nuestro dominio, el cuadrado $[0, 1] \times [0, 1]$.