Least Squares Approximation

In many situations interpolation is not warranted. Errors in the data, smoothness constraints, data compression, local control, computational cost, physical theory, etc. are all reasons for allowing the approximating function to pass "near" - rather than "through" - the data.

This data-fitting, or curve-fitting, is fundamentally an optimization problem: "What is the 'best' approximating function (from some fixed set) for this data?" Two things need to be decided for this problem to be well defined. First, what class of function is one considering? Polynomials, trigonometric functions, wavelets, exponential functions, splines, and eigenfunctions associated with some related operator, are all examples of classes of functions commonly used in this context. Second, what does one mean by "best"? Here the best will minimize the sum of the squares of the (vertical) error ($\|\cdot\|_2$ minimization), but there are as many reasonable definitions of "best" for a given application as there are reasonable measures of distance for that application.

Let S denote the set of functions that we are approximating with. If we are given an integrable function F over some interval $[a, b]$, the *continuous least squares* problem is to find a function $f \in S$ which minimizes the "sum" of the squares of the difference, i.e.

$$f = \operatorname*{argmin}_{g \in S} \left(\int_a^b (F(x) - g(x))^2 dx \right)^{1/2}.$$

On the other hand, if we are given the data $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$, and we let $x = (x_1, x_2, \ldots, x_n)^t$ and $y = (y_1, y_2, \ldots, y_n)^t$, then the *discrete least squares* solution for this data is a function

$$f = \operatorname*{argmin}_{g \in S} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 \right)^{1/2}.$$

In some ways the distinction between the continuous and the discrete cases is transparent:

$$f = \operatorname*{argmin}_{g \in S} \|y - g(x)\|_2.$$

In general, this can be an arbitrarily difficult problem (e.g. if S is a large discrete set), but if S is a vector space then this *linear least squares* problem is much more simple, in fact we can do the finite dimensional case right here: Let $\phi_1, \phi_2, \ldots, \phi_m$ be any basis for S and let V be the $n \times m$ matrix $V = [\phi_j(x_i)]$ (this is a bit different for the continuous case). Then $f = \sum_{i=1}^{m} c_i \phi_i$, where

$$c = \arg \min_{c \in \mathbb{R}^m} \|y - Vc\|_2.$$

We can use calculus (set the gradient equal to zero), or inner products (set $(y - Vc) \perp S$) to show that the solution to the minimization problem above uniquely satisfies the *normal equations*

$$V^t V c = V^t y.$$

How you compute f is another story...