

Computational Physics / PHYS-UA 210 / Problem Set #6
Due October 11, 2017

You *must* label all axes of all plots, including giving the *units*!!

This homework focuses on fitting a linear model to a data set. (Please note as an important point: when we say “fitting a linear model” it means “fitting a model whose predictions vary linearly with its parameters,” not “fitting y vs. x with a line.”).

1. Generate a set of “random” (x, y) data with constant noise, using x in the range from 0 to 1, and with y determined by an 8th order polynomial:

$$y_i = \left[\sum_{j=0}^8 \alpha_j (x - 0.5)^j \right] + \text{Gaussian noise} \quad (1)$$

Choose reasonable α_j , and reasonable Gaussian noise (i.e. noticeable but not much larger than the features in your polynomial).

2. For some (possibly different) set of coefficients, β_j , sum-squared residuals of the model \hat{y} are:

$$S = \sum_i (y_i - \hat{y}_i)^2 = \sum_i \left[y_i - \sum_j \beta_j (x_i - 0.5)^j \right]^2 \quad (2)$$

which can be written as:

$$S = \left| \mathbf{A} \cdot \vec{\beta} - \vec{y} \right|^2 \quad (3)$$

Construct the matrix \mathbf{A} given your random \vec{x} .

3. Use SVD to find the $\vec{\beta}$ that minimizes S . This is the linear least-squares estimate of $\vec{\alpha}$. Compare your model for y with the y_i values and with the original, correct y . Try using different numbers of random draws: 6, 8, 32, 128. Compare including Gaussian noise to not adding any noise.
4. Compare using SVD to solving the “normal equations.” The normal equations result from finding where:

$$\frac{\partial S}{\partial \vec{\beta}} = 0 \quad (4)$$

and yield the equation:

$$(\mathbf{A}^T \cdot \mathbf{A}) \cdot \vec{\beta} = \mathbf{A}^T \cdot \vec{y} \quad (5)$$

This matrix equation can be solved by inverting $\mathbf{A}^T \cdot \mathbf{A}$, which is $N \times N$. E.g. you can just use the `numpy.linalg.inv` implementation of LU decomposition. Try this technique for the examples you used SVD on and describe any differences you see.