# MonomialHyperribbonsHintsPython

January 27, 2024

Monomial hyperribbons

(Sethna)

We saw in "Sloppy monomials' ' that the monomial coefficients for polynomial fits to data are ill-determined, and have sloppy eigenvalues for their Hessian. While linear fits with unconstrained coefficients have an unbounded model manifold (an infinite hyperplane, so not a hyperribbon), they allow arbitrarily large gradients in the resulting fit, which are not usually expected in practice and often suppressed by nonlinearities in realistic models (where parameters can often go to infinity in ways that keep the predictions bounded).

```
[ ]: # Sometimes gives interactive new windows
     # Must show() after plot, figure() before new plot
     # %matplotlib

     # Adds static figures to notebook: good for printing
     %matplotlib inline

     # Interactive windows inside notebook! Must include plt.figure() between plots
     # %matplotlib notebook

     # Better than from numpy import *, but need np.sin(), np.array(), plt.plot(),␣
      ↪etc.
     import numpy as np
     import matplotlib.pyplot as plt
```

Here we consider the model manifold for polynomials $y(x) = \sum_{\alpha=0}^{N-1} \theta_\alpha x^\alpha$ with bounds on the parameters $\theta_\alpha$. The Jacobian

$$J_{m\alpha} = \left.\frac{\partial y}{\partial t}\right|_{x_m} \tag{1}$$

can be viewed as mapping vectors in parameter space onto vectors $= \delta y(\mathbf{x})$ in prediction space, where $\mathbf{x} = \{x_1, \dots, x_M\}$ are the locations of the data points being fit. That is, $y_+(x_m) = y(x_m) + \sum_\alpha J_{m\alpha}\delta_\alpha$.

(a) Show (or note) that $J_{m\alpha} = x_m^\alpha$.

Your answer here. Double click to edit.

Thus

$$J = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_M & x_M^2 & \cdots & x_M^{N-1} \end{pmatrix} \tag{2}$$

is the famous Vandermonde matrix. If $M = N$, this matrix is square, and its determinant is the ratio of the volume of the allowed parameters  and the volume of the resulting model manifold. The famous result is that this determinant is given by $\det(J) = \prod_{1\leq i<j\leq N}(x_i - x_j)$. This can be seen by observing that $\det J$ obeys the rule that swapping rows of $J$ swaps the sign of the determinant, that the determinant has the correct net degree in the $x_i$ (the same degree $N(N-1)/2$ as the product of the diagonal entries), and then checking for the overall multiplicative constant.

Let us consider the constrained system where the monomial coefficients lie in a sphere $\sum_\alpha \theta_\alpha^2 < N$, and consider fits over the unit interval $x \in (0,1)$. Each $\theta$ on average then has a variance of order one, and we are considering the behavior over a distance of order one. This bound corresponds to controlling the derivatives of the function at zero, since $\theta_\alpha = y^{(\alpha)}(0) = \partial^\alpha y\,/\partial t^\alpha|_{x=0}$. Our general theorem for nonlinear least-squares models demands a stronger constraint on the functions $f(x)$ – that the sum of the squares of the $m$th derivatives be less than $N$ times a radius of convergence $R^{-m}$ for every $x$ in the interval. They also use the fact that polynomials have the biggest range of predictions given the Taylor series bounds, so your calculation is reproducing much of the qualitative physics of the rigorous proof.

If the typical distance between the points $x_i$ is $\Delta x$, then the determinant $\det J$ is $\sim (\Delta x)^{N(N-1)/2}$, which becomes really, really tiny as $N$ gets large and the minimal spacing $L/M$ gets small. As soon as the number of data points per radius of convergence becomes larger than two, the volume of the model manifold gets progressively smaller.

(b) Taking $M = 6$ equally spaced points on $[0,1]$ and $N = 6$ parameters, numerically check that the determinant of our Vandermonde matrix $J$ is tiny.

```
[ ]: def J(M,N):
         """ Vandermonde matrix for points x_m and powers up to N"""
         return np.array([[... for ... in range(N)] for ... in np.linspace(0,1,M)])

     print(J(6,6))
     print("Determinant =", np.linalg.det(...)
```

Is this volume small because the the predictions are squeezed into a hyperribbon? If the widths along the $n$th direction scale as $w_n = (\Delta x)^n$, then this would work, since $\prod_{n=1}^N w_n = (\Delta x)^{\sum_{n=1}^N n} = (\Delta x)^{N(N-1)/2}$. But usually the number of predictions $M$ is larger than the number of parameters $N$, so $J$ isn't a square matrix. What mathematical operation gives us the shape of the image of the unit sphere? Singular value decompositions is a powerful generalization of eigenvector decomposition, and precisely serves this purpose.

Singular value decomposition is not well studied in physics, where we usually care about square matrices that are symmetric or Hermitian. See the excellent Wikipedia article on SVD. The theory says that any matrix of real numbers can be decomposed into a product of three matrices:

$$\begin{aligned} J &= U\Sigma V^T \\ J_{i\alpha} &= U_{ij}\Sigma_{j\beta}(V^T)_{\beta\alpha}. \end{aligned} \tag{3}$$

Here $U$ is $M \times M$, $V$ is $N \times N$, and $\Sigma$ is $M \times N$ and diagonal (until the diagonal hits one of the far sides of the rectangle). Here the columns of $U$ and $V$ (and hence the rows of $V^T$) are an orthonomal basis for the behavior space and the parameter space, and are called the left-singular vectors and the right-singular vectors of J, respectively. (This makes $U^TU$ and $V^TV$ the $M \times M$ and $N \times N$ identity matrices: they are unitary.) Assuming $M > N$, the first $N$ basis vectors of $U$ span the tangent to the model manifold. The $\alpha$th right singular vectors of $V$ maps onto the $\alpha$th left singular vector of $U$ after being stretched an amount given by $\sigma_{\alpha\alpha}$.

(c) Taking $M = 11$ equally spaced points and $N = 6$ parameters, dig up the appropriate SVD routine and find $U$, $\Sigma$, and $V$ for our our Vandermonde Jacobian J. (The left singular vectors of $U$ are unit vectors on our sphere in parameter space. In behavior space, our sphere becomes an ellipsoid, with axes along the right singular vectors.) By how much are the unit axes of our sphere of parameters being squashed in behavior space? Is our hyperellipsoid model manifold roughly thinner by a constant factor for each new parameter?

```
[ ]: U, Sigma, V = np.linalg.svd(...)
     print(Sigma)
     print("ratios = ", ...)
```

Your answer here. Double click to edit.

By how much are the unit axes of our sphere of parameters being squashed in behavior space? Is our hyperellipsoid model manifold roughly thinner by a constant factor for each new parameter?

Finally, let us connect the skewness of $J$ and its singular values $\Sigma_{\alpha\alpha}$ with the Hessian $\mathcal{H}_{\alpha\beta} = g_{\alpha\beta} = (J^TJ)_{\alpha\beta}$.

(d) Show analytically using the singular value decomposition that the eigenvalues of $\mathcal{H}$ are the squares of the singular values of $J$. What are the eigenvectors, in terms of the left and right singular vectors?

Your answer here. Double click to edit.