Report Q1: Is the polynomial part needed and why? [1]

Polynomials are equations of a single variable with nonnegative integer exponents. The poly function converts the roots back to polynomial coefficients.

Polynomials are, essentially by definition, precisely the operations one can write down starting from addition and multiplication. More formally, polynomials with coefficients in a commutative ring RR are precisely the morphisms in the Lawvere theory of commutative RR-algebras. So in some sense caring about polynomials is equivalent to caring about commutative rings and, more generally, commutative algebras.

Non-linear polynomial regression uses a series of non-linear combinations of input variables to establish the relationship with the output, but this requires certain knowledge of the relationship between input and output. The training regression algorithm model generally uses stochastic gradient descent (SGD). Advantages: rapid modeling, effective for small data volumes and simple relationships; linear regression models are very easy to understand, which is conducive to decision analysis.

Report Q2: Write down the linear algebra used to represent the spline fitting problem and the solution. Cite the key formula in the report in the relevant part of your code. [6]

Polynomial equations are equations that contain polynomial expressions on both sides of the equation. Here's the standard form of a polynomial equation.

$$a_n x^n + \ a_{n-1} x^{n-1} + \ldots + a_1 x + \ a_0 = 0$$

Note that an, an-1, ... ao can be any complex numbers for and the exponents can only be whole numbers for these to be considered as polynomial expressions.

```
function [u_x,u_y,u_z] = evaluate(x,y,z,control,alpha,sigma)

  [K_x,K_y,K_z] = RBFSpline.kernel_gaussian(x,y,z,control,sigma);

     K_x = alpha(1,:) .* K_x;
       K_y = alpha(2,:) .* K_y;
       K_z = alpha(3,:) .* K_z;

       u_x = x' + sum(K_x,2);
       u_y = y' + sum(K_y,2);
       u_z = z' + sum(K_z,2);
   end
```

Report Q3: What is the best linear algebra algorithm should be implemented to solve this spline fitting problem and why? [1]

Regression splines is one of the most important non linear regression techniques. In polynomial regression, we generated new features by using various polynomial functions on the existing features which imposed a global structure on the dataset. To overcome this, we can divide the distribution of the data into separate portions and fit linear or low degree polynomial functions on each of these portions.

The best linear algebra algorithm should be implemented to solve this spline fitting problem is One of the most common piecewise functions is a Step function. Step function is a function which remains constant within the interval. We can fit individual step functions to each of the divided portions in order to avoid imposing a global structure. Here we break the range of X into bins, and fit a different constant in each bin.

Because Binning has its obvious conceptual issues. Most prominently, we expect most phenomena we study to vary continuously with inputs. Binned regression does not create continuous functions of the predictor, so in most cases we would expect no relationship between the input and output.

Report Q4: What are the control points here? Can we choose any points as control points at evaluation stage, and why? [2]

Moving control points is the most obvious way of changing the shape of a B-spline curve. The local modification scheme discussed on an earlier page states that changing the position of control point Pi only affects the curve C(u) on interval [ui, ui+p+1), where p is the degree of a B-spline curve. In fact, the shape change is translational in the direction of the control point being moved. More precisely, if control point Pi is moved in certain direction to a new position Qi, then point C(u), where u is in [ui, ui+p+1), will be moved in the same direction from Pi to Qi. However, the distance moved is different from point to point. In the following figures, control point P4 is moved from the position in the left figure to a new position in the middle figure and finally to its final position in the right figure. As you can see those points corresponding to knots (marked with little triangles) are moved in the same direction.

Report Q5: Do we need the weighting parameter lambda at evaluation stage, and why? [2]

The exact form of the penalty matrix Ω is actually not so important. What should pay attention to is that there is an extra term λβT Ωβ in compared to the usual criterion for regression splines; this is called a regularization term, and it has the effect of shrinking the components of the solution ˆβ towards zero. The parameter λ ≥ 0 is a tuning parameter, often called the smoothing parameter, and the higher the value of λ, the more shrinkage.

Report Q6: Describe the details of your vectorisation strategies for kernel computing for large point sets. [2]

One common approach to low-rank spline smoothing (e.g. Ruppert et al., 2003) is to use K ≪ n knots and choose the κk to 'mimic' the x is. A simple strategy is to draw a random sample of size K from the x is. Alternatively, one can use deterministic rules that aim to somehow 'fill the space' of the x is. For one-dimensional fitting (d = 1)taking κk ≃ (k/K)the sample quantile of the unique xi's achieves this aim. For higher dimensions distance-design algorithms such as those used by Nychka and Saltzman (1998)can be used. Let D be a subset of observed points xi called design points and C be a subset of observed points xi called candidate points with $D \cap C = \varphi$.

Report Q7: Discussion of the utility of the Gaussian kernel parameter, sigma

Due to the central limit theorem, the Gaussian can be approximated by several runs of a very simple filter such as the moving average. The simple moving average corresponds to convolution with the constant B-spline (a rectangular pulse), and, for example, four iterations of a moving average yields a cubic B-spline as filter window which approximates the Gaussian quite well. A moving average is quite cheap to compute, so levels can be cascaded quite easily.

Borrowing the terms from statistics, the standard deviation of a filter can be interpreted as a measure of its size. The cut-off frequency of a Gaussian filter might be defined by the standard deviation in the frequency domain yielding

Report Q8: What is a reasonable approach to randomly displace the control points, (e.g. what distribution the random transform need to draw from and is there any constraint needed), such that the resulting transformation represented by the moved control points are considered biophysically reasonable. [2]

The two key choices are in the number and spacing of the knots and the use (or not) of a penalty function, e.g., the integrated second derivative of the spline. When there is no penalty, the creation of the transformed variables can be done separately and the new variables are simply included in a standard model fit; no modification of the underlying regression procedure is required. This approach is often referred to as regression splines; the flexibility of the resulting non-linear function is entirely a function of the number of knots. The inclusion of a smoothing penalty, on the other hand, requires modification of the fitting routine in order to accommodate it. This has to be included in each regression function separately. The resulting smoothing splines have several desirable properties, but the added complexity of the smooth function can be a reason for not been used more often in applied settings.

Report Q9: Does it mean that the interpolated voxel coordinates, driven by the moved control points, represent biophysically plausible deformation? [2]

The dense deformation field $\vec{v}$ is parameterized by a sparse set of control points, which are uniformly distributed throughout the fixed image’s voxel grid. This results in the formation of two grids that are aligned to one another: a dense voxel grid and a sparse control point grid. In this scheme, the control point grid partitions the voxel grid into many equally sized regions called tiles. A spline curve is a type of continuous curve defined by a sparse set of discrete control points. Generally speaking, the number of control points required for each dimension is n+1, where n is the order of the employed spline curve.

Report Q10: Describe in details of the steps taken to compute a warped 3D image. [4]

Image editing

Given a contour of a subject, artists might want to warp/deform the subject in some way by moving control points along the contour

Texture mapping

Given two sets of facial landmarks in the input image and the texture space, we can warp the input image to the texture so that it could be displayed in 3D.

Image morphing

Given two sets of facial landmarks in two images, one face could be warped to make it look like the other face.

image is taken from google

Radial Basis Function interpolation

Before introducing the Thin Plate Spline warping algorithm, we will quickly go through the radial basis function interpolation, which is the general form of the thin plate spline interpolation problem. We will just focus on explaining the 1D example introduced in the last section.

Report Q11: Based on the visualisation, what can you observe to support or update your answer to Q9?

Since we are working with cubic B-splines, we require 4 control points in each dimension, which results in 64 (43) control points for each tile. The deformation field at any given voxel within a tile is computed by utilizing the 64 control points in the immediate vicinity of the tile. Furthermore, because we are working in three dimensions, three coefficients (Px,Py,Pz) are associated with each control point, one for each dimension. Mathematically, the x-component of the deformation field for a voxel located at coordinates (x,y,z) in the fixed image can be described as

(2.1) $v\ x(x\rightarrow)=\sum i=03\sum j=03\sum k=03\ \beta\ i(u)\ \beta\ j(v)\ \beta\ k(w)Px(l,m,n)$,

where β (.) are the spline basis functions obtained as follows. Let Nx,Ny, and Nz denote the distance between control points in terms of voxels in the x,y, and z directions, respectively.

ReportQ12:Summarise the effect of the three changes, with example images included in the report. [6]

The thin plate spline method is often used to fit data in high dimensions. Standard thin plate splines require the solution of a dense linear system of equations whose size increases with the number of data points and can be expensive when used on large data sets. we present a spline method that uses polynomials with local support defined on finite element grids.

It produces smooth surfaces, which are infinitely differentiable.

There are no free parameters that need manual tuning.

It has closed-form solutions for both warping and parameter estimation.

There is a physical explanation for its energy function.