Finding Best Fit for Hand-Drawn Curves using Polynomial Regression

Bhaumik Choksi
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

Ajay Venkitaraman
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

Swati Mali
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

ABSTRACT
Curve fitting gives the user a mathematical function that best fits to a series of data points while considering the constraints of the data. This paper presents an algorithm to determine the equation of a hand-drawn curve using polynomial regression. The hand-drawn curve may be digitally drawn, or manually drawn on paper and scanned. Polynomial regression is used to estimate the order of the equation that fits the curve and determine the coefficients of the equation.

General Terms
Regression, Machine Learning, Curve Fitting, Hand-Drawn, Image Processing.

Keywords
Polynomial Regression, Regression, Curve Fitting.

1. INTRODUCTION
Curve fitting is a technique that is used to determine a mathematical equation that fits a given set of data points in such a way that the deviation of the points from the equation is minimized. In order to fit two-dimensional data, polynomial regression is used. The factor that is being predicted (the factor that the equation solves for) is called the dependent variable. The factors that are used to predict the value of the dependent variable are called the independent variables [1].

Curve fitting has been widely used for many research problems over the years, but the problem of curve fitting has been looked at mainly from a mathematical perspective, where analysis of hand-drawn curves hasn't been given much importance. This was the motivation for this paper's work, so as to implement an efficient curve fitting technique which analyses curves drawn on the console.

This paper discusses a technique to extract the data points from the image, which are the two dimensional coordinates of the points in the curve. Using the data captured, polynomial regression is used to fit the curve to an equation in such a way that the error is minimized and at the same time, overfitting is avoided.

Section 1 gives the introduction to the problem identified while section 2 introduces the readers to the basic elements of polynomial regression and issues associated with it. Section 3 explains the proposed methodology and implementation and section 4 is used for results. Section 5 concludes the paper while section 6 discusses the future scope.

2. POLYNOMIAL REGRESSION: BASIC CONCEPTS AND ISSUES
Polynomial regression is a regression technique which is used to model the relationship between a dependent variable (denoted by y) and an independent variable (denoted by x) to a polynomial over variable x in degree n. A polynomial regression equation of degree n can be represented using the following equation:

$$y = a_0 + a_1x^1 + a_2x^2 + \cdots + a_nx^n$$

where y is the dependent variable, x is the dependent variable and a0 … an are the regression coefficients. It can be seen that polynomial regression is a special case of multilinear regression. This is because y is still dependent on powers of the same dependent variable x (it should not be confused with the literal meaning of being polynomial where multiple independent variables come into picture). For least square analysis, polynomial regression can be performed using the techniques of multiple regression. This is done by treating each power of the independent variable i.e. x as an independent variable during multiple regression. Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem it is linear, in the sense that the regression function is linear in the unknown parameters that are estimated from the data [10]. It is evident that a higher degree polynomial will fit any given data better. However, this may also lead to overfitting. If the sample size is extremely small, the curve may produce highly inaccurate predictions during testing and validation.

Polynomial models have a shape/degree tradeoff. In order to model data with a complicated structure, the degree of the model must be high, indicating that the associated number of parameters to be estimated will also be high. This can result in highly unstable models [10]. The first degree polynomial equation could also be an exact fit for a single point and an angle while the third degree polynomial equation could also be an exact fit for two points, an angle constraint, and a curvature constraint. Several other combinations of constraints are possible for these and for higher order polynomial equations.

3. PROPOSED METHODOLOGY AND IMPLEMENTATION
The proposed system offers the user a graphic UI wherein the user can input the data in the form of an image and performs various pre-processing tasks before extracting data points from it, since the image may contain noise or background data which is unnecessary. Then, the data points are extracted from the image using the pixel intensity values. These points are fit into a polynomial regression model of varying degrees, in order to determine the error, and therefore, the equation with the best fit.

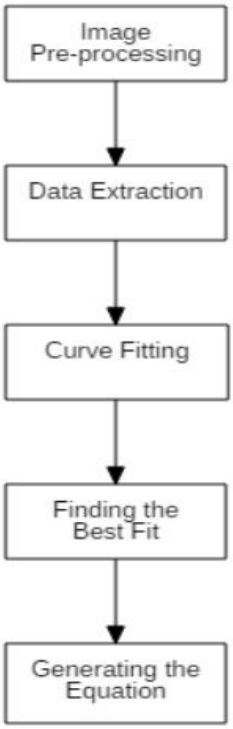

Fig 1: Flowchart for the proposed algorithm

3.1 Image Pre-processing

Pre-processing is required since the input image may contain distortions, noise and background data which is useless for this application. The first step is to convert the image into grayscale form, in order to simplify further processing.

Once the grayscale image is obtained, a Laplacian mask is applied to it, in order to highlight the edges, which are high frequency components. The hand-drawn curve will act as a continuous edge. The Laplacian mask will eliminate low-frequency background components. The resultant image will contain the highlighted curve, with little to no background data.

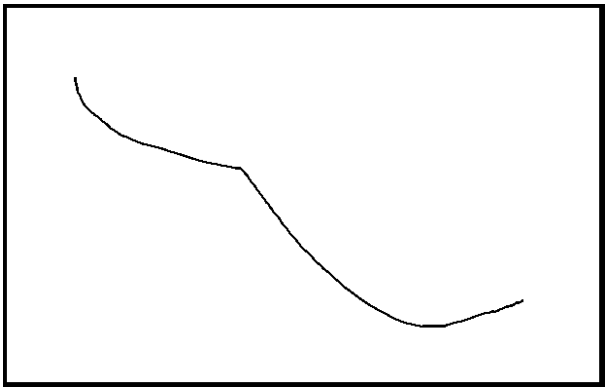

Fig 2: Original image of the curve

3.2 Data Extraction

The pre-processed image data is then read into a matrix. Every value in the matrix will correspond to either of the two brightness levels, black, or white. The points on the hand-drawn curve will be white, and the remaining pixel values will be black. Hence the matrix forms the representation of the pixel values of the image. The coordinates of the image must match the coordinates of the Cartesian plane, where the origin (0,0) lies in the bottom left corner.

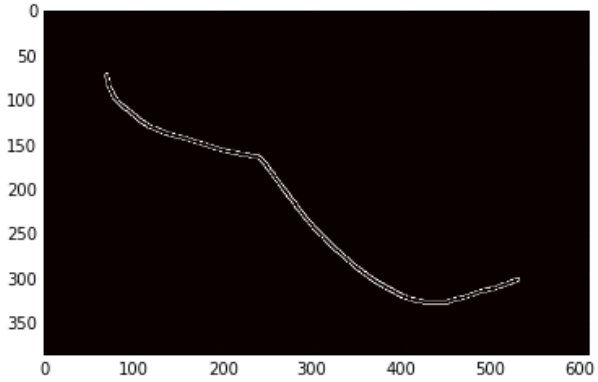

Fig 3: Image of the curve after Pre-Processing

Now, all the values in the matrix are scanned, in order to extract pixel coordinates that are set to white during the process of edge detection. While scanning, if a white pixel is encountered, the row and column value of the pixel, which correspond to the x and y coordinates of the point, are stored as part of the training data. Any black pixel, which corresponds to the background, is ignored.

If the image resolution is very high, the number of training points obtained will be high too. The number of points may be reduced in order to speed up the process of polynomial regression and reduce the training time.

3.3 Curve Fitting

Once all the points are obtained for training, some of these points must be set aside for testing, after the fit is complete. This is done because the data obtained from the image of the curve is limited. After the training and testing sets are ready, one can proceed with the curve fitting process using polynomial regression.

A polynomial regression model is trained using the training data points obtained from the image. The points are fit into equations of degrees from 1 to n, where n depends on the user and the type of fit required. Once training is complete, the model is tested with the testing dataset of points in order to determine the error. The errors for every degree are stored along with their corresponding degrees in a database which will be used to determine the best fit.

3.4 Finding the Best Fit

Since the data used for fitting is constrained by the resolution of the image, the probability of overfitting increases manifold. In order to avoid this, higher degree fits are penalized so as to get a more generalized fit.

$$cost = e \times d$$

where e is the mean squared error obtained during testing, d is the degree of the polynomial.

The mean squared error is given by the formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y - \bar{Y})^2$$

where Y is the expected output, $\bar{Y}$ is the output of the model and MSE is the mean squared error.

This cost is used to calculate the best fit. The degree corresponding to the minimum cost is chosen as the best fit and to determine the coefficients for the equation of the curve.

Table 1. Corresponding Mean-Squared Errors (MSE) for each degree of equation for the given curve

Sr. No.	Degree	Total *MSE*	(Degree) × (Total *MSE*)
1	1	493.47	493.47
2	2	403.75	807.50
3	3	143.59	430.77
4	4	104.06	416.24
5	5	23.22	116.10
6	6	40.97	245.82

As is evident from the results, the cost is minimum for the equation with degree 5, since the cost reaches its local minima at degree 5. Hence, the equation generated for the curve will be of the degree 5.

3.5 Generating the Equation

The intercept and the coefficients obtained through the process of fitting are mapped to the corresponding polynomial terms. Depending on the implementation, the equation will be represented either in string format or a character array. The equation will be of the form:

$$y = a_0 + a_1x^1 + a_2x^2 + \cdots + a_nx^n$$

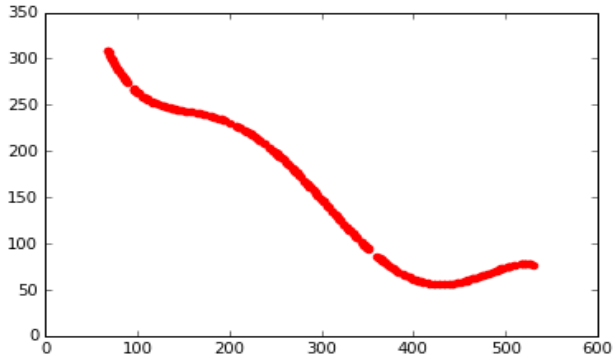

Fig 4: Plot of the equation predicted by the model

4. RESULTS

In order to determine the accuracy of the model, its output was calculated for curves whose equations were known beforehand (see Table 2). Since the equation also depends on the placement of the curve in the image, the results vary accordingly. Moreover, the image coordinates start at the origin, but the curve never intersects the axes because of the way the image is scanned, the curve is assumed to be in the first quadrant of the Cartesian plane.

Table 2. Actual equation and predicted equation by fitting for some sample curves

Sr. No.	Actual equation	Equation of curve obtained by fitting
1	$y = x + 100$	$y = x + 102$
2	$y = 0.01x^2 + 0.01x + 100$	$y = 0.01x^2 + 0.1x + 50$
3	$y = 0.0001x^3 - 0.01x^2 - 0.1x + 100$	$y = 0.000093x^3 - 0.0067x^2 - 0.34x + 100.05$

Since the image is scanned pixel-by-pixel, every pixel corresponds to unit distance. Hence, it is possible to obtain different equations for the same curve, depending on the resolution of the image.

Fig 5: Image of the curve used as input

Therefore, the evaluation cannot be done by directly comparing the equations. It is done by comparing the shapes of the given curve and the curve of the equation predicted by the model.

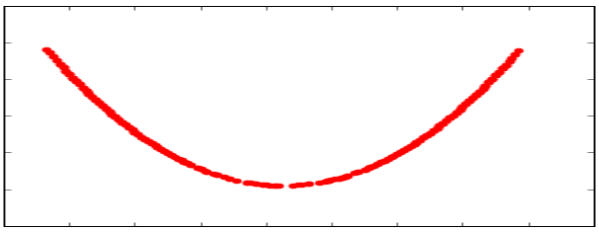

Fig 6: Plot of the equation predicted by the model

5. CONCLUSION

The proposed technique successfully identified the degree of the polynomial represented by the curve from the image and also correctly estimate its equation. The equation was subject to the resolution of the image provided and the positioning of the curve in the image. The curve of the predicted equation was visually similar to the given curve, and had the same degree as the given curve.

6. FUTURE SCOPE

The proposed model is capable of identifying the shape and degree of curves where the relationship between y and the various powers of x is linear. However, there are other two-dimensional curves such as the sine curve, or circles, which exhibit a nonlinear relationship between the x and y variables. Since this model is based on polynomial regression, which is a form of linear regression, it cannot identify such non-linear relationships.

7. REFERENCES

[1] Gupta, Swati. "A Regression Modeling Technique on Data Mining." International Journal of Computer Applications 116.9 (2015).

[2] Seber, George AF, and Alan J. Lee. Linear regression analysis. Vol. 936. John Wiley & Sons, 2012.

[3] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." IEEE transactions on systems, man, and cybernetics 9.1 (1979): 62-66.

[4] Lancaster, Peter, and Kestutis Salkauskas. Curve and surface fitting: an introduction. Academic press, 1986.

[5] Akima, Hiroshi. "A new method of interpolation and smooth curve fitting based on local procedures." Journal of the ACM (JACM) 17.4 (1970): 589-602.

[6] Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. Introduction to linear regression analysis. John Wiley & Sons, 2015.

[7] Weisberg, Sanford. Applied linear regression. Vol. 528. John Wiley & Sons, 2005.

[8] Motulsky, Harvey J., and Lennart A. Ransnas. "Fitting curves to data using nonlinear regression: a practical and nonmathematical review." The FASEB journal 1.5 (1987): 365-374.

[9] Bates, Douglas M., and Donald G. Watts. Nonlinear regression analysis and its applications. Vol. 2. New York: Wiley, 1988.

[10] Peckov, Aleksandar. A Machine Learning Approach to Polynomial Regression. Diss. PhD thesis, Jozef Stefan International Postgraduate School, Ljubljana, 2012.

[11] Gallant, A. Ronald, and Wayne A. Fuller. "Fitting segmented polynomial regression models whose join points have to be estimated." Journal of the American Statistical Association 68.341 (1973): 144-147.