Curve fitting

Project by Daria Fitisova

Contents:

Introduction

Many physical phenomena of the everyday surroundings deemed to be continua, although our quantitative measurements of them are discrete. From such discrete values we often try to reconstruct the continuum in order to analyze its features. Moreover we assume that the phenomenon is not standalone: it depends on some underlying variables (the so-called independent variables). Here arises the need to construct a curve or mathematical function (which will include independent variables) that has the best fit to the data points. This process is called *curve fitting* [4, p.4].

As far as mathematical description of any continuous process, its model identification seem to be general targets which are common for many scientific fields, curve fitting is highly advantageous practically in any sphere where people have to deal with the data. Population, epidemiology, environmental toxicity data analyses are just several examples of the fields of its implementation [1].

Generally speaking, curve fitting pursues three main aims:

1. data visualization (trend spotting),
2. curve representation (getting the values of a function where no data are available),
3. forecasting.

We should mention here that the data visualization seems to be rather standard curve fitting application being preceding and necessary procedure for both approximation (2) and forecasting (3): "it is helpful to plot the data first for understanding the type, pattern and trend of the random data sets" [7, p.8]. The least two objectives are mostly independent of each other, determining therefore the curve fitting technique which should be used in one or another case.

This 'duality' is hold in the current paper: we cover least squares approaches which are used for both approximation and forecasting, splines and Taylor approximation which are appropriate for curve representation only. Every section is based on theoretical formalization of the method and provided by the empirical example, simulated in Matlab or R-package. The same data is used for all the made models and described in the following section.

Data description

As an example we took data series of Usain Bolt's 100m world record (WR) kinetics, which was set on the World Championship 2009 in Berlin[1]. Namely, we have used video-script of speed dynamics during the whole distance. Here and below we take speed (km/h) for dependent variable and distance (m) for independent one.

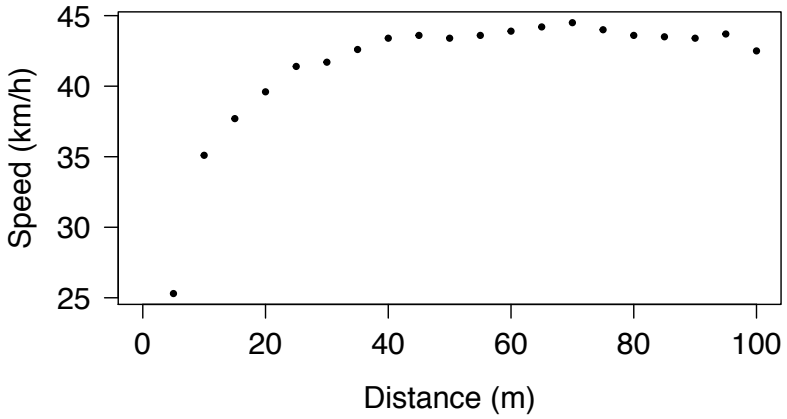

Fig.1. Usain Bolt's WR speed dynamics

One can notice that the speed gain is very strong during the first half of the distance. In the interval 40-80 mm. the line is quite stable, but there's a little spike and further decline on the last several meters.

In the following sections we will try to approximate this series and forecast the possible speed if the distance is longer.

Least squares approach

It was said above that curve fitting is a construction of a curve of mathematical function which 'fits' the data best. But what does it mean 'to fit'? It is mentioned in [2, p.287] that the most common approach to 'best fit' is *the method of least squares* (LS). It is based on the desire to minimize the difference between the given data points and the approximating function. We will introduce the least squares approach through the example of linear regression model.

Linear function

Lets suppose that we could simulate a line corresponding to the general direction of the data cloud:

$$y_i = \alpha + \beta x_i + \varepsilon_i \ , i = 1, \dots, n$$

[1] The data was taken from URL: https://www.youtube.com/watch?v=SyY7RgNLCUk

here α is an intercept, β - the slope coefficient An error term (ε) is the deviation of the estimated line from the real data points. It can also be expressed mathematically:

$$\varepsilon_i = y_i - \widehat{y_i} = y_i - (\alpha + \beta x_i)$$

But there are several advantages to using the square of the difference at each point [2, p.290], because:

1. Greater errors are weighted more heavily and small errors – lightly.
2. Positive differences do not cancel negative differences.

Therefore,

$$\varepsilon_i^2 = (y_i - \widehat{y_i})^2 = (y_i - (\alpha + \beta x_i))^2$$

According to the logic described above, the 'best fit' could be achieved by minimizing the sum of squared errors, which could be done by taking the derivatives of the error with respect to α and β, setting each to zero:

$$\frac{d\varepsilon_i^2}{d\alpha} = -2\sum_{i=1}^{n}\left(y_i - (\alpha + \beta x_i)\right) = 0$$

$$\frac{d\varepsilon_i^2}{d\beta} = -2\sum_{i=1}^{n} x_i\left(y_i - (\alpha + \beta x_i)\right) = 0$$

Solving the equations with respect to α and β we can put the result into the matrix form:

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} * \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

Therefore, the rule for finding the coefficients of the 'best fit' line could be expressed as follows:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2$$

Nevertheless the fact that the given data curve is doubtlessly nonlinear, we have simulated the least squares linear regression for educational purposes :

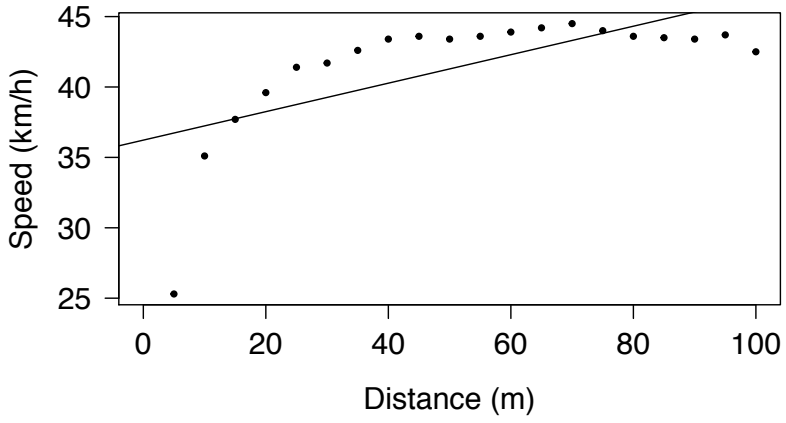

Fig.2. Usain Bolt's WR speed dynamics and its LS linear approximation

Table 1

Linear LS model summary

	Estimate	**Std. Error**	**t value**	**Pr. (>|t|)**
Intercept	36.23	1.60	22.57	0
distance	0.10	0.03	3.77	0

According to the information from the table 1, we got an equation $\hat{y}$=36.23+0.1*(distance), but from the figure 2 it comes obvious that this approximation is inappropriate as far as the error terms are big (the line underestimates and overestimates the given data).

Later we will discuss the formal methods of models evaluation, but now we'd like to consider logarithmic LS function.

Polynomial: quadratic, cubic and higher order functions

The procedure of simulating the LS polynomials with order two and higher is the same as it was by the linear model. By taking the derivatives with respect to every coefficient and setting it to zero one can get a matrix-form equation. Solving the least lets to compute LS-regression coefficients.

We have computed coefficients for quadratic, cubic, forth order polynomials:

Table 2

Quadratic, cubic, 4-th order LS models summary

		Estimate	**Pr. (>|t|)**
Quadratic LS model	Intercept	29.22	0
	distance	0.48	0
	distance^2	-0.01	0
Cubic LS model	Intercept	24.27	0
	distance	0.99	0
	distance^2	-0.02	0
	distance^3	0	0
4-th order LS model	Intercept	19.57	0
	distance	1.724	0
	distance^2	-0.05	0
	distance^3	0	0
	distance^4	0	0

From the table 2 one can notice that all the coefficients are significant, that means that using the information above we can compute 2-, 3- and even 4-order polynomial, which will have the 'best fit'. Moreover, we have computed 5-, 6- order polynomials (not mentioned in the table), which

also have the significant coefficients. Is it always good or not would be discussed in the following sections.

As it can be seen from the figure 3 (below), the more is the order of polynomial, the closer is the line to the real data curve. But it's not always the case. If we take the polynomial of order much higher than five, we would probably also face the problem of overestimation [7, p.15]. What does the overestimation mean? It is 'over-doing' the requirement for the estimation to match the data points. In other words, high order polynomials can perfectly match the real data points, making huge 'noise' by its squiggles. Model evaluation helps to understand by which polynomial order it is better to stop.

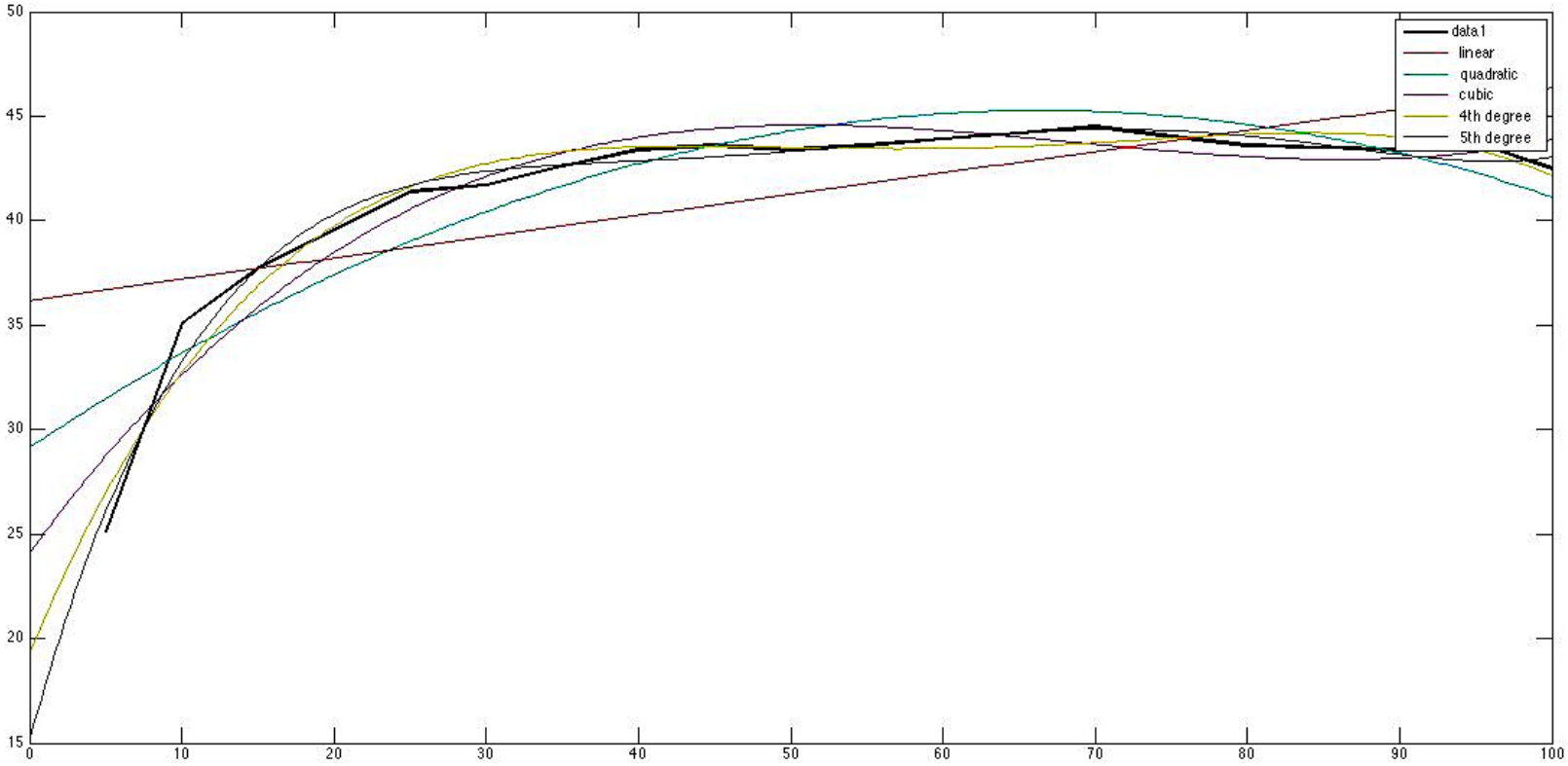

Fig.3 Usain Bolt's WR speed dynamics (black bold curve) and its LS polynomial approximations

Logarithmic function

Exponential and logarithmic function stay a little bit beyond the common logic of LS curve simulation. The thing is that when we take partial derivatives of sum of squares with respect to the coefficients we get nonlinear equations, which can not be solved as it was described above. The solution for such a problem is *to linearize* the equation. For instance, if we have an exponential function

$$y = Ce^{\beta x}$$

it is necessary to take logarithm of the bot sides and get rid of exponential:

$$\ln(y) = \ln(Ce^{Ax}) = \beta x + \ln(C)$$

which seems to be linear if y=ln(y), X=x, α=ln (C). Such a transformation lets to minimize the sum of squares and to find the appropriate coefficients.

We have computed logarithmic LS approximation to a given data.

Table 3

Logarithmic LS model summary

	Estimate	**Std. Error**	**t value**	**Pr. (>|t|)**
Intercept	23.24	2.30	10.10	0
distance (ln)	11.31	1.39	8.13	0

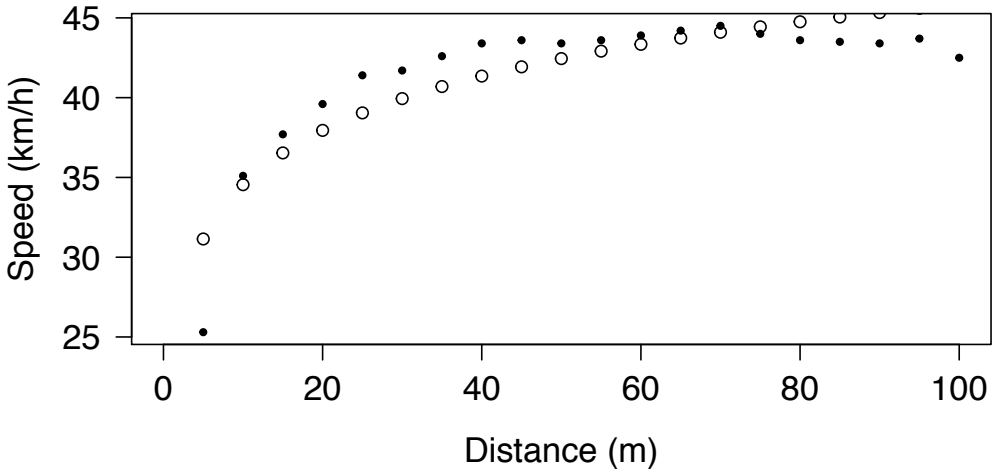

Fig.4 Usain Bolt's WR speed dynamics (black points) and its LS logarithmic approximation

As it follows from the table 3, all the coefficients are significant. But comparing the figure 4 with the figure 3 we see that there functions which have much better approximation *they are much closer to the real data curve). In order to understand: 1) is the logarithmic curve has the worse 'fit' than polynomials and 2) if yes, which polynomial is the best one, we should consider several methods of models evaluation.

Evaluation methods and forecasting

The aim to fit data in the best way is similar with the objective to explain as much real data variation as it is possible. In order to measure how much variance does the estimated model explain the coefficient of determination is used.

The coefficient of determination is R^2 [3, p.96]:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} = \frac{explained\ variation}{total\ variation}$$

The problem arise when we have to handle with polynomials as far as every extra added regressor makes R^2 bigger, but it does not necessarily mean that it explain extra variability in y. In other words, the coefficient of determination may be high even if irrelevant regressor are included. The solution to this problem is to use adjusted R^2 which neutralize this effect to some extent.

The adjusted coefficient of determination is R^2_{adj} [3, p.107]:

$$R^2_{adj} = R^2 - \frac{p(1-R^2)}{n-(p+1)}$$

Lets compare simulated models in terms of R^2 and R^2_{adj}:

Table 4

The comparison of the simulated models by the comparison of explained variation

	1st order	2nd order	3rd order	4th order	ln
R^2	0.44	0.81	0.92	0.97	0.79
R^2_{adj}	0.41	0.79	0.90	0.96	0.77

R^2=1 means that all the variation of real data was explained by the estimation. According to this criteria, the 4th order polynomial seems to be the 'best fit' among the models above.

As it was mentioned above, the crucial benefit of the least squares approach is that the outcome is a model using which we can both get the missing data and forecast. Using 4th order polynomial

$$y = 19.57 + 1.724 * (distance) - 0.04517 * (distance^2) + 0.0005083 * (distance^3) - 0.000002065 * (distance^4)$$

we can state that Usain Bolt's speed on the 92nd meter was 42.54 km/h. Moreover if we assume that the distance is longer than 100m, no other factors has an impact on his speed (like weariness, concentration, etc.) and the speed gain is got by the rule we have determined we can say that his speed on the 120th meter would be 23.7 km/h. Hopefully, no one runs more than 100 meters according to the speed function we got.

Taylor approximations

In this section we consider a method of approximating a function at a specific value of x. Unfortunately, this approach can be implemented just to represent the curve (no forecasting is possible). But the Taylor polynomial (of degree n) gives the highest possible order of contact between the function and the polynomial [2, p.316]. The Tailor polynomial at x=a is:

$$p(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

According to the formula above, the more is the order of the function, the more precise is the approximation. We have made several Taylor approximations for a cubic model setting extension point a=0:

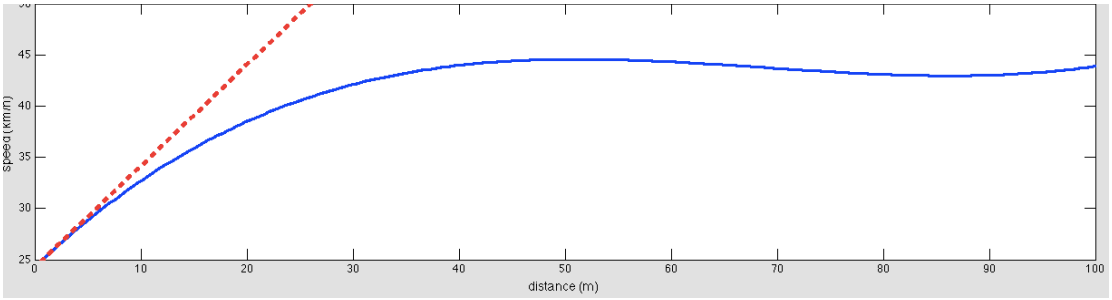

Fig.5 Taylor approximation (n=1, a=0)

$$T(x) = 0.988 * x + 24.27$$

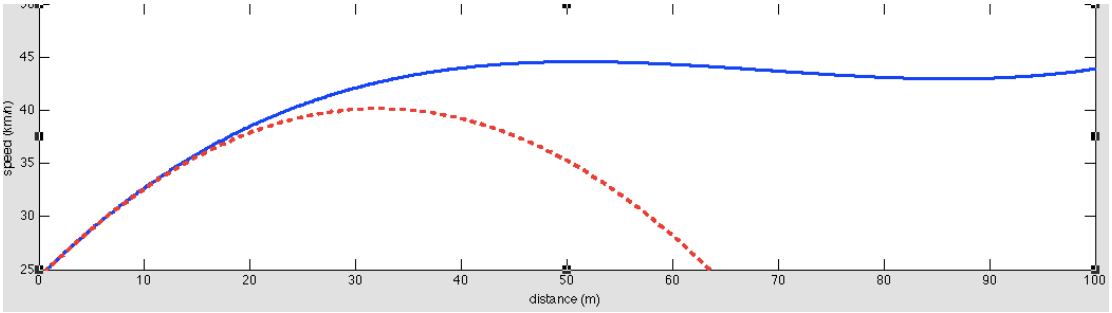

Fig.6 Taylor approximation (n=2, a=0)

$$T(x) = 0.988 * x - 0.1539x^2 + 24.27$$

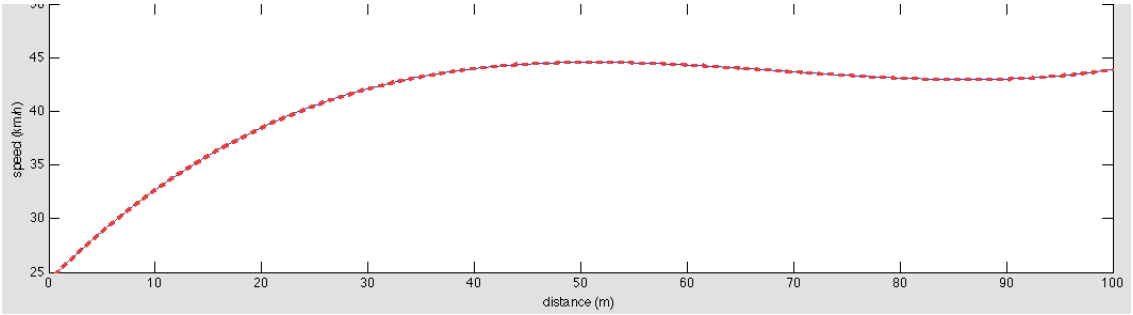

Fig.7 Taylor approximation (n=3, a=0)

$$T(x) = 0.988 * x - 0.1539x^2 + 0.00007464x^3 + 24.27$$

We should conclude that according to the results of Taylor approximation we got the input cubic function. But we also got a quite precise quadratic approximation on the interval [0;15] and at x=0.

We suppose that Taylor approximations could be very efficient if one has to handle with exponential or logarithmic curve as far as such computations are nonlinear and difficult to solve 'by hand'. The prospective to get approximated linear function instead of exponential or logarithmic seems to be desirable.

Taylor expansion approach seems to be very useful when one has to get missing values which are not represented in the real data series. Moreover, the calculation of least would be comparably easy as far as Taylor function is a polynomial and would just little (if it would) differ from the real data because of the approximation precision.

Splines

Spline is a numeric function that is piecewise-defined by polynomial functions, and which possesses a high degree of smoothness at the places where the polynomial pieces connect (which are known by knots.) [6]. Respectively, the more the number of knots, the more precise is the approximation.

We've simulated several splines for the real data series:

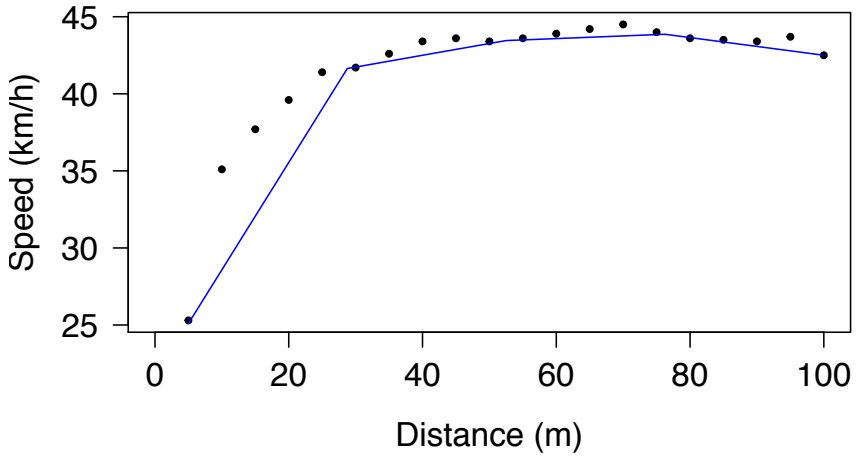

Fig.8 Splines with 5 knots

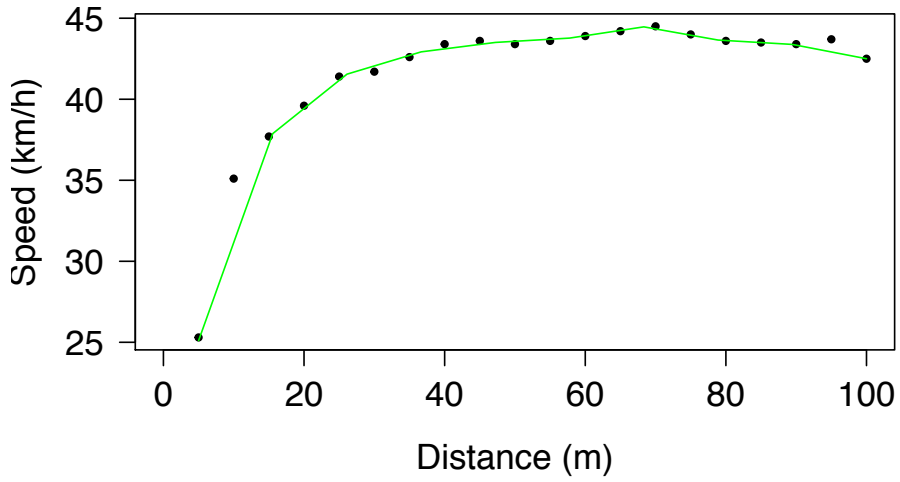

Fig.8 Splines with 10 knots

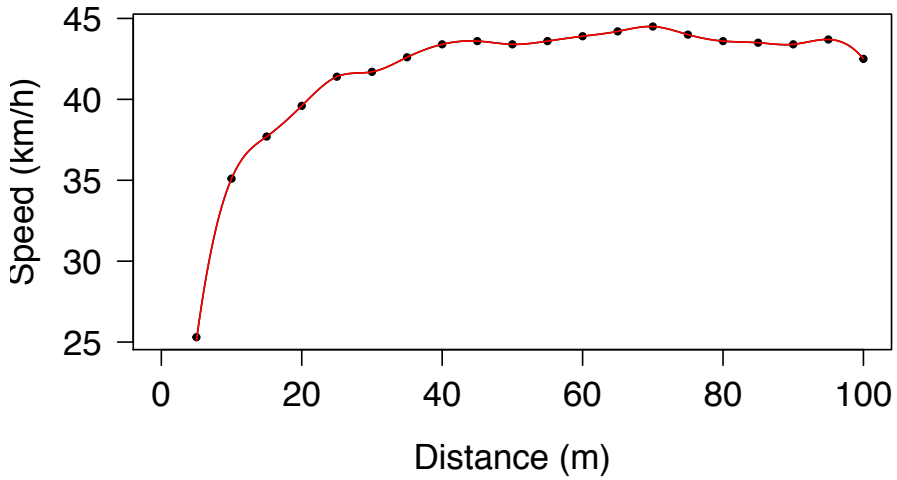

Fig.8 Splines with 400 knots

There is a crucial benefit of splines implementation in R. The package computes not just approximation, but also calculates most of the X-Y dependencies. Such an option could be extremely useful for curve representation. For instance, from the data set we know that y equals to 25.1 corresponds to the x equals to 5. Due to the splines approximation (n=400) we can also say that with the speed of 25.84 or 42.73 km/h Usain Bolt was overgoing the distance of 5.23 and 99.52 meters respectively.

Conclusion

In the current paper the least squares, Taylor expansion and splines approaches were introduced as the basic ones. We should mention, first of all, that the list of curve fitting techniques

is much wider than it was covered here. Choosing the approach to fit the real data series one should start from answering the question: what is the aim of the fitting? The are two main options available: to forecast or to represent the curve. If one has to deal with the least problem, Taylor approximation and splines functions seem to be more precise, easier to implement and more 'intelligent' so to say. Unfortunately, they are not suitable for making forecasts. But at this area least squares technique shows itself pretty good.

Bibliography:

1. Arlinghaus, S.L. (1994). *Practical handbook of curve fitting*. Boca Raton: CRC Press.
2. Fausett , L.V. (1999). *Applied numerical analysis using Matlab*. New Jersey: Prentice Hall.
3. Härdle, W., & Simar, L. (2014). Applied multivariate statistical analysis. Heidelberg: Springer.
4. Lancaster, P., & Šalkauskas, K. (1986). *Curve and surface fitting: an introduction*. London: Academic press.
5. Tarter, M.E., & Lock M.L. (1993). *Model-free curve estimation*. New York: Chapman and Hall.
6. Wikipedia. URL: https://en.wikipedia.org/wiki/Spline_(mathematics)
7. Zheng, X., & Gong P. (1997). Linear feature modeling with curve fitting: parametric polynomial techniques. *Geographic information sciences*, 3, 7-19.