Decision Analytics

– SEMINAR Summer Semester 2015 –

Generalized Additive Model, LOWESS & Kernel Regression

– SEMINAR PAPER –

Submitted by:
Julius Gordon

Advisor:
Stefan Feuerriegel

1. Table of Contents

1. Introduction

This paper will give an introduction to three non-parametric methods for data analysis and evaluation. These methods are the *Generalized Additive Model*, *Locally Weighted Scatter Plot Smoothing* and *Kernel Regression*. To garner an appreciation and therefore an understanding of why and when a non-parametric evaluation method is required, it is important to begin with the assumptions and thus limitations of the standard parametric model of linear regressions. Firstly, the estimation of a parametric regression requires an underlying theory to provide a strong functional linkage between the covariates. Secondly, the user must make an assumption regarding the underlying distribution of the population and of the error term. Finally, while there are tools to deal with non-linear relationships in the standard linear model, for example polynomials, these methods still require a presumption regarding the functional form of the model. Figure 1 illustrates an example of a non-linear data generating process and the difficulty of fitting a linear *Ordinary Least Squares* (OLS) model.

The methods introduced in this paper require none of the assumptions and thus do not have the limitations of the standard parametric linear model. All three concepts are highly adaptive methods that can be easily applied to evaluate non-linear and non-parametric problems in order to find the true form of the data generating process. Non-parametric methods require no theoretical basis for the functional form of the model or assumptions regarding the underlying distribution. The results are data driven, as the aim is to estimate the best fit based upon the data itself. Figure 1 illustrates a simple comparison between the fitted values of the linear OLS model and the fitted values of the non-parametric Kernel Regression model.

The Generalized Additive Model extends the linear model by allowing for both parametric and non-parametric covariates. This allows the user to easily incorporate both linear and non-linear relationships between the variables of interest. Locally Weighted Scatter Plot Smoothing uses a locally weighted regression to fit a non-parametric smooth function to the data. Finally the Kernel Regression method estimates the conditional expectation and fits a non-parametric smooth function using kernel functions to compute the unknown probability density function.

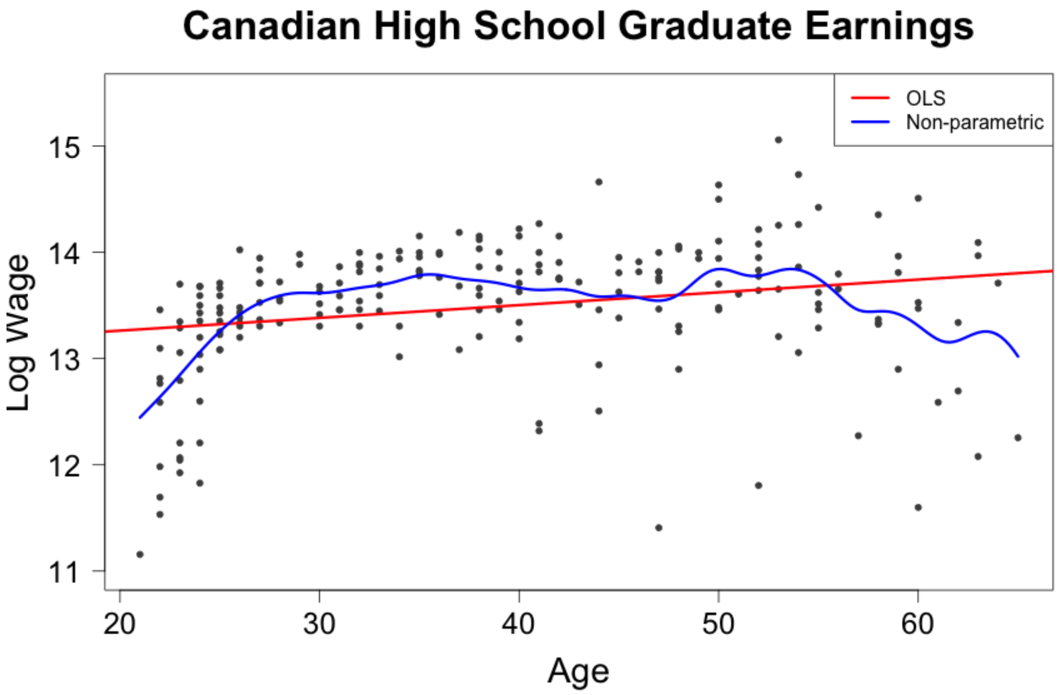

Figure 1: The dataset cps71 plotted with both a linear and a non-parametric estimation for comparison.

Commands in R will be listed in red and the output in blue.

This paper provides an introduction into the three different non-parametric evaluation techniques. Section 2 introduces the Generalized Additive Model, the algorithm used to fit the model and an example of the application of the method in R. Section 3 introduces the LOWESS method, the algorithm and an example of its application in R. Finally Section 4 introduces the Kernel Regression method via an explanation of the Kernel Density Estimator; again, an example of its application in R is also provided. The dataset cps71 was used for the application of each of the methods; it is pre-packaged with the software package R. It is a cross-sectional dataset of Canadian high school graduate earnings. The variables are logwage and age. There are 205 observations in total.

2. Generalized Additive Model

The Generalized Additive Model (henceforth GAM) is a natural extension of the linear generalized model that allows non-linear relationships between the dependent and independent variables. The key feature of GAM is that it is additive. This allows the user to combine parametric and non-parametric covariates while still being able to conduct inference and interpret individual effects in a ceteris paribus manner.

2.1 Method

GAM is an extension of the linear generalized model. To explain the concept let us begin with the traditional linear model that has the following structure

$y_i = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n + \varepsilon_i$.

The idea of the GAM method is to replaces the linear component $\beta_i x_i$ of the model with a smooth non-linear function $f_n(x_n)$. The resulting functional form may appear like this

$y_i = \beta_0 + f_1(x_1) + \ldots + f_n(x_n) + \varepsilon_i$.

The conditional mean denoted $\mu(x)$ of the response variable is related to the independent variables via the link function $g(\mu(X))$. The link function depends on the error distribution of the response variable and can be selected accordingly from the exponential family of sampling distributions. Three common link functions (as taken from Hastie et al. 2008) are the gaussian $g(\mu) = \mu$, the binomial $g(\mu) = logit/probit(\mu)$ and the poisson $g(\mu) = \log(\mu)$.

The function $f_i(x_i)$ is estimated non-parametrically and thus automatically reveals the degree of non-linearity in x_i. The covariates may take multiple forms including parametric, semi-parametric and non-parametric. This allows the combination of qualitative, linear and non-linear variables. The following are examples of different potential functions forms of GAM method given by Hastie et al. (2008):

$$g(\mu) = X^T\beta + \alpha_k + f(Z)$$

where X is a linear vector of predicators, α_k is a qualitative factor and Z is a non-parametric covariate. The second example given by Hastie et al. (2008) is

$$g(\mu) = f(X) + g_k(Z)$$

where $g_k(Z)$ is an interaction term between a qualitative and non-parametric covariate. The final example given by Hastie et al. (2008)

$$g(\mu) = f(X) + g(Z, W)$$

where the term $g(Z, W)$ is a non-parametric function in two features.

2.1.1 Fitting the Model

There are multiple options to estimate the function f_i, the default generally being a scatter plot-smoothing algorithm. The LOWESS method (see section 3) is just one variation. Smoothing can estimate a non-parametric fit to non-linear data based on the idea of estimating local regressions. This makes it very flexible, allowing its application to a range of data scenarios. Further, the level of smoothing required gives the user an insight into degree non-linearity of the data. The GAM output will list the degrees of freedom (henceforth df) for each variable which indicates the required amount of smoothing. The higher the degrees of freedom, the more non-linear the data; for example, *df=8* indicates that the data is highly non-linear while *df=1* indicates the data is linear. The final step is then to estimate simultaneously all the functions $f_i(x_i)$ in order to estimate the model $Y = \alpha + \sum_{1=i}^{p} f_i(x_i) + \varepsilon$ using the backfitting algorithm (Hastie et al. 2008).

2.1.2 The Backfitting Algorithm

The algorithm applies a cubic smoothing spline S_i to the targets $\{y_j - \hat{\alpha} - \sum_{k\neq j} \hat{f}_k\left(x_{jk}\right)\}$ as a function of x_{ji} to obtain the estimate $\hat{f}_i$. It repeats this for each predicative variable, one after the other, using the current estimate of the other functions $\hat{f}_k$. This continues until the estimator is stabilised. The algorithm is adaptable and different smoothing techniques S_i can be implemented for example kernel methods, surface smoothers and periodic smoothers. The Backfitting Algorithm attempts to fit all the predicators, which, if there are many, is not feasible (Hastie et al. 2008).

The algorithm was developed by Leo Breiman and Jerome Friedman and is set out in Hastie et al. (2008).

1. Initialise: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i\,, \hat{f}_i \equiv 0, \forall i$

2. Cycle: i=1,2,…p,…1,2,…p until convergence is achieved.

$\hat{f}_i \leftarrow S_i\left[\{y_j - \sum_{k\neq j} \hat{f}_k\left(x_{jk}\right)\}_1^N\right]$, the backfitting step

$\hat{f}_i \leftarrow \hat{f}_i - \frac{1}{N}\sum_{i=1}^N \hat{f}_i(x_{ji})$, mean centring of the estimated function.

Continue until $\hat{f}_\mathrm{i}$ changes less than a predetermined amount.

2.2 Application in R

There are two packages available for installation to implement the generalized additive model in R. The first package `gam` was coded by Trevor Hastie and uses the Backfitting Algorithm outlined section 2.1.2. The second package `mgcv` developed by Simon Wood uses an alternative method of penalized splines with automatic smoothness selection. This section will illustrate how to implement GAM in R using the appropriate function call signs.

2.2.1 Function Call Sign

```
install.packages(“mgcv”)

library(mgcv)
```

The arguments for the code are the same for both methods

```
gam(formula, family=gaussian, data, method)
```

`formula` the regression function `response ~ predicators`. The selection of the smoothing method can be implement by `s` for smoothing splines or `lo` for loess. Additional smoothers can be added via interface function.

`family` gives a description of the error distribution and the link function to be used in the model. Options include `Gaussian, binominal` etc.

`data` the data set from which contains the selected variables.

`method` the method used to fit the parametric part of the model.

2.2.2 Example

The model is then estimated using the following command.

```
gam_2 <- gam(logwage ~ s(age), data=cps71)

summary(gam_2)
```

```
Family: gaussian
Link function: identity

Formula:
logwage ~ s(age)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.48988    0.03698   364.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
        edf Ref.df     F  p-value
s(age) 6.69  7.801 12.03 4.36e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.308   Deviance explained =   33%
GCV = 0.29129  Scale est. = 0.28036   n = 205
```

The code summary(gam_2) reveals the regression output shown above. The effective degrees of freedom(edf), reveals the degree of non-linearity. The predicator variable age has an edf of 6.69, this is further evidence that its effect is non-linear.

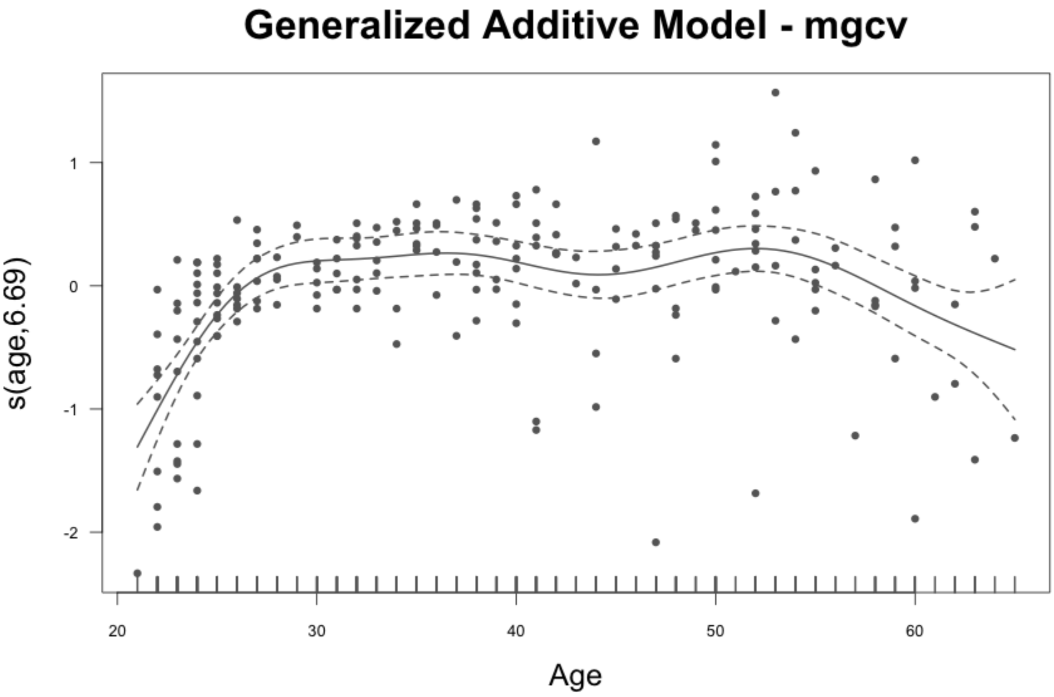

Figure 2: GAM estimated using the package mgcv with standard error bounds. The y-axis is the estimated degrees of freedom of the covariate age.

2.3 Summary

The Generalized Additive Model is a data driven approach that allows the user to easily incorporate non-linear relationships between covariates. The model replaces the linear component with a function computed using a smoothing algorithm. It retains the additivity and thus interpretability of the traditional linear model. However it tends to have a significant advantage in predictive power when compared with the linear model as it is able to combine parametric and non-parametric terms. The downside of additivity is that all predictors need to be added manually. If a model has large numbers of predicators important interactions may be missed (James et al. 2013).

3. Locally Weighted Scatter Plot Smoothing

Locally Weighted Scatter Plot Smoothing (henceforth LOWESS) is a data analysis technique to estimate smooth values for noisy data. The user can estimate fitted values without needing to specify a function form for the model, thus allowing this method to be easily applied to non-linear data. The systematic relationship between a set of variables is captured by the fitted LOWESS curve. The addition of the fitted curve can, when implemented correctly, enrich the visual information given by a scatter plot of the dataset.

LOWESS falls into the category of both local and non-parametric regressions. The idea of the local regression method is to estimate a target point using nearby observations, which are weighted based on their proximity. The user therefore cannot make a global assumption regarding the functional form of the relationship however locally it is possible (Cleveland 1979). A non-parametric regression does not require the functional form of the model to be pre-specified, as explained above.

3.1 Method

Given the data generating process $y_i = g(x_i) + \varepsilon_i$, where $g(\circ)$ is an unknown function, our aim is to compute a smooth estimate of it. The data can be represented in a scatter plot (x_i , y_i) where *i=1…n.* The user selects a smoothing parameter $f \in (0,1)$. The smoothing parameter dictates the proportion of the data that will be within each bandwidth used to estimate the smoothed value. Within each bandwidth, a linear polynomial is fit to the data using a weighted least squares local regression. The weighting function $w_k(x_i)$ is centred at the target observation for each bandwidth (Cleveland 1979). Applying the nearest neighbour algorithm, the observations are weighted based on their proximity to the target observation. For example, the closer that observation x_i is to the target point, the higher weight it receives in estimating $\hat{y}_i$ (Liu 2015). This because the closer data points are within the explanatory variable space, the more highly they are correlated.

The estimation of the smoothed value can often be distorted due to the presence of outliers within the data (Liu 2015).To ensure the smoothing procedure is robust to outliers, an extension can be selected that weights the residuals of the fitted values. The robustness weight given to outliers is small, thus reducing the impact on the estimate values (Cleveland 1979). This procedure can be repeated for as many iterations as required until a desired fit is achieved. The final result is given by the point $(x_i, \hat{y}_i)$, where $\hat{y}_i$ is called the fitted value at the smoothed point at x_i (Cleveland 1979).

3.1.1 The Algorithm

The algorithm comes prepacked with the statistical package R. This makes a practical application as easy as selecting the required data set and one line of code. The art in the application is in the choice of the smoothing parameter. The larger the parameter the bigger the neighbourhood of influential points and thus the smoother the fitting of the LOWESS curve. The key is to strike the balance between minimising variability and not distorting the pattern in the data (Cleveland 1979).

Cleveland (1979) defines the following: $\mathrm{h_i}$ = the distance between $\mathrm{x_k}$ and its $\mathrm{q^{th}}$ nearest neighbor, x_k= the value of the linear polynomial fit, $\mathrm{x_k}$= the $\mathrm{i^{th}}$ observation and $w_k(x_i) = W\left(\frac{(x_k - x_i)}{h_i}\right)$ where W is a tricube weight function.

1. Compute the coefficient estimates $\hat{\beta}(x_i)$ for each i. The parameters are estimated using a polynomial regression of degree 1 of y_k on x_k which is fit using by weighted least squares with weights $w_k(x_i)$; $\hat{y}_i$ is the fitted value of the regression at x_i.

2. Estimate outlier robust fitted values by reweighting using Bi-Square Function.

The Residuals $\varepsilon_k = y_i - \hat{y}_i$

Median Residual $s = \widetilde{\varepsilon_k}$

Apply Bi-Square Function $B(x) = \begin{cases}(1-x^2)^2 & for\ |x|<1 \\ 0 & ,for\ |x|\geq 1\end{cases}$

Robustness Weights $\delta_k = B(\varepsilon_k/6\mathrm{s})$

3. Estimate the robust fitted values $\hat{y}_i$ by refitting the polynomial of degree 1 using a weighted least squares regression with the combined weights $\delta_i w_k(x_i)$ for each observation i.

4. Repeat steps 2 & 3 for t iterations.

5. The fitted values can then be plotted at equally spaced points and connected to fit a LOWESS curve to the data.

3.2 Application in R

The ease of application makes LOWESS a very useful tool especially as it can be applied to complex processes without the need for a theoretical model to explain the relationship.

3.2.1 Function Call Sign

The code for the LOWESS method comes pre-packaged with the software package R.

The R documentation provides the function call sign and descriptions of the key arguments.

```
lowess(x, y, f=2/3, iter=3, delta=0.01*diff(range(x)))
```

The Arguments

`x,y` vectors giving the coordinates of the points in the scatter plot

`f` the size of the smoother bandwidth. This gives the proportion of point in the plot which used to estimate the smooth value.

`iter` the number of iterations undertaken to make the estimation of the smoothed value outlier robust.

`delta` data points within delta distance from the previous estimated value are not computed in order to speed up computation.

3.2.2 Example

I plotted the data and the estimated LOWESS function using various smoothing parameters.

```
fm <- lowess(cps71$age, cps71$logwage, f=0.5)
```

Figure 3 combines the estimated LOWESS functions estimated using different smoothing parameters.

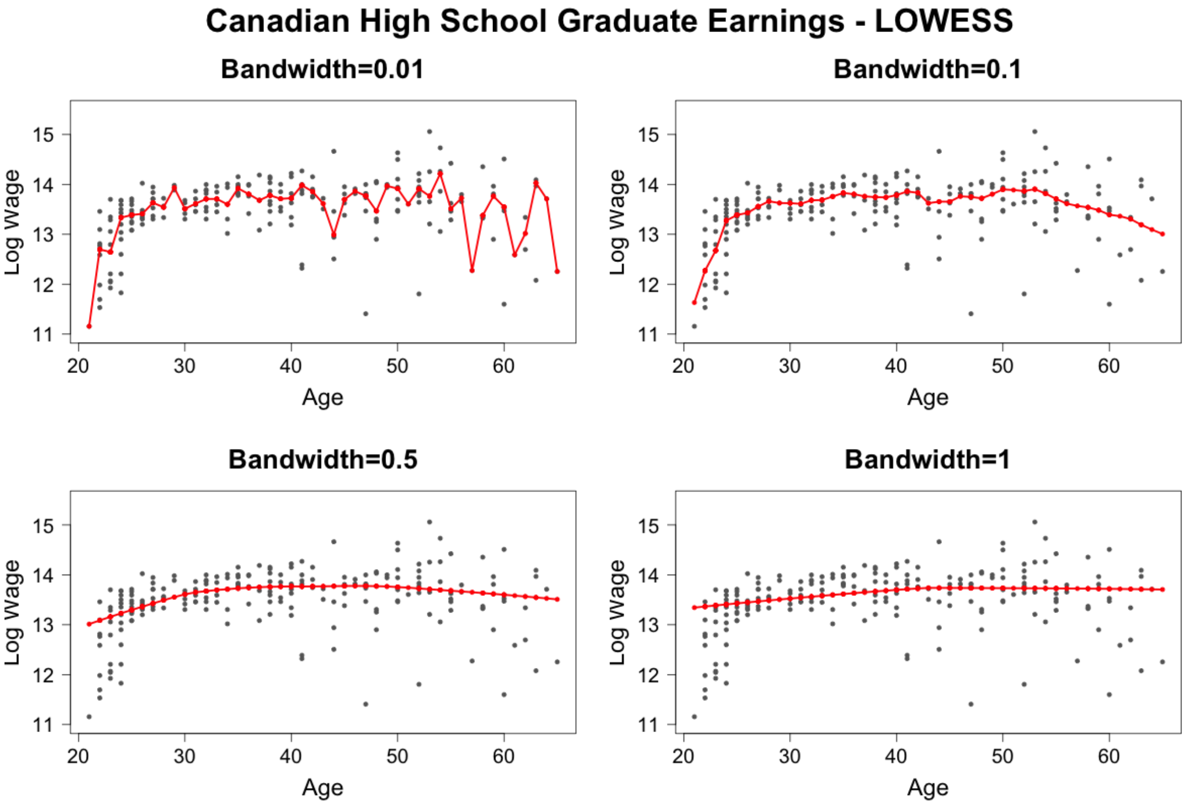

Figure 3: LOWESS Functions estimated in R using the command `lowess`.

3.3 Summary

Locally weighted scatter plot smoothing is a non-parametric method based on the idea of local regressions. The algorithm estimates smoothed values without requiring assumptions regarding the functional form of the covariates or the distribution of the data generating process. The LOWESS curve of the smoothed values, when added to the scatter plot of the original data, proves to be a useful visual tool. LOWESS is easily applied in the software package R using the command `lowess`.

4. Kernel Regression

The Kernel Regression method is a purely data driven non-parametric technique to estimate the conditional expectation of a random variable. The user is able to generate a highly adaptive and non-linear function to the data generating process. The method requires neither a theoretical presumption of the relationship between the covariates, nor the need for assumptions regarding the underlying distribution of the variables. The Kernel Regression method allows the user great freedom to fit the model to the data.

4.1 Kernel Density Estimator

Kernel Regression are built upon a non-parametric method to estimate the probability density function of a random variable, the ‘Kernel Density Estimator’. The method allows the user to estimate a completely unknown probability density function. This makes it extremely flexible in regards to the situations it can be applied, as it requires no prior assumptions in regards to the data generating process. This flexibility transfers to the Kernel Regression method.

To estimate the density of the function $f(x)$, a bandwidth is selected which is the maximum distance an observation can deviate from the target data point and still be used in the estimation. The density is then estimated using a kernel function to weight observations within the selected bandwidth. An example is the uniform kernel function

$$K(u) = \frac{1}{2} I(|\frac{x-X_i}{h}| \leq 1).$$

Where I is an indicator function that returns that returns 1 if the distance between the observation X_i and the target x is less than the specified bandwidth *h*. The uniform kernel function assigns a weight of 0.5 to each observation whose distance from the target point is less then bandwidth. There are many different kernel functions which can be used to estimate density. Figure 3 illustrates some of the common kernel functions and their weights for observations. The choice dictates the weighting given to observations within the bin. The probability density function $\hat{f}(x)$ estimated using the Kernel Density estimator

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - X_i)$$

$$K(\bullet) = \frac{1}{h} K\left(\frac{\bullet}{h}\right)$$

where $K(\bullet)$ is the selected kernel function.

The size of the bandwidth h determines the smoothness of the density estimator. Its choice presents a trade-off for the estimator; between bias indicating the bandwidth is too wide and high variance indicating the bandwidth is too narrow. Bandwidth selection is very important both for the density estimator and Kernel Regression.

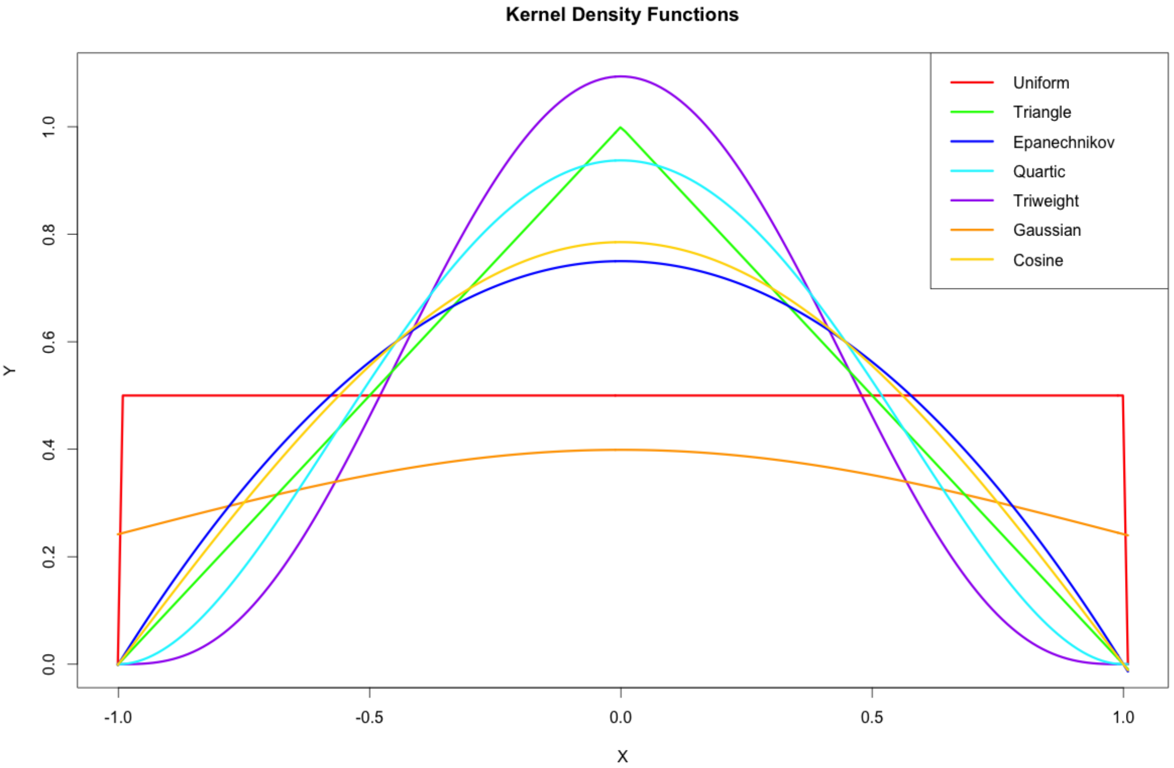

Figure 3: kernel functions illustrating how observations are weighted differently within the specified bin.

4.2 Method

The conditional expectation of the random variable Y given a random variable X is equal to $E(Y|X) = m(X)$ where $m(X)$ is an unknown function. There are no restrictions on the form of $m(X)$ (for example linear) due to a prior theoretical belief of the relationship between the variables (Härdle et al. 2004). The goal is to estimate the unknown function using a Kernel Regression

$$m(x) = E(Y|\mathrm{X} = \mathrm{x}) = \int y \frac{f(x,y)}{f_X(x)} dy = \frac{\int y f(x,y) dy}{f_X(x)}$$

where $f_X(x)$ is the marginal probability density function of X and $f(x,y)$ is the conditional probability density function of Y given X.

The Kernel Density Estimator (see section 4.1) can be used to estimate both density functions

$$\hat{f}(x) = \frac{1}{n}\frac{1}{h}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \text{ and}$$

$$\hat{f}(x,y) = \frac{1}{n}\frac{1}{h^2}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) K(\frac{y-y_i}{h}).$$

Rearranging the estimated density functions gives us the Nadaraya-Watson estimator for $m(X)$, which is the weighted sum of all the values of Y divided by all the kernels

$\widehat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x-x_i)y_i}{\sum_{i=1}^{n} K_h(x-x_i)}$.

The estimator can be rearranged to illustrate that it is analogue to the local (i.e. within the bandwidth) weighted local average of the dependent variable Y

$\widehat{m}_h(x) = \frac{1}{n}\sum_{i=1}^{n} W_{hi}(x)Y_i$.

The estimator is consistent so it converges in probability towards to the population function $\widehat{m}_h(x) \overset{p}{\rightarrow} m(x)$.

The importance of bandwidth selection was highlighted previously for Kernel Density Estimator (see section 4.1). The bandwidth is commonly referred to as the smoothing parameter because it determines the degree of smoothness of the estimator $\widehat{m}_h(x)$. To illustrate, Härdle et al. (2004) select the extreme example: if $h = 0$ then the weights are zero so the function is undefined unless $x = X_i$ - this results in interpolation of the data. If $h = 1$ then the weights also equal one for all values of the predicator variable *X.* The estimator is then a constant function, which assigns the sample mean of $Y$to X(Härdle et al. 2004).

Selecting the size of the bandwidth represents a trade-off between variance and bias. The larger the bandwidth, the larger the bias, while a smaller bandwidth increases the variance. In order to select the optimal bandwidth, Härdle et al. (2004) argue it needs to fit two criteria. Firstly it should have desired theoretical properties, in other words the estimate should be close as possible to the population function. Secondly the method should be easy to apply in practice. The second criterion restricts the use of most theoretically desirable bandwidth optimization methods.

There are a number of theoretical measures for optimising the trade off. The mean square error $MSE(x,h) = E[\{\widehat{m}_h(x) - m(x)\}^2]$ is the difference between the squared deviations of the unknown function and the estimator at single point. A further downside is that this is only a local measure. The Integrated Square Error $ISE\{\widehat{m}_h\} = \int_{-\infty}^{\infty}\{\widehat{m}_h(x) - m(x)\}^2 w(x) f_X(x)dx$ is a global discrepancy measure, however depending on the selected sample the estimates for $\widehat{m}_h(x)$ will be different and with them, the error. The inclusion of $w(x)$ a weighting function, reduces the variance in regions of sparse data by limiting their influence (Härdle et al. 2004). The Average Square Error $ISE\{\widehat{m}_h\} = \frac{1}{n}\sum_{j=1}^{n}\{\widehat{m}_h(X_j) - m(X_j)\}^2 w(X_j)$ is a discrete approximation of the ISE. The problem with all three measures is that they include the unknown function that is being estimated, therefore making its application in practice difficult.

There is a solution that allows the user to optimise the size of the bandwidth and that fits both criteria. The idea is to replace the unknown function $m(x)$ with the observations of *Y*, however the problem is that Y is used to predict $\widehat{m}_h(x)$. Cross validation presents a simply solution by utilising a method called leave one out estimation (Härdle et al. 2004)

$$\widehat{m}_{h,-1}(X_i) = \frac{\sum_{i\neq j}^{n} K_h(X_i-X_j)Y_i}{\sum_{i\neq j}^{n} K_h(X_i-X_j)}.$$

The i^{th} observation is not used in the estimation of $\widehat{m}_{h,-i}$ which replaces $\widehat{m}_h$. The cross validation function is then given by

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \widehat{m}_{h,-i}(X_i)\right\}^2 w(X_i).$$

The bandwidth h should be selected to minimise the cross validation function; this is equivalent to minimising the average square error.

4.2.1 Multivariate Datasets

Kernel Regressions can be applied to multivariate datasets; however one particular problem needs to be accounted for. This is the curse of dimensionality, due to the fact that each added dimension increases the number of bins by a square factor. As the number of bins increases, so does the likelihood that they will be sparsely populated.

4.3 Application in R

There are two options to implement Kernel Regression in R. The Kernel Regression Smoother `ksmooth` comes pre-packaged with the software and will compute the Nadaraya-Watson estimator. To optimize the bandwidth via the cross validation method requires the installation of the package Summary Information `sm`. The alternative requires the installation of the package Non-parametric Kernel Smoothing Methods for Mixed Data Types `np`. The package `np` allows the user to combine various data types. It automatically selects the bandwidth. I will illustrate the application of both methods.

4.3.1 Function Call Sign

```
library(sm)
```

```
cv <- hcv(x, y, hstart=NA, hend=NA)
```

`x` vector of covariate values for the non-parametric regression.

`y` vector response variable for the non-parametric regression.

`hstart` the small value of the grid points to be used for the initial search.

`hend` the largest value for the grid points to be used for the initial search.

The Kernel Regression smoother

```
ksmooth(x, y, kernel = "normal", bandwidth=cv)
```

`x` input values x.

`y` input values y.

`kernel` the kernel function to be used.

`Bandwidth` the bandwidth set to the optimal level given by cross validation.

```
library(np)
```

```
npreg(x, y)
```

`x` input values x.

`y` input values y.

4.3.2 Example

The Kernel Regression is then estimated using the code.

```
ksmooth(age, logwage, kernel="normal", bandwidth=0.5)
```

I varied the size of the bandwidth selected to illustrate impact on the estimated function. It is easy to see the impact on the estimate function of reducing the bandwidth. To select the optimal bandwidth, select the value that minimizes the cross validation function. The optimal bandwidth is approximately equal `3.27`.

```
cv <- hcv(age, logwage, display="lines")
```

```
cv

 [1] 3.271217

ksmooth(age, logwage, kernel="normal", bandwidth=cv)
```

Figure 4 plots the Nadaraya-Watson estimator for various bandwidths including the cross validation optimal measure.

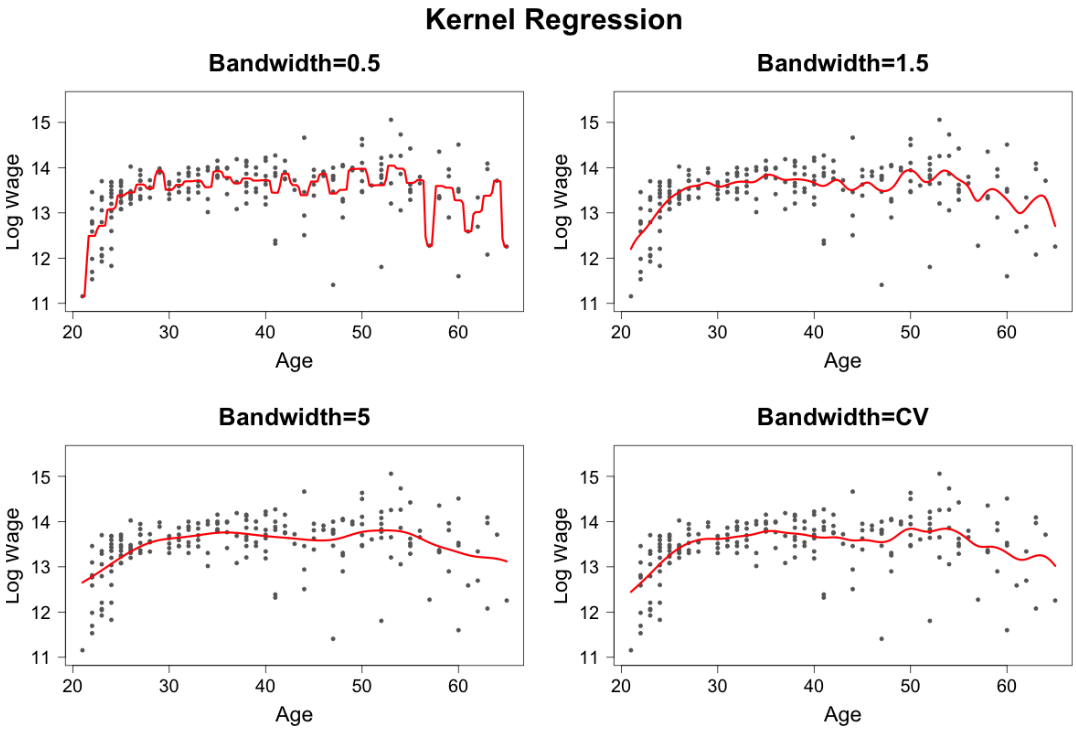

Figure 4: Kernel Regression estimated in R using the Kernel Smoother command ksmooth.

The code for the estimation of the Kernel Regression function using the Non-parametric package np.

```
k <- npreg(age, logwage)

plot(k,plot.errors.methods="asymptotic",plot.error.style=
"band", ylim=c(11,15.5))

points(age, logwage, cex=0.25)
```

Figure 5 plots the results of the non-parametric regression. The results are very similar to those attending using the cross validation optimal bandwidth.

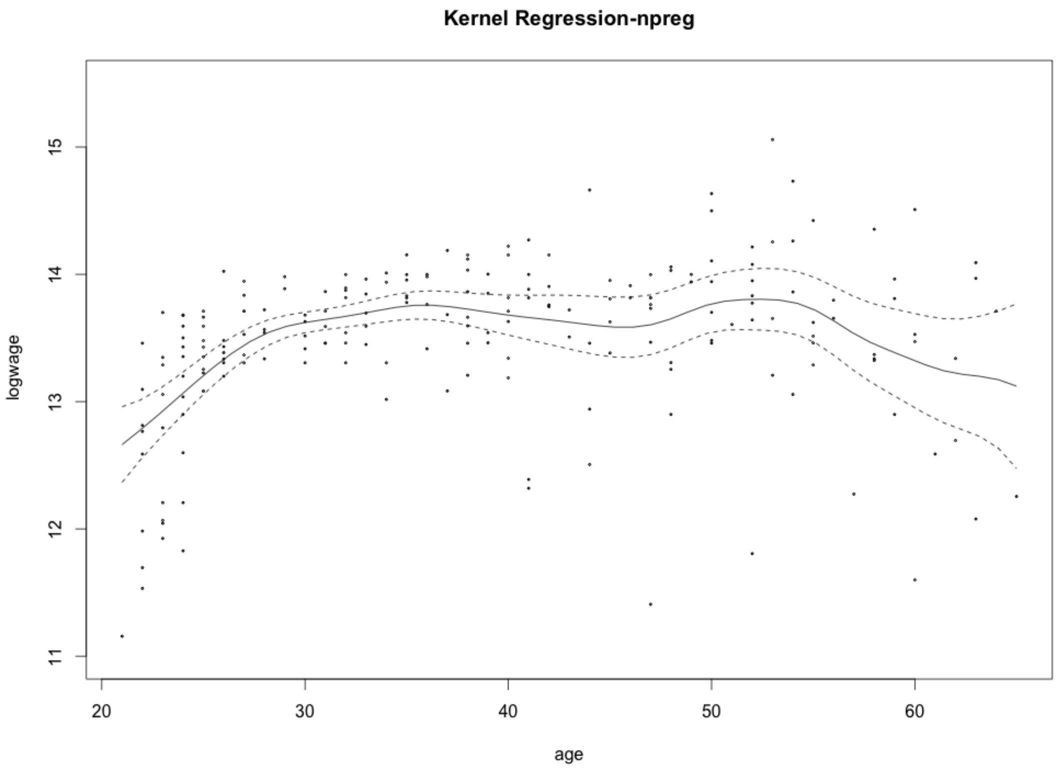

Figure 5: Kernel Regression estimated in R using the non-parametric package `npreg` with the asymptotic variability bounds.

4.4 Summary

Kernel Regressions allow the user to estimate the $E(Y|X) = m(X)$ where $m(X)$ is an unknown function. The method is based on the idea of estimating the probability density functions using the Kernel Density Estimator. The method requires no theoretical presumption of the relationship between the covariates, nor the need for assumptions regarding the underlying distribution of the variables. To obtain a good estimation, bandwidth selection is important. The selection presents a trade-off between increasing bias and increasing variance. The optimal bandwidth is obtained by minimising the cross validation function. Kernel Regressions can be easily applied in the software package R using the commands `ksmooth` or `npreg`.

5. Comparison

Table 1: Method Comparison

Generalized Additive Model	LOWESS	Kernel Regression
1. Extends the linear model by allowing parametric, semi-parametric and non-parametric covariates therefore the user can account for non-linear relationships between variables.	1. A non-parametric data driven method of fitting a smoothed function to the data generating process.	1. A non-parametric data driven method of estimating the conditional expectation of a random variable.
2. Requires pre-specified functional form. However the linear model, individual covariates can therefore still be interpreted in a ceteris paribus manner and conventional inference can be undertaken.	2. No assumption required regarding the relationship between the covariates, their distributions or the functional form of the model.	2. No assumption required regarding the relationship between the covariates, their distributions or the functional form of the model.
3. Applies a smoothing method to estimate the functions that makes up the individual components of the model.	3. Applies a local weighted regression to determine the weight of each observation within the bandwidth in estimating the fitted value.	3. Applies a kernel function to data within a pre-determined bandwidth to estimate the unknown marginal and conditional probability density functions required to compute the conditional expectation.

6. Conclusion

This paper gives an introduction to three methods for evaluating non-linear and non-parametric data generating processes. The linear regression model is an excellent tool for data analysis, however it requires restrictive assumptions about the data generating process to be successfully estimated. The concepts I introduced in this paper allow the user greater flexibility to successfully estimate models based purely on information contained within the data. The Generalized Additive Model is an extension of the generalized linear model that incorporates smoothing to estimate the impact of non-linear covariates. Locally Weighted Scatter Plot Smoothing fits a smooth function by applying a local weighted regression to data points within a user-selected bandwidth. Kernel Regression estimates the conditional expectation via the estimation of the marginal and conditional probability density functions within a specified bandwidth using a kernel function. The paper introduces and then carefully explains the idea and mathematical formulas. I illustrate how to implement each method in the software package R by listing the call signs and explaining the arguments. Finally I apply each method to the non-linear dataset `cps71`.

7. References

Clarke, Michael. Generalized Addititve Models: Getting Started with Additive Models in R. Centre for Social Research University of Notra Dame, 2012.

Cleveland, William S. "Robust Locally Weighted Regression and Smoothing Scatter Plots." Journal of American Statistical Association 74.368 (1979): 829-836.

Härdle, Wolfgang., Müller, Marlene., Sperlich, Stefan., and Werwatz, Axel. Nonparmetric and Semiparametric Models. Berlin: Springer, 2004.

Hastie, Trevor., Robert, Tibshirni., and Freidman, Jerome. The Elements of Statistical Learning: Data Mining, Inference and Predication. Springer, 2008.

Heckert, Alan., and Filliben, James J. "LOWESS Smooth." Dataplot Reference Manual Volume 1. National Institute of Standards and Technology. 12 07 2015 <http://www.itl.nist.gov/div898/software/dataplot/refman1/ch3/lowess_s.pdf>.

James, Gareth., Witten, Daniela., Hastie, Trevor., and Tibshirni, Robert. An Introduction to Statistical Learning with Applications in R. New York: Springer, 2013.

Liu, Peng. "Statistics 416 ." Iowa State University. <http://streaming.stat.iastate.edu/~stat416/LectureNotes/handout_LOWESS.pdf>.

Wikipedia. Backfitting algorithm. 02 07 2015 <https://en.wikipedia.org/wiki/Backfitting_algorithm>.

Wikipedia. General Additive Model. 01 07 2015 <https://en.wikipedia.org/wiki/Generalized_additive_model>.

Wikipedia. Kernel Regressions. 03 07 2015 <https://en.wikipedia.org/wiki/Kernel_regression>.

Wikipedia. Local Regressions. 01 07 2015 <https://en.wikipedia.org/wiki/Local_regression>.

Wikipedia. Nonparametric Regressions. 2015. 12 07 2015 <https://en.wikipedia.org/wiki/Nonparametric_regression>.