**Instituto Nacional de Matemática Pura e Aplicada**

# Tree-Based Model for Estimating the Local Volatility Surface

**Yuri Resende Fonseca**

Advisor: **Yuri Fahham Saporito**
Co advisor: **Rodrigo dos Santos Targino**

**Rio de Janeiro**
**Agosto de 2018**

To my parents.

**Acknowledgements**

First, I would like to thank all my professors that contributed with dedication to my academic and personal development. To the Institute for Pure and Applied Mathematics for the opportunities provided and most of all, to my parents, girlfriend, and sister, for their love, support and patience.

**Abstract**

In this work we proposed a new method to estimate local volatility functions with a non parametric approach. The method is based on the statistical learning literature and uses gradient boosting with smooth transition trees as base learners. The smoothness and robustness of the method generates well behaved local volatility functions, capable of replicating *vanilla* option prices and the implied volatility surface. Furthermore, the method proved to be useful for pricing exotic options. We tested the method for simulated data, Asian calls, Float strike calls and Barrier knockout options.

**Keywords:** Local volatility function, implied volatility, gradient boosting, tree methods.

**Resumo**

Nesse trabalho proponho um novo método para estimação da superfície de volatilidade local com uma abordagem não paramétrica. O método é baseado na literatura de *statistical learning* e usa *gradient boosting* com *smooth trees* como *base learner*. A suavidade e robustes do método gera superfícies de volatilidade local bem comportadas, capazes de replicar o preço de opções *vanilla* e a superfície de volatilidade implícita. Além disso, o método se provou útil para precificar opções exóticas. O método foi testado com dados simulados, opções de venda Asiáticas, opções de venda com Strike flutuante e opções com Barreira e *knockout*.

**Palavras-chaves:** Superfície de volatilidade local, volatilidade implícita, gradient boosting, métodos baseados em árvores.

# Contents

# List of Figures

# List of Tables

# CHAPTER 1

# Introduction

It is well accepted that under the hypothesis of efficient markets, asset prices cannot be predicted [28, 30]. Therefore, financial derivative instruments have been widely used in financial markets in order to mitigate some risks of assets and portfolios. The most common types of derivatives are forwards, futures, options and swaps. Pricing these derivatives is a problem almost as old as the financial markets themselves. However, the seminal work of Louis Bachelier [5] is considered the origin of modern quantitative finance.

Despite the fact that the work was not recognized as it should at the time, Louis Bachelier's "Theory of Speculation" was of significant importance years later for the work developed by Fischer Black and Myron Scholes [7], leading to the so-called Black Scholes formula.

In [7], the authors proposed a Geometric Brownian Motion for the underlying (usually an asset price) and demonstrated that the price of an European Call option, as a function of the underlying and time, should satisfy a parabolic Partial Differential Equation. After a change of variables, the explicit analytical formula for the European call option was derived. Using a non-arbitrage argument, it is possible to find the analytical formula for the European put option as well. With the analytical expressions for the *vanilla options* (European calls and puts), one could now build the well known delta-hedge portfolios, replicating asset movements with derivatives.

Despite the importance of the Black Scholes formula and the attempts to build delta-hedge portfolios, which were supposedly risk-free, the market soon begun to realize that *vanilla* options with different strikes and maturities had different values for their *implied volatilities*. The implied volatility is the unique values that one should plug into the the

Black and Scholes formula in order to reproduce market prices. This surfaces of Implied Volatility, called *smiles*, suggested that the original hypothesis of an asset with prices following a Geometric Brownian Motion with constant diffusion parameter is indeed too strong. Therefore, more general models started to be proposed in the late 1980's.

Local Volatility models, are specially interesting in Financial Mathematics due to the fact that the diffusion coefficient is a deterministic function of the current asset price and time. This function is commonly called the volatility surface. Moreover, since we are not adding additional risk factors, we keep the completeness of the market. Hence, one can use Local Volatility models to compute options price and build hedge portfolios. The main problem related to the estimation of the local volatility surface is that it is an ill-posed problem in the Hadamard sense (see the definition in [6]). Therefore, techniques such as Tikhonov regularization are commonly used [22, 17, 2].

In this work, a different approach will be used to deal with the estimation of the local volatility surface that also accounts for regularization. However, instead of conventional finite difference methods [30]. An algorithm based on regression trees which combines Gradient Boosting [19] with smooth transition trees [11] will be used. The new methodology is completely data-driven. Moreover, it is more stable then conventional numerical models for solving PDEs and the final output is an analytic smooth function that allows the calculation of the volatility surface in any strike value, maturity and their respective derivatives.

The work will be divided as follows, in Sections 2 and 3, a literature review for Local Volatility and tree based algorithms will be presented. In Section 4, the proposed algorithm is developed. In Section 5, the numerical results are showed and finally, in Section 6, we have the concluding remarks.

# CHAPTER 2

# Local Volatility Model

## 2.1 The Black and Scholes Formula

Following the hedging argument shown in [1] and assuming that prices moves according to a Geometric Brownian Motion, it is possible to demonstrate that the price of a Call option with Strike $K$ and maturity $T$ for a non-dividend payment stock can be expressed in terms of the Partial Differential Equation (PDE):

$$\begin{aligned} &\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0, \quad 0 \leq t \leq T, \\ &C(S_T, T) = (S_T - K)^+. \end{aligned} \tag{2.1}$$

Changing the variables, it is possible to transform the previous PDE into a standard heat equation [7], which can be solved analitically. This solution is given by the Black and Scholes formula:

$$\begin{aligned} &C^{BS}(t, S_t, T, K, \sigma) = S_t\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \\ &d_1 = \frac{\ln(S_t/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = d_1 - \sigma\sqrt{T-t}, \end{aligned} \tag{2.2}$$

where $t \in [0, T]$, $r$ is the risk free rate and $\sigma$ is a positive constant called volatility and $\Phi$ is the cumulative function of the standard normal distribution. Using a no-argitrage argument [30], one can construct the so-called put-call parity (2.3) and build an analytical

solution for the price of a put option.

$$C^{BS}(t, S_t, T, K, \sigma, r) + e^{-r(T-t)}K = S_t + P^{BS}(t, S_t, T, K, \sigma). \tag{2.3}$$

The sensitivity of the option price with respect to the different parameters is the so called Greeks. The most common ones are defined as:

$$\begin{aligned}
\Delta &= \frac{\partial C_{BS}}{\partial S} = N(d_1) > 0, \\
\Gamma &= \frac{\partial^2 C_{BS}}{\partial S^2} = \frac{n(d_1)}{S_t\sigma\sqrt{T-t}} > 0, \\
\Lambda &= \frac{\partial C_{BS}}{\partial \sigma} = n(d_1)S_t\sqrt{T-t} > 0, \\
\Theta &= \frac{\partial C_{BS}}{\partial t} = -\frac{n(d_1)S_t\sigma}{2\sqrt{T-t}} - rKe^{-r(t-t)}N(d_2) < 0, \\
\rho &= \frac{\partial C_{BS}}{\partial r} = K(T-t)e^{-r(T-t}N(d_2) < 0, \\
V &= \frac{\partial^2 C_{BS}}{\partial \sigma^2} = n(d_1)S_t\sqrt{T-t}\frac{d_1 d_2}{\sigma}, \\
\Psi &= \frac{\partial^2 C_{BS}}{\partial S_t \sigma} = -n(d_1)\frac{d_2}{\sigma}.
\end{aligned} \tag{2.4}$$

The Greeks are specially important when building a portfolio. For instance, if one would like to hedge a portfolio against moves of the asset price, $\Delta$ will give the number of stocks that one should sell in order to hedge a Call. Thus, the portfolio will be given by $\Pi_t = C_t(K,T) - \Delta S_t$.

Another interesting Greek is $\Lambda_{BS}$. Since $\Lambda_{BS} > 0$, we have that (2.2) is an increasing function with respect to $\sigma$. In other words, the bigger the risk the bigger the premium of an option. In another direction, the shorter the maturity, the shorter the premium of an option. This intuitive concept is a consequence of $\Theta_{BS} < 0$.

## 2.2 Implied Volatility

Given the price of a call or a put option, the *implied volatility* $\sigma_{IV}$ is the unique value used in the Black Scholes formula to recover the option price. Since $\Lambda_{BS}$ is always positive, we can guarantee that $\sigma_{IV}$ is unique for the price in the range of no-arbitrage. Moreover, when $\sigma \to 0$, ($\sigma \to \infty$), the Black and Scholes formula approaches the lower and upper no-arbitrage bounds for the call, respectively. See [26]. Therefore, one can calculate the implied volatility using a root-finding technique or solving the optimization

problem:

$$\sigma_{IV} = \underset{\sigma \in \mathbb{R}^+}{\arg\min} \left( C(K,T) - C^{BS}(S_t, r, K, T, t, \sigma) \right)^2, \tag{2.5}$$

where $C(K,T)$ is the option price observed in the market and $C^{BS}$ is the formula given in (2.2).

As suggested by [26], rather than just the hypothesis of constant volatility, the implied volatility surface can be seen as a language to understanding option prices in an alternative way, which is more useful than just raw options prices. The implied volatility can be used as a metric to compare options across strikes, maturities, underlyings and observations times.

For the first case, let the volatility be a function of time. Then, one can write:

$$dS_t = rS_t dt + \sigma(t) S_t dW_t. \tag{2.6}$$

Defining $\bar{\sigma} = (\frac{1}{T}\int_0^T \sigma^2(t)dt)^{1/2}$, and applying Itô's formula in (2.6), we can see that:

$$\begin{aligned} d(\ln S_t) &= (r - \frac{\sigma^2(t)}{2})dt + \sigma(t)dW_t, \\ \ln S_T &= \ln S_0 + \int_0^T r dt - \int_0^T \frac{\sigma^2(t)}{2} dt + \int_0^T \sigma(t) dW_t, \\ &= \ln S_0 + (r + \frac{\bar{\sigma}^2}{2})T + \int_0^T \sigma(t) dW_t. \end{aligned}$$

Therefore, $\ln S_T$ has mean $(r + \frac{\bar{\sigma}}{2}2)T$ and variance $\bar{\sigma}^2 T$, as a consequence:

$$C(K,T) = C_{BS}(K, T, \bar{\sigma}).$$

Thus, the implied volatility is the time average of instantaneous volatility for this specific case, where the asset price follows (2.6).

Many works tried to model directly the implied volatility surface, see for example [15], [23] and [25]. However, the conventional approach is to use Stochastic Volatility models and Local Volatility functions, since both of them are capable of generating smiles [21], [16].

## 2.3 Local Volatility

How consistent are market's option prices with the Black Scholes formula? This question has raised an important issue regarding one of the Black and Scholes hypotheses, whether volatility is constant or not in the stochastic process that models asset prices.

Since the 90's, it is a stylized fact that inverting the BS formula using market data and plotting the implied volatility as a function of strike of maturity, one would see not a constant value for the implied volatility as expected in the Black and Scholes world, but rather a surface commonly called smile.

To deal with this stylized fact, [13] and [16] suggested that instead of using constant volatility, the price of a option's underlying asset should be a function of both the current level of the asset and time. Hence, instead of (2.1) one would have (2.8).

$$\begin{aligned} &\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 C}{\partial S^2} - rC = 0, \\ &C(S_T, T) = (S_T - K)^+. \end{aligned} \tag{2.7}$$

At this point, it is interesting to introduce the Feynman-Kac's formula, which states that if $u \in C^{2,1}(\mathbb{R} \times [0,T])$ satisfies the PDE:

$$\begin{aligned} &\frac{\partial u}{\partial t}(x,t) + \mu(x,t)\frac{\partial u}{\partial x}(x,t) + \frac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2}(x,t) = r(x,t)u, \\ &u(x,T) = g(x), \end{aligned} \tag{2.8}$$

then. it can be written as:

$$u(x,t) = \mathbb{E}[e^{-\int_t^T r(X_u,u)du} g(X_T)|X_t = x], \tag{2.9}$$

for a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the process $X$ being given by the equation:

$$dX_t = \mu(X_t,t)dt + \sigma(X_t,t)dW_t, \tag{2.10}$$

where $W$ is a standard brownian motion under $\mathbb{P}$.

Therefore, applying (2.9) in (2.20), we can see that the SDE for the price process $S$ is given by:

$$dS_t = rS_t dt + \sigma(S_t, t)S_t dW_t, \quad t \geq 0, \tag{2.11}$$

where the function $\sigma$ is the so-called Local Volatility surface.

The strike $K$ and maturity $T$ are fixed in (2.8). Therefore, this PDE is a backward equation for the option prices. However, it is also possible to write the forward equation, fixing $S_t$ and $t$. Following [20] we can write the forward price of a call option as:

$$\widetilde{C}(K,T) = e^{r(T-t)}C(K,T). \tag{2.12}$$

Using (2.9), we have:

$$\widetilde{C}(K,T) = \mathbb{E}^{\mathbb{Q}}[(S_T - K)] = \int_K^\infty (x-K)\varphi(x,T)dx, \tag{2.13}$$

where $\varphi(.,T)$ is the probability density of $S_T$. Taking the derivative in (2.13) twice with respect to $K$:

$$\frac{\partial \widetilde{C}}{\partial K}(K,T) = \int_K^\infty -\varphi(x,T)dx, \tag{2.14}$$

$$\frac{\partial^2 \widetilde{C}}{\partial K^2}(K,T) = \varphi(K,T). \tag{2.15}$$

Moreover, $\varphi$ satisfies the Fokker-Planck equation (6):

$$\frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma^2 x^2 \varphi(x,T)) - \frac{\partial}{\partial x}(rx\varphi(x,T)) = \frac{\partial \varphi}{\partial T}(x,T). \tag{2.16}$$

Hence, the derivative of (2.13) with respect to $T$ will be given by:

$$\begin{aligned}\frac{\partial \tilde{C}}{\partial T}(K,T) &= \int_K^\infty (x-K)\frac{\partial \varphi}{\partial T}(x,T)dx \\ &= \int_K^\infty (x-K)\left\{\frac{1}{2}\frac{\partial}{\partial x^2}(\sigma^2 x^2 \varphi(x,T)) - \frac{\partial}{\partial x}(rx\varphi(x,T))\right\}dx.\end{aligned} \tag{2.17}$$

Integrating equation (2.17) by parts we have:

$$\frac{\partial \tilde{C}}{\partial T}(K,T) = \frac{\sigma^2 K^2}{2}\varphi(x,T) + \int_K^\infty rx\varphi(x,T)dx. \tag{2.18}$$

Using (2.15) we find Dupire's equation:

$$\begin{aligned} &\frac{\partial \tilde{C}}{\partial T} - \frac{1}{2}\sigma^2(K,T)K^2\frac{\partial^2 \tilde{C}}{\partial K^2} + rK\frac{\partial \tilde{C}}{\partial K} = 0, \\ &\tilde{C}(K,T) = (S_0 - K)^+. \end{aligned} \tag{2.19}$$

Using (2.12) in (2.19) we have:

$$\begin{aligned} &\frac{\partial C}{\partial T} - \frac{1}{2}\sigma^2(K,T)K^2\frac{\partial^2 C}{\partial K^2} + rK\frac{\partial C}{\partial K} + rC = 0, \\ &\tilde{C}(K,T) = (S_0 - K)^+. \end{aligned} \tag{2.20}$$

It is also interesting to note the connection between the Local Volatility function and the Implied Volatility. Using (2.8), one could write:

$$\sigma^2(K,T) = 2\frac{\frac{\partial \tilde{C}}{\partial T} + rK\frac{\partial \tilde{C}}{\partial K}}{K^2\frac{\partial^2 \tilde{C}}{\partial K^2}}. \tag{2.21}$$

Following [20], one can use the *log-moneyness* $y = \ln(\frac{K}{S_t e^{r(T-t)}})$ and define the total implied variance as $w(\tau, y) = \tau\sigma_{IV}^2(\tau, y), \tau = T - t$. By (2.2) we have that:

$$\begin{aligned} C(K,T) &= C^{BS}(\tau, y, K, \sigma_{IV}(\tau, y), r) \\ &= S_0\bigg(\Phi\big(-\frac{y}{\sqrt{w}} + \frac{1}{2}\sqrt{w}\big) - e^y\Phi\big(-\frac{y}{\sqrt{w}} - \frac{1}{2}\sqrt{w}\big)\bigg). \end{aligned} \tag{2.22}$$

Moreover, using the *log-moneyness* in (2.20), Dupire's equation becomes:

$$\frac{\partial C}{\partial T} = \frac{\sigma^2(S,t)}{2}\left(\frac{\partial^2 C}{\partial y^2} - \frac{\partial C}{\partial y}\right) + rC. \tag{2.23}$$

Taking the derivatives of the Black Scholes formula with respect to the total implied variance and *log-moneyness* one can check that:

$$\begin{aligned}
\frac{\partial^2 C_{BS}}{\partial w^2} &= \left( -\frac{1}{8} - \frac{1}{2w} + \frac{y^2}{2w^2} \right) \frac{\partial C_{BS}}{\partial w}, \\
\frac{\partial^2 C_{BS}}{\partial y \partial w} &= \left( \frac{1}{2} - \frac{y}{w} \right) \frac{\partial C_{BS}}{\partial w}, \\
\frac{\partial^2 C_{BS}}{\partial y^2} - \frac{\partial C_{BS}}{\partial y} &= 2 \frac{\partial C_{BS}}{\partial w}.
\end{aligned} \tag{2.24}$$

Using (2.22) to transform (2.20) in terms of implied variance we have that:

$$\begin{aligned}
\frac{\partial C}{\partial y} &= \frac{\partial C_{BS}}{\partial y} + \frac{\partial C_{BS}}{\partial w} \frac{\partial w}{\partial y}, \\
\frac{\partial^2 C}{\partial y^2} &= 2 \frac{\partial^2 C_{BS}}{\partial y \partial w} \frac{\partial w}{\partial y} + \frac{\partial^2 C_{BS}}{\partial w^2} \left( \frac{\partial w}{\partial y} \right)^2 + \frac{\partial C_{BS}}{\partial w} \frac{\partial^2 w}{\partial y^2}, \\
\frac{\partial C}{\partial T} &= \frac{\partial C_{BS}}{\partial T} + \frac{\partial C_{BS}}{\partial w} \frac{\partial w}{\partial T} = \frac{\partial C_{BS}}{\partial w} \frac{\partial w}{\partial T} + r(T) C_{BS}.
\end{aligned} \tag{2.25}$$

Now, using (2.24) and (2.25), after some algebra we find the PDE (2.26) below that connects the Local Volatility function and the Implied Volatility in terms of total implied variance.

$$\sigma^2(\tau, K) = \frac{\dfrac{\partial w}{\partial \tau}}{1 - \dfrac{y}{w} \dfrac{\partial w}{\partial y} + \dfrac{1}{4} \left( -\dfrac{1}{4} + \dfrac{1}{w} \dfrac{y^2}{w^2} \right) \left( \dfrac{\partial w}{\partial y} \right)^2 + \dfrac{1}{2} \dfrac{\partial^2 w}{\partial y^2}}. \tag{2.26}$$

Moreover, [20] shows that for very short maturity ($\tau \to 0$), and ATM (at the money) options ($y \sim 0$), we have

$$\frac{\partial \sigma}{\partial y}(0,0) = 2 \frac{\partial \sigma_{IV}}{\partial y}(0,0). \tag{2.27}$$

Thus, the Local Volatility skew (the derivative of the volatility with respect to the log-moneyness) is twice the Implied Volatility skew.

## 2.4 Estimation of Local Volatility

First, let us write (2.20) as:

$$\sigma^2(K,T) = 2\frac{\frac{\partial C}{\partial T} + rK\frac{\partial C}{\partial K} + rC}{K^2\frac{\partial^2 C}{\partial K^2}}. \tag{2.28}$$

Therefore, we need to calculate the partial derivatives in the expression above in order to calculate the values of the local volatility. One strategy is to consider the following Taylor expansion:

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \frac{f'''(x)h^3}{3!} + o(h^4), \\ f(x-h) &= f(x) + f'(x)h + \frac{f''(x)h^2}{2!} - \frac{f'''(x)h^3}{3!} + o(h^4). \end{aligned} \tag{2.29}$$

Hence, $f'$ can be expressed as the forward difference, backward difference or centered difference, where the later is given by:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + o(h), \tag{2.30}$$

and the second derivative will be given by:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + o(h^2). \tag{2.31}$$

Since usually there are not enough different maturities and strikes in the observed dataset, simply using the derivative approximations considered in (2.30) and (2.31) to calculate the local volatility function is not reasonable. Therefore, one could interpolate options prices and then use these approximations in order to calculate the local volatility values. One of the methods commonly used for this interpolation is the Kahalé method [24].

However, it is well-known that theoretically and empirically, estimation of local volatility surfaces is a highly ill-posed problem [2], [12], [17]. Hence, just use interpolated observations for estimating numerically the local volatility might not lead to reasonable results. To solve this problem, techniques that improve the robustness of the estimation of the local volatility function have been proposed. Among successful strategies, different types of regularization techniques have been used, such as entropy-based regularization, in the

spirit of [4] or convex type regularization [2], [12], [17].

In a general approach, one seeks to minimize the Tikhonov functional:

$$\mathcal{F}(x) = ||F(x) - y||^2_{L^2(D)} + \beta f_{x_0}(x), \tag{2.32}$$

over $\mathcal{D}(F)$. Where $\beta$ is the regularization parameter and $f_{x_0}$ is the regularization functional which can be interpreted as an *a priori* information about $F$.

For example, when $f_{x_0}$ is given by:

$$f_{x_0} = KL(x_0, x) = \int_D \left( x \ln\left(\frac{x_0}{x}\right) - (x_0 - x) \right) dx, \tag{2.33}$$

then we have the Kullback-Leibler regularization.

In a different way, instead of using regularization techniques and combine then with conventional methods for solving PDEs, we will use tree based techniques from the statistical learning literature to estimate the local volatility function. Thus, in the next section a brief review about tree based methods will be presented.

# CHAPTER 3

# Tree Based Methods

In the fields of *Statistical and Machine Learning*, tree based methods have been widely used for approximating highly non-linear functions and forecasting in high dimensional problems, [18]. Although tree based models such as Random Forest [9] and Gradient Boosting with Regression or Classification Trees [19] were published in the early 2000's, we will first introduce the Classification and Regression Trees (CART) models [10] developed in the 80's.

## 3.1 CART

A regular regression tree is a non-parametric model that approximates a non-linear function with local predictions using recursive partitioning of the space of the predictor variables. A tree may be represented by a graph as in figure 3.1.

More formally, let $x = (x_1, \ldots, x_m)'$ be a set of $m$ explanatory variables for the response variable $y$. The relationship between $y$ and $x$ is mapped by an unknown function $f$ such that:

$$y \approx f(x) + \varepsilon, \tag{3.1}$$

where $\varepsilon$ is an error term with zero mean, orthogonal to x and no assumptions regarding its distribution. Following the notations in [11], a Regression Tree model with $T$ terminal nodes (leaves) is a recursive partitioning model that approximates $f$ by a general nonlinear function $h(., \psi)$, where $\psi$ is a vector of parameters. In regular Regression Tree, $h(., \psi)$ is a piecewise function with $T$ subregions which cuts that are orthogonal to the axis of the

predictor variables. Each subregion represents one terminal node, and they are denoted by $T_j(\theta_j)$, $j = 1, \ldots, T$. The parameter $\theta_j$ characterizes each subregion such that:

$$\hat{f}(x) = \sum_{j=1}^{T} \beta_j I_j(x; \theta_j), \tag{3.2}$$

where $I_j(x; \theta_j)$ is an indicator function such that:

$$I_j(x; \theta_j) = \begin{cases} 1 & \text{if } x \in T_j(\theta_j), \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

In this case $\psi = (\beta_1, \ldots, \beta_T, \theta'_1, \ldots, \theta'_T)$, taking value 1 if an observation belongs to that partitioning and 0 otherwise. By definition, the same observation may not be in two subregions at the same time.

In figure 3.1, one can check an example for the database *mtcars* avaiable at R. In this case, we are trying to use the explanatory variables *hp* and *wt* to predict the output variable *mpg*. The variables represents the power, weight and miles per gallon performance of different type of car engines. Indeed, one could easily agree that the higher the weight and the higher the power of a motor, more fuel it is going to consumes.

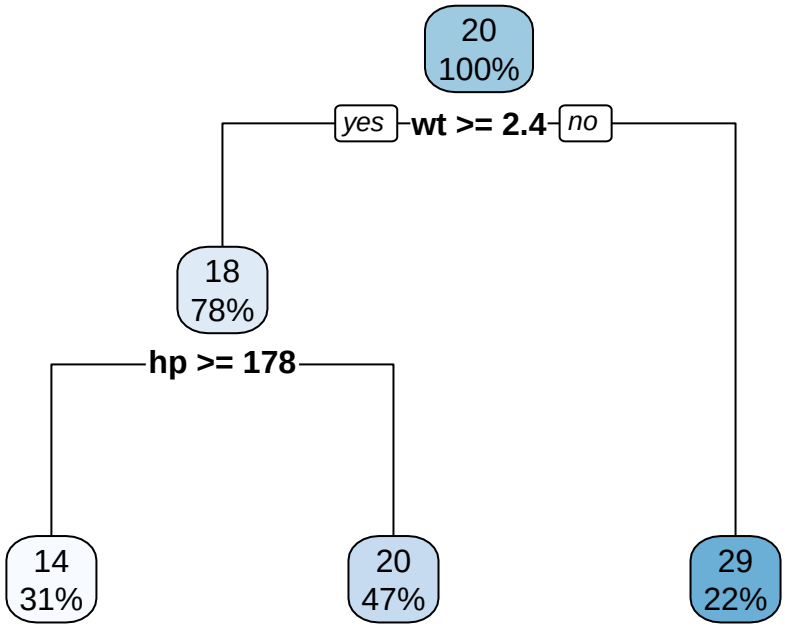


*Figure 3.1:* Example of Regression Tree

In order to adjust a CART model one needs to minimize the empirical risk:

$$\min \sum_{t=1}^{T} \sum_{i \in R_t} (y_i - \hat{y}_{R_t})^2, \tag{3.4}$$

by choosing the best set of partitions, which is normally a unfeasible problem. Therefore, *greedy top-down* algorithms are commonly used [18].

Even tough a tree with a large number of partitions is good for *in sample* predictions, it may not generate good *out of sample* predictors. Hence, what is commonly called "pruning the Tree" is used for regularize the size of the model.

More formally, (3.4) is modified to incorporate a penalization factor responsible for control the depth of the tree (which is the number of terminal nodes). Thus, we have:

$$\min \sum_{t=1}^{T} \sum_{i \in R_t} (y_i - \hat{y}_{R_t})^2 + \alpha T, \tag{3.5}$$

where $T$ is the number of terminal nodes and $\alpha$ is the *tuning/penalty* parameter that controls the growth of the tree.

## 3.2 Smooth Tree

In order to introduce smoothness, we employed the Smooth Tree (ST), in the same spirit of [11], changing the indicator function in (3.2) by a logistic function $H$ defined as:

$$H(x; \gamma, c, s) = \frac{1}{1 + e^{-\gamma(x_s - c)}}, \tag{3.6}$$

where $\gamma$ represents the smoothness of the transition, $c$ is the location of the transition and $s \in \{1, ..., m\}$ represents the variable used to evaluate the logistic function. One can show that if $\gamma \to \infty$, then the logistic function converges to the indicator function as shown in Figure 3.2.

It is important to note that differently from the Regression tree model, where the domain is partitioned, in the ST model, all observations can belong to every possibles terminal nodes with a certain "probability". The prediction for a observation $x$ under the ST model is given by:

$$\hat{f}(x, \{\theta\}_1^T) = \sum_{j=1}^{T} \beta_j \boldsymbol{H}_j(x; \theta_j), \tag{3.7}$$

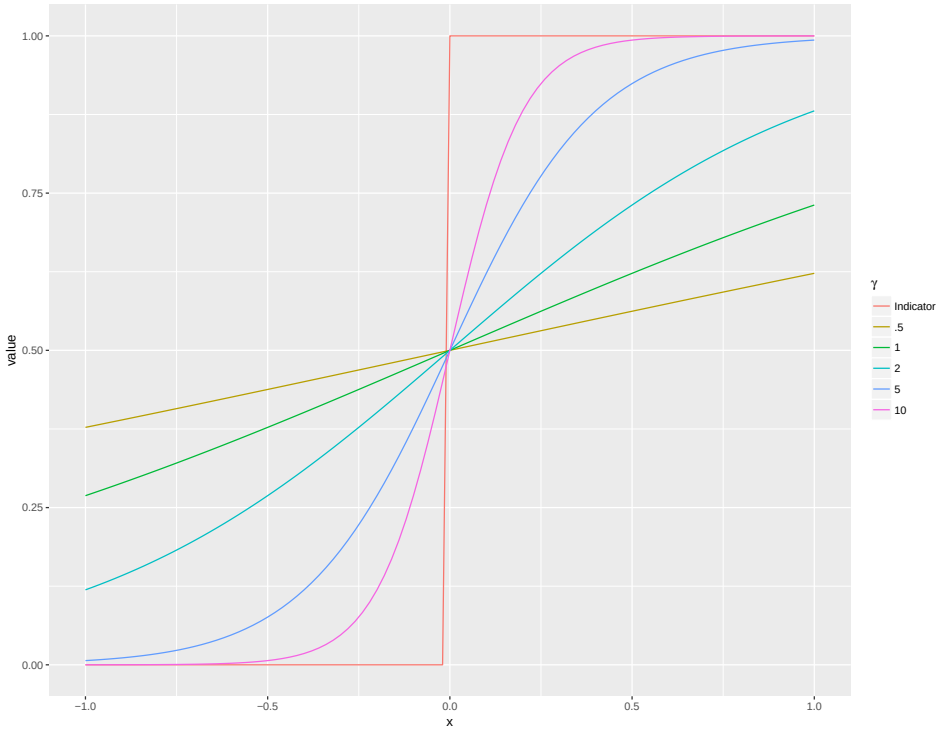


*Figure 3.2:* Example of Logistic Functions

where $\boldsymbol{H}$ is a product of logistic functions $H$ given by (3.6).

One can easily note that $\sum_{j=1}^{T} \boldsymbol{H}_j(x_i;\theta_j) = 1$. Thus, instead of the notion of belonging to a specific partition, the observations have a probabilistic or *fuzzy* interpretation of belonging to a specific terminal node.

## 3.3 Random Forests

Although Tree Models have good *in-sample* performance when predicting non-linear functions, it is remarkably unstable even with pruning procedures [8, 18]. Therefore, models that combine different trees, taking advantage of their non-correlation, can be used in a bagging procedure (boostrap aggregation) to increase *out-of-sample* performance [8].

Random Forests [9] perform bagging of trees that are forced to reduce their correlations due to stochastic procedures introduced in the algorithm. To achieve this, each time that a new node is grown, one needs to search among random selected explanatory variables $p < m$. Normally, in each step, $p \sim \sqrt{m}$ variables are used, where $m$ is the total number of explanatory variables or covariate.

The output of the model will be given by:

$$f_B(x) = \frac{1}{B}\sum_{i=1}^{B} \hat{f}^i(x, \{\theta\}_{j=1}^{T_i}), \tag{3.8}$$

where each $\hat{f}$ is given by (3.2) and $B$ is the total number of boostrap samples.

## 3.4 Gradient Boosting

Boosting is another greedy method from the statistical learning literature to approximate no-linear functions [19, 18] with trees growing sequentially. Therefore, differently from Random Forests algorithm, in a Boosting model each tree is correlated with the others trees.

Following [31], let $\mathcal{F} = \{F : \mathbb{R}^p \to \mathbb{R}\}$ be a set of real-valued functions. We want to find $F^\star$ such that:

$$F^\star = \arg\min_{F \in \mathcal{S}} \mathcal{L}(F),$$

where $\mathcal{L}$ is a function that is convex in $F$.

For problems such as regression or classification, in order to estimate $F^\star$, one can proceed as [19], defining the following form for $\mathcal{L}(F)$:

$$\mathcal{L}(F) = \mathbb{E}_{y,x}[L(y, F(x))], \tag{3.9}$$

where $\mathbb{E}_{y,x}$ is the expectation with respect to the joint distribution of $(y, x)$ and $L$ is a specific loss function that is convex in the second term, such as, the square loss function.

Therefore, we seek to solve:

$$F^\star = \arg\min_{F \in \mathcal{F}} \mathbb{E}_{y,x}[L(y, F(x))]. \tag{3.10}$$

In our case, since we have finite sample, we need to approximate (3.9) by the empirical risk $R_F$ given by:

$$R_F(y, x) = \frac{1}{N}\sum_{i=1}^{N} L(F(x_i), y_i), \tag{3.11}$$

where the function $F$ is restricted to an additive expansion of the form:

$$F(x; \{\rho_m, \theta_m\}_{m=1}^{M}) = \sum_{m=1}^{M} \rho_m h(x; \theta_m). \tag{3.12}$$

where $h$ is a base learner, for instance, a smooth-tree of the form (3.7), and $\theta$ is a finite set of parameters that characterizes each base learner and each base learner can be seen as a direction that minimizes (3.10) in functional space [27].

In order to find $F$ that minimize (3.11), we follow the greedy approach proposed by [19]. The Gradient Boosting algorithm is based on the *steepest-descent* algorithm, where at each iteration of the algorithm, we take a step in the opposite direction of the gradient of the loss function $L$.

The algorithm works as following: assuming that we have computed the algorithm until iteration $m-1$, the gradient at the $m$-th iteration is calculated as

$$u_m(x) = \left. \frac{\partial \mathbb{E}_{x,y}[L(y, F(x))]}{\partial F(x)} \right|_{F=F_{m-1}}, \tag{3.13}$$

under the appropriate regularity conditions [19]

$$u_m(x) = \left. \frac{\partial R_F(y, x)}{\partial F(x)} \right|_{F=F_{m-1}}, \tag{3.14}$$

with

$$F_{m-1}(x) = \sum_{i=0}^{m-1} \rho_i h_i(x). \tag{3.15}$$

Using a two step procedure, first we solve:

$$\theta_m = \arg\min_{\theta} \sum_{i=1}^{N} \big( - u_m(x_i) - h(x_i; \theta) \big)^2, \tag{3.16}$$

then we calculate the step size or line search in the direction of $-u_m$ by solving:

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i; \theta_m)). \tag{3.17}$$

Finally, the updated model and forecast at the $m$-th step are:

$$F_m(x) = F_{m-1}(x) + \rho_m h(x, \theta_m), \tag{3.18}$$

and the predictions of the final model will be given by:

$$\hat{y}_i = F_M(x_i) = f_0 + \sum_{m=1}^{M} \rho_m h(x_i, \theta_m), \tag{3.19}$$

where $M$ is the total number of base learners and $f_0$ is the initial guess. Another feature commonly used in Grandient Boosting is the addition of a *shrinkage* factor $\nu$ in (3.18). In this case, the update equation and prediction are:

$$F_m(x) = F_{m-1}(x) + \nu\rho_m h(x, \theta_m), \tag{3.20}$$

$$\hat{y}_i = F_M(x_i) = f_0 + \sum_{m=1}^{M} \nu\rho_m h(x_i, \theta_m). \tag{3.21}$$

It is worth noticing that the parameter $\nu$ is not used in the estimation of (3.16) and (3.17). However, theoretical and empirical results shown that it is necessary for both convergence and consistency of the gradient boosting [31].

# CHAPTER 4

# Local Vol and Smooth Trees

## 4.1 Problem Statement and Solution

The main purpose of this work is to estimate a local volatility surface capable of replicate the observed market price for the *vanilla* options and replicate the implied volatility surface. Moreover, properties such as smoothness of the estimated local volatility function and robustness of the method used is desired. To estimate the Local Volatility function we will use Gradient Boosting using Smooth Transition Trees as base learner.

The main difference between the proposed method and conventional Gradient Boosting with a generic base learner, is that our methodology makes use of a compound function when calculating the loss function. In a conventional set-up as in (3.10), one have a loss-function calculated in $y$ and $F(x)$.

However, in our case, the variable $y$ is the observed *implied volatility*, and the function that we are estimating is not the prediction of the *implied volatility*, rather it is the local volatility function. Therefore, we needed to make use of the functional $\mathcal{G}$ that connects the local volatility surface with the *implied volatility*. In other words, instead of solving (3.10), we are solving:

$$F^{\star} = \arg\min_{F \in \mathcal{F}} \mathbb{E}_{y,x}[L[y, \mathcal{G}(F, x)]]. \tag{4.1}$$

First, let us define the loss function $L$ as the square-error loss of predicted and observed variable, such that for a given observed vector $y \in \mathbb{R}^N$ and a predicted vector $\hat{y} \in \mathbb{R}^N$:

$$L(y, \hat{y}) = \frac{1}{2}||y - \hat{y}||_2^2 = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{4.2}$$

where $\hat{y} = F(x)$ and $x$ is a set of explanatory variables. Hence, the gradient of $L$ with respect to the model $F$ is:

$$\nabla L = -(y - F(x)) = -(y - \hat{y}). \tag{4.3}$$

The methodology proceeds as follows: first, we observe market prices, strike and maturity for a set of *vanilla* options. Second, we need to calculate what we call "observed implied volatility" of these *vanilla* options by inverting the Black and Scholes formula following (4.3) for example. We consider the observed strikes and maturities as explanatory variables and the implied volatility $\sigma_{LV}$ as the dependent variable.

To start the algorithm first we need a guess for the Local Volatility function $\sigma_0$. Let us call $X$ the set of all possible values for $K$ and $T$, one needs only that $\sigma_0 : X \to \mathbb{R}^+$. One suggestion is to start with a constant local volatility function, for example the historical volatility of the underlying asset.

Let $\mathcal{G}$ be a mapping of the local volatility functional space $\mathcal{F}_{LV}$ to the option's price functional space $\mathcal{F}_c$. With the first guess $\sigma_0$, is possible to calculate

$$\mathcal{G}(\sigma_0) = C_0, \tag{4.4}$$

where the operator $\mathcal{G}(\sigma_0)$ represents solving the Local Volatility PDE (2.8) with $\sigma_{LV} = \sigma_0$. Now, with the estimated price function $C_0$, one should calculate the implied volatility function denoted by

$$\phi_0 = C_{BS}^{-1}(C_0). \tag{4.5}$$

The next step is to calculate the gradient (4.3) and estimate a base learner $\hat{u}_1$ in the opposite direction of the gradient.

$$\begin{aligned} u_1 &= -(\sigma_{IV} - \phi_0), \\ -\hat{u}_1 &= F(x), \end{aligned} \tag{4.6}$$

where $x_{i,j} = \{K_i, \tau_j\}$ is the strikes and maturities of the finite number of observed options. The exactly methodology used to estimate the $F$ function is described in Section 4.4. Now, one needs to find the parameter $\rho_1$ such that:

$$\rho^\star = \min_{\rho \in \mathbb{R}} ||\sigma_{IV} - \phi_1(\rho)||_2, \tag{4.7}$$

where

$$\phi_1(\rho) = C_{BS}^{-1}(\mathcal{G}(\sigma_0 + \rho\hat{u}_1)). \tag{4.8}$$

The algorithm continues making the updating of the estimated local volatility by updating

$$\sigma_1 = \sigma_0 + \nu\rho_1\hat{u}_1, \tag{4.9}$$

where $\nu$ is the *shrinkage* factor that controls for *over-fitting* and performs regularization [19]. The parameter $\nu$ is also called the *learning rate* in the *machine learning* literature.

Finally, one can recalculate the mapping $G(\sigma_1) = C_1$ with the updated local volatility function solving (2.20) and repeat the steps above, recalculating the price and implied volatility.

The algorithm is organized as a pseudo code in 1:

**Algorithm 1:** BooST for Local Volatility

**Data:** Observations of $K, T, C_{mkt}$

**Result:** $\hat{\sigma}_{LV} = \sigma_M$, $\hat{\sigma}_{IV} = \phi_M$ and $C_M$

initialize $\sigma_0$;

calculate $\sigma_{IV} = C_{BS}^{-1}(C_{mkt})$;

calculate $C_0 = \mathcal{G}(\sigma_0)$;

calculate $\phi_0 = C_{BS}^{-1}(C_0)$;

**for** *m=1,...,M* **do**

- make $u_m = \sigma_{IV} - \phi_{m-1}$;
- grow a ST in $u_m$, such that $\hat{u}_m = \sum_{t=1}^{T} \beta_k \boldsymbol{H}(x; \boldsymbol{\theta}_k)$ ;
- make $\rho_m = \arg\min_\rho ||\sigma_{IV} - \phi_m(\rho)||$ where $\phi_m(\rho) = C_{BS}^{-1}(\mathcal{G}(\sigma_{m-1} + \rho\hat{u}_m))$;
- update $\sigma_m = \sigma_{m-1} + \nu\rho_m\hat{u}_m$;

**end**

In the next sections one can find the specific details of the methods used to solve the steps of Algorithm 1.

## 4.2 Solving the Local Volatility PDE

It is known that parabolic PDE can be solved using numerical methods, for instance finite difference methods (FDM) in a explicit way, using the forward difference, or in a implicit way is the backward difference. See for example [3]. Although it has some critics due to it's stability [14], the Crank-Nicholson Scheme has been widely used for pricing options [29], [32], [30].

However, in this work, every time that a PDE needed to be solved, we wrote the stochastic counterpart given by (2.11) and used a Monte Carlo method. To do that, we proceeded by simulating several paths for the asset, explicitly calculating the payoff of the derivative of each simulation, and then calculating the discounted price.

## 4.3 Inverting the BS formula

Once we have the estimated/observed option prices, the inversion to discover the estimated/observed implied volatility was done using the Brent algorithm. The optimization problem is described in (4.10). The relative quadratic error was used to give more stability to the objective function. Moreover, to keep the search in the feasible space, it was used the following parametrization: $\sigma_{IV} = \exp(p)$, where $p$ is the parameter given by the algorithm.

$$\sigma_{IV} = \underset{p \in \mathbb{R}}{\arg\min} \left( \frac{C(K,T) - C^{BS}(S_t, r, K, T, t, \exp(p))}{C(K,T)} \right)^2. \tag{4.10}$$

## 4.4 Fitting Smooth Trees

A crucial step in the algorithm is the growth of base learner at each step of the gradient boosting algorithm. As mentioned before, in our approach each base learner is a smooth tree in the form of product of logistic functions. In our algorithm, the number of terminal nodes of each tree is defined *a priori* and set equal to five. The methodology follows the same idea than conventional CART models, which is a greedy procedure.

First, we receive the sequence of observations $\{y_i, \tau_i, K_i\}$ where $y$ is the gradient $\phi_m$ at the $m$-step of the algorithm and $\tau, K$ are the maturity and strike of the options. The algorithm starts looking for the best variable $K, \tau$, splitting point $c$ and coefficients $\beta_0, \beta_1$ such that:

$$\theta_1^\star = \underset{c,s,\beta_0,\beta_1 \in \mathbb{R}^+}{\arg\min} \sum_{i=1}^{N} (y_i - [\beta_1 H(x;\gamma,s,c) + \beta_0(1 - H(x;\gamma,s,c))])^2, \tag{4.11}$$

where $H$ is the logistic function (3.6) and $N$ is the number of observations. The parameter $\gamma$ is randomized for each step of the algorithm and $s$ indicates the variable used as splitting variable to compute the logistic function value.

To create the second node, we need to apply at each terminal node the procedure described before. Since we have one split, we have to solve the optimization problem for two terminal nodes. For the first one, the problem can be formulated as:

$$\begin{aligned}\theta_2^\star = \underset{c,s,\beta_2,\beta_3 \in \mathbb{R}^+}{\arg\min} \sum_{i=1}^{N} (y_i - [\beta_1 H_1(x;\theta_1^\star) + \beta_2(1 - H_1(x;\theta_1^\star)(1 - H_2(x;\gamma,s,c))) \\ + \beta_3(1 - H_1(x;\theta_1^\star))H_2(x;\gamma,s,c)])^2,\end{aligned} \tag{4.12}$$

and for the second possibility we have:

$$\begin{aligned}\theta_2^\star = \underset{c,s,\beta_2,\beta_3 \in \mathbb{R}^+}{\arg\min} \sum_{i=1}^{N} (y_i - [\beta_2 H_1(x;\theta_1^\star)(1 - H_2(x;\gamma,s,c)) + \beta_3 H_1(x;\theta_1^\star)H_2(x;\gamma,s,c) \\ + \beta_0(1 - H_1(x;\theta_1^\star))])^2,\end{aligned} \tag{4.13}$$

where $\theta_1^\star$ are the parameters $(s,\gamma,\beta_0,\beta_1)$ defined in the first split. Finally, the second split will be in the terminal node with the lower objective value. The procedure continues until we have a total of five terminal nodes (4 splits).

Since at each step we need to solve one optimization problem in each terminal node, in order to estimate a smooth tree with 5 terminal nodes, we need to solve 10 optimization problems in each step of the algorithm. At each optimization, the parameter $s$ can be found solving the problem for both strike and maturity, and the parameters $c, \beta_m, \beta_{m+1}$ were estimated using BFGS.

As mentioned before, the methodology is quite similar to conventional CART models. The main difference is that we still need to use all observations when estimating each terminal node, whereas in CART models the number of observations used decreases in each step.

## 4.5 Data Generating Process

In order to analyze the performance of the proposed model, first we generated synthetic options prices to simulate observed prices. The first steep is to consider the SDE of a local volatility model as in (2.11). We simulated an asset price such that $S_0 = 1$, in a market with risk-free rate $r = 0$ and a local volatility surface based in [2]. The discretized version of the SDE using the Euler-Maruyama scheme is presented in (4.14).

$$\begin{aligned} \Delta \ln S_{t_i} &= (r - \frac{\sigma(S_{t_i}, t_i)^2}{2})\Delta t + \sigma(S_{t_i}, t_i)\sqrt{\Delta t}\epsilon_i, \\ \ln S_{t_{i+1}} &= \ln S_{t_i} + \Delta \ln S_{t_i}, \\ \epsilon &\overset{i.i.d}{\sim} N(0,1), \end{aligned} \tag{4.14}$$

where $i = 1, ..., M$, $M = 252$ is the maximum number of iterations in time, $\Delta t = \frac{1}{M}$, $\sigma$ is the local volatility function described in (4.15) with annualized volatility.

$$\sigma(S,t) = \begin{cases} \frac{2}{5} - \frac{4}{25}\exp(-0.5\tau)\cos(0.8\pi y), & \text{if } -\frac{2}{5} \leq y \leq -\frac{2}{5}, \\ 2/5 & \text{otherwise,} \end{cases} \tag{4.15}$$

where $y$ is given by (4.16).

$$y = \ln \frac{Se^{-rt}}{S_0}. \tag{4.16}$$

We simulated $N$ paths using the equations described previously. Taking the expectation of the payoff function using the simulated prices give us the option price at time $M$. Moreover, discounting the risk-free rate one can get the option prices at any time $i$.

Thus, since we are in a market with $r = 0$, we have that:

$$C_j = \mathbb{E}^{\mathbb{Q}}[(S_{T_j} - K_j)^+] \approx \frac{1}{N}\sum_{i=1}^{N}(S^i_{T_j} - K_j)^+, \tag{4.17}$$

where $N$ is the total number of simulations, $S^i_{T_j}$ is the $i$-th simulated price at time $T_j$, $K_j$ and $T_j$ is the strike and maturity of option $j$, respectively.

# CHAPTER 5

# Numerical Results

## 5.1 Simulated data

In order to generate observed option prices, we fixed the strikes varying from $K = 0.8, 0.85, \ldots, 1.2$ and maturities varying from $\tau = \frac{10}{30}, \frac{15}{30}, \ldots, 1$, giving the total number of 45 observed options. We use $N = 10^6$ and $M = 252$. The number $N$ used in the creation of the synthetic data is greater than the number of simulation used to test the method. Figures 5.1 and 5.2 shows the simulated path for 100 realizations of the process. Moreover, to estimate the local volatility surface, it was defined a set of derivatives (*vanilla* Calls).

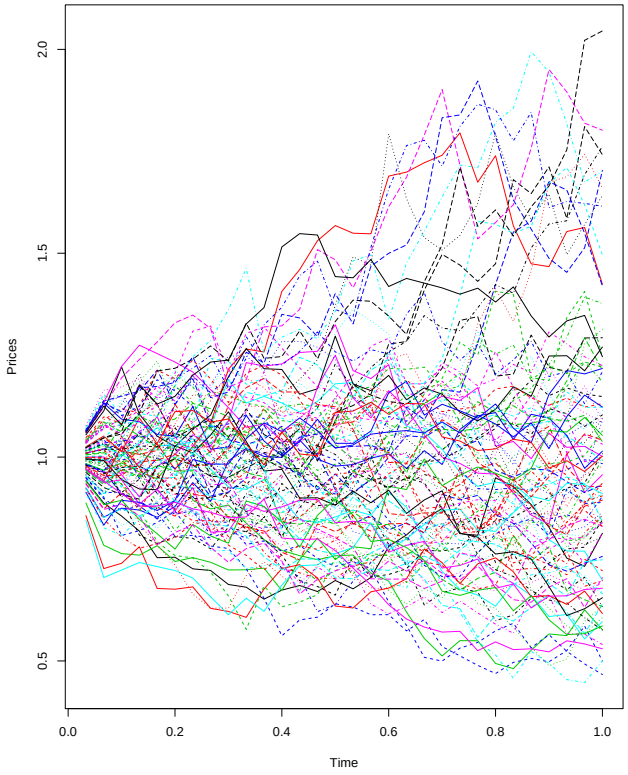


*Figure 5.1:* Example of simulated prices

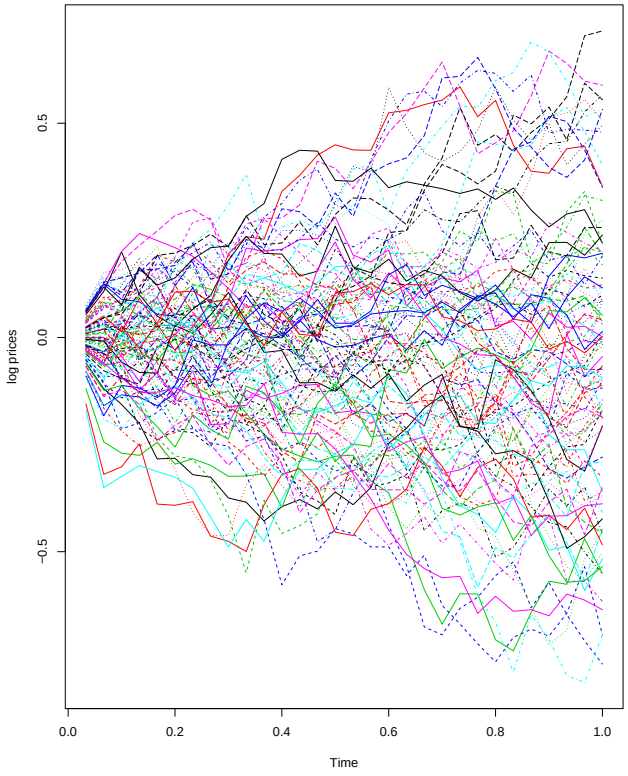


*Figure 5.2:* Example of simulated logprices

In Figure 5.3 is possible to check the value of the local volatility function for each derivative used in the numerical experiment. The complete local volatility surface is displayed at Figure 5.4. In the Figure, the $z$ axis is the volatility, the $y$ axis is the strike and $x$ is the maturity. It is possible to see that the surface is not symmetric with respect to the ATM option. In addition, the surface increases with maturity and when the derivative is *in the money* or *out of the money*.

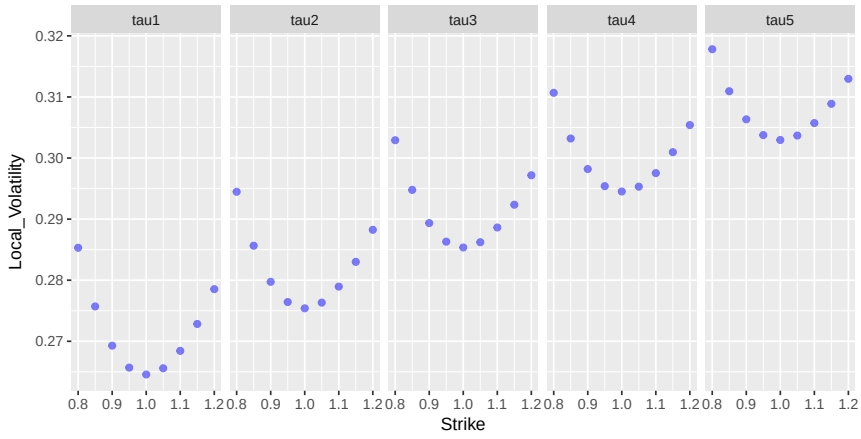


*Figure 5.3:* Local Volatility Surface

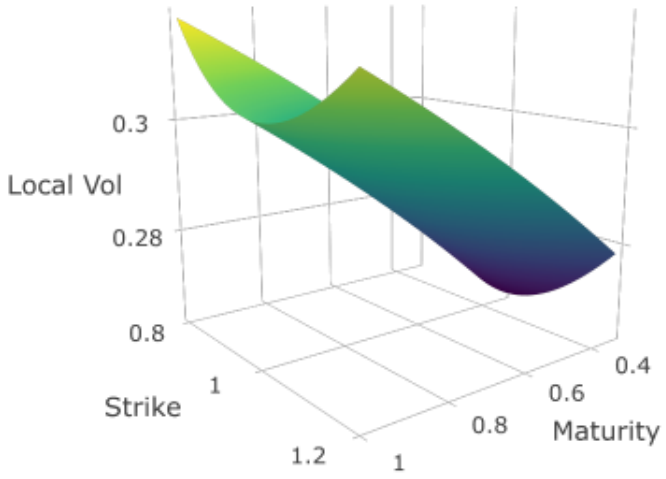


*Figure 5.4:* Local Volatility surface

At Figure 5.5 we have the calculated prices for each derivative that will be used by the proposed model. The full pricing surface is exposed shown in Figure (5.6). Naturally, when maturity gets shorter, the price curve gets closer to the payoff function of the derivative.

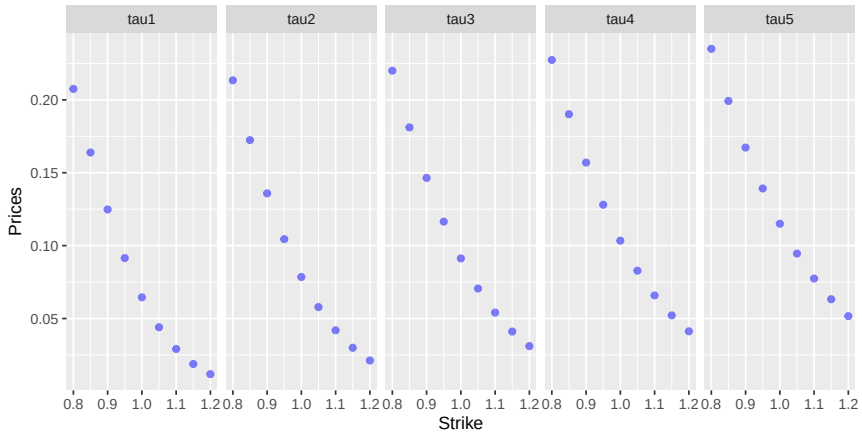


*Figure 5.5:* Calculated prices of the derivatives used

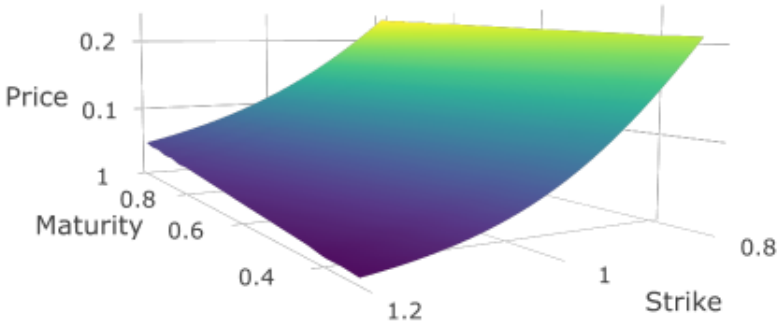


*Figure 5.6:* Option prices surface

Applyying the methodology explained in Subsection 4.10, ie. inverting the Black and Scholes formula, it was possible to construct the Implied Volatility surface. At 5.7 one can observe the implied volatility for each option observed. The full plot of the Implied Volatility surface (in a finer grid) is exposed at 5.8.

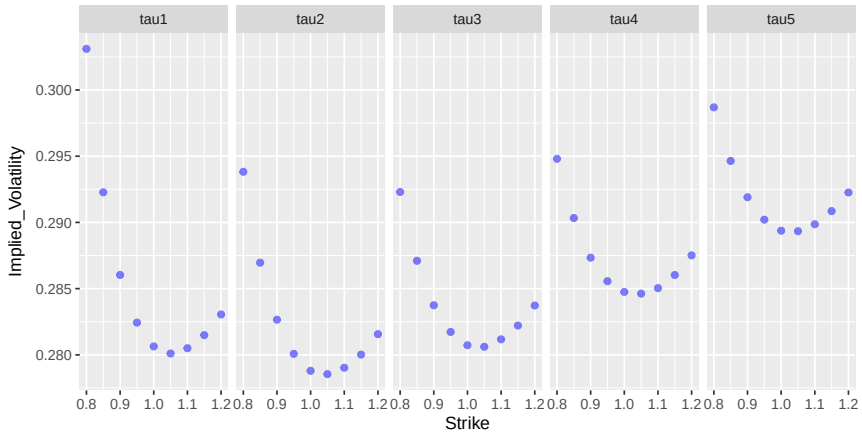


*Figure 5.7:* Implied volatility of each derivative used

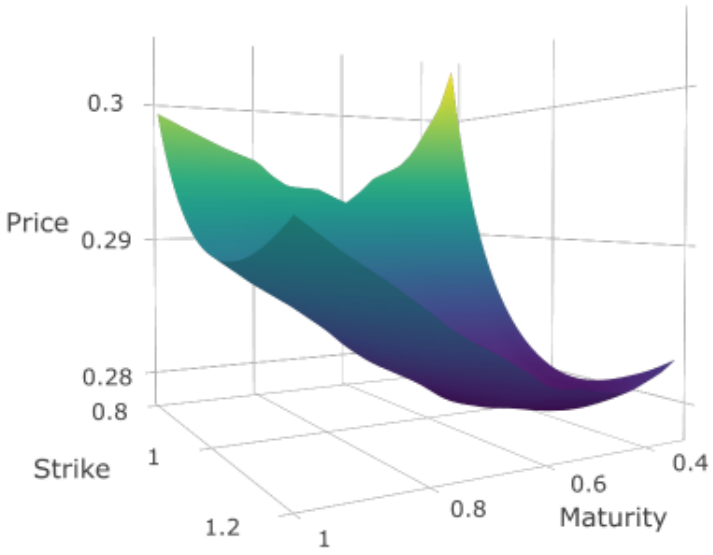


*Figure 5.8:* Implied volatility surface

Using the implied volatility of each observed option it was possible to apply Algorithm 1. At each step of Algorithm 1, we can compute the RMSE up to that point. Therefore, each time a tree is grown, we can observe the *in sample* RMSE and see how the graphic is decaying. In Figure 5.9 we can check the rmse of predicted implied volatility as the number of tree grows.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\sigma_{IV,i} - \hat{\sigma}_{IV,i})^2} \tag{5.1}$$

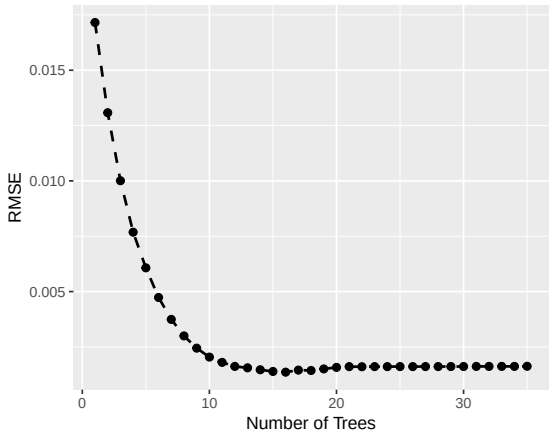


*Figure 5.9:* Root mean square as function of the number of trees

It is evident that the algorithm converges after a relative small number of trees. There is no advantage in terms of RMSE of a model with more than 20 trees. In Figure 5.10 one can see a comparison between the local volatility estimated using a different number of trees and the true local volatility of the options (black dots).

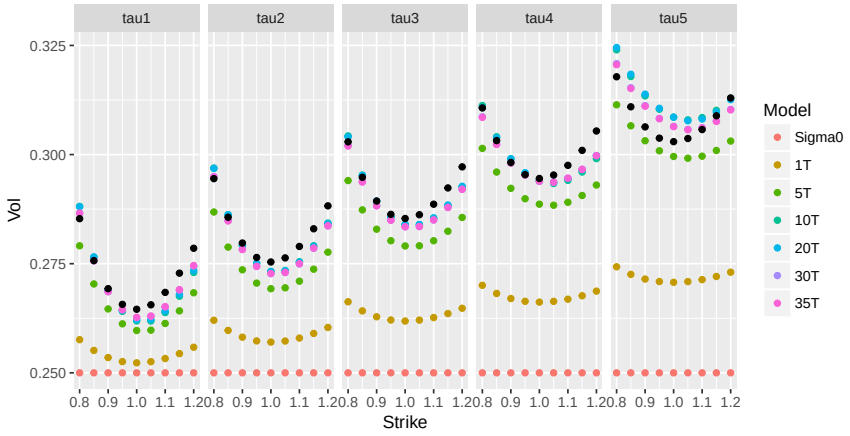


*Figure 5.10:* Evolution of the estimated Local Volatility

In Figure 5.10 one can check the implied volatility generated by different number of trees and compare with the true surface (black dots).

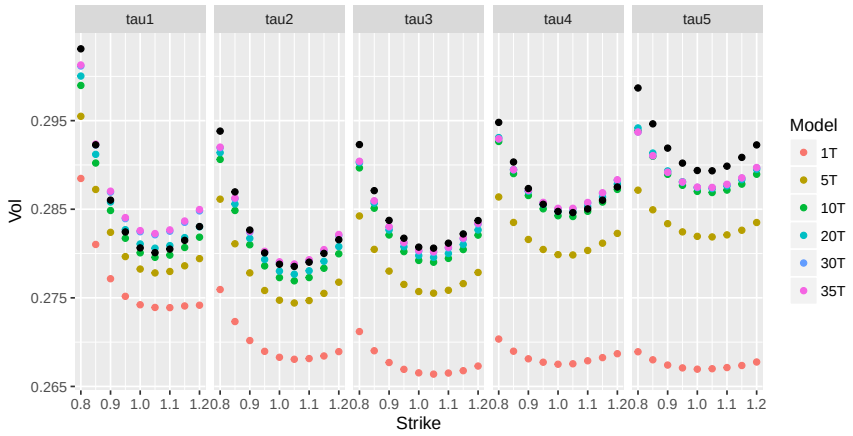


*Figure 5.11:* Evolution of the estimated Implied Volatility

It is common in the statistical learning literature to use an empirical technique to choose the appropriate size of the model [18]. One approach, is to select the number of trees that minimizes the RMSE, which is 16 trees in this example. In Figure 5.12 we can see the comparison between the true local volatility for each observed option, which is unknown for the model, and the estimated local volatility for each option by the model. Figure 5.13 compares the full slocal volatility surface of the estimated model and the true function (2.21).

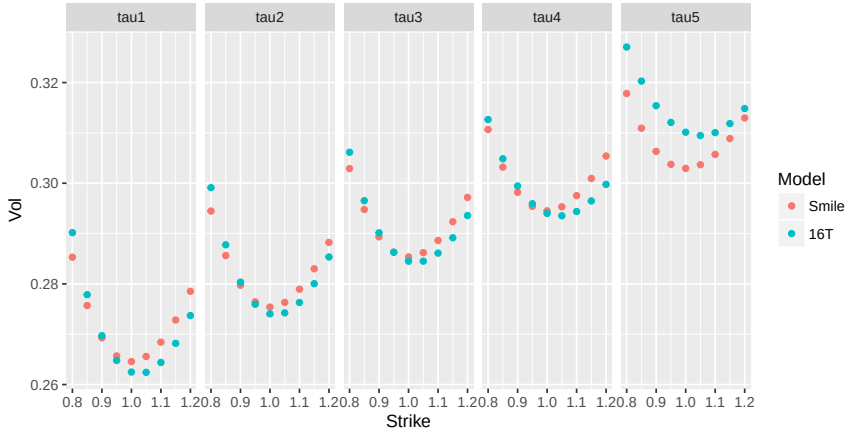


*Figure 5.12:* Estimated Local Volatility for the model with the least rmse

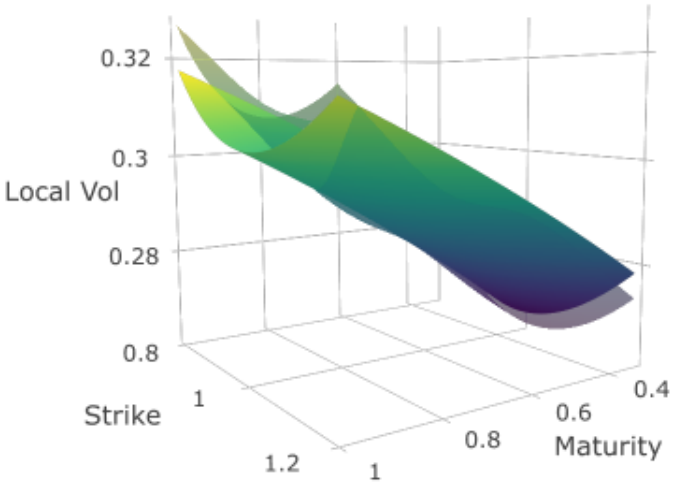


*Figure 5.13:* Comparison of the Local Volatility surfaces

Using the estimated local volatility surface, which is in fact a continuous function rather than estimated values in a specific grid such as in numerical methods for solving PDE, we computed the option prices using Monte Carlo simulations. For the model with sixteen trees, the maximum error is less than 0.1%. In Figure 5.15 we have a comparison between the prices generated using (2.21) and the prices generated with the estimated model. We can see that even for strikes and maturities not used in the estimation, the approximation keeps the same order of error.

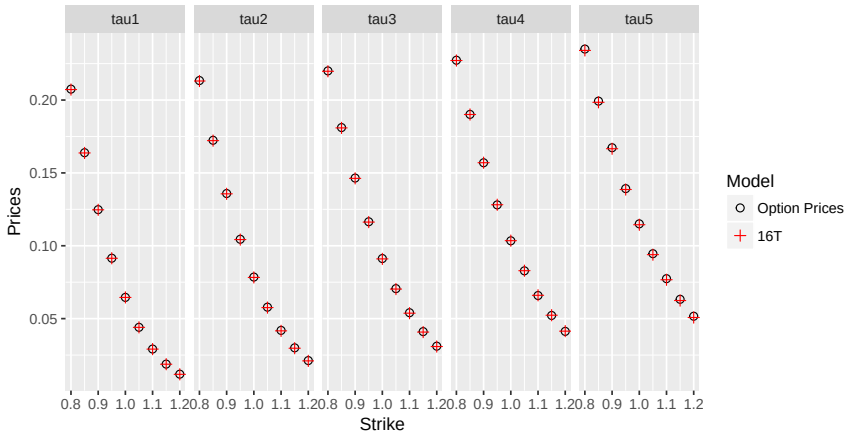


*Figure 5.14:* Estimated Option Value for the model with the least rmse

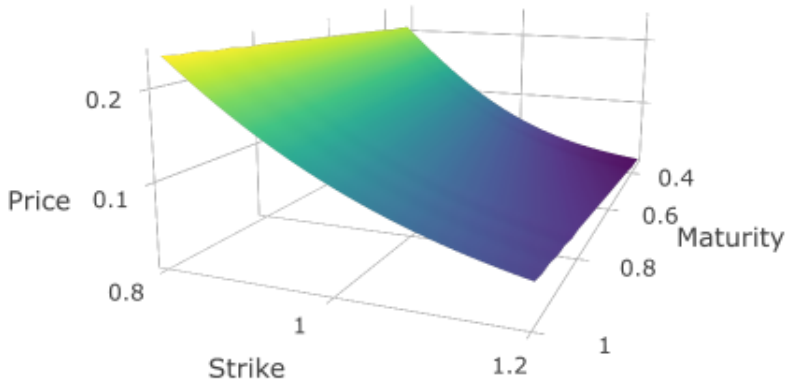


*Figure 5.15:* Comparison of the Option prices surfaces

Using the Black  Scholes formula and the estimated prices we compared the Implied Volatility surface generated by the model and by the true function. In Figure (5.16) we have a comparison for the observed options and Figure 5.17 shows the comparison of the full surface.

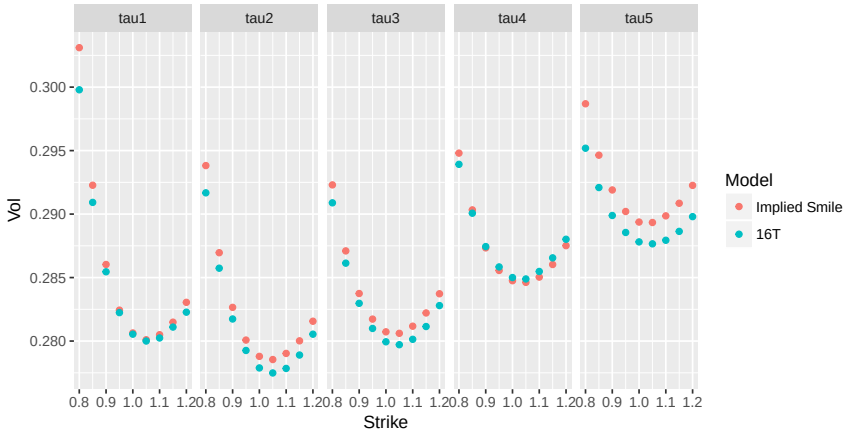


*Figure 5.16:* Estimated Implied Volatility for the model with the least rmse

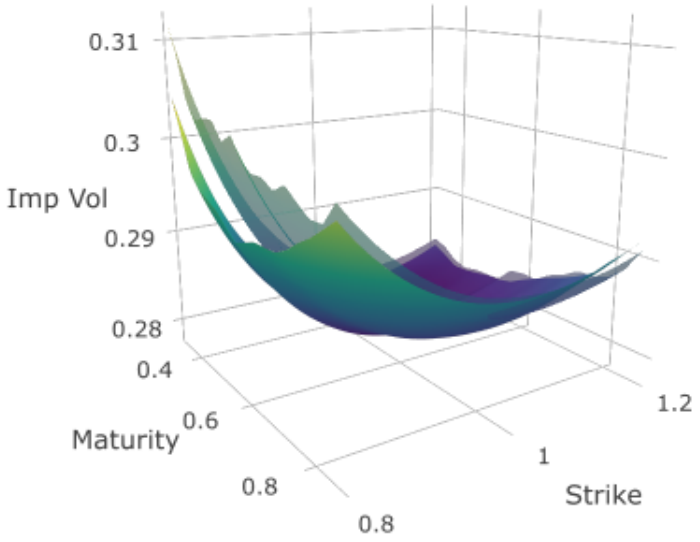


*Figure 5.17:* Comparison of the Implied Volatility surfaces

In Tables 5.1, 5.2 and 5.3 we can check error measurements for local volatility, option prices and implied volatility respectively. It worth to notice that the prices where replicated almost exactly. Furthermore, the highest absolute error of the local volatility and implied volatility are both less than 1%. We attributed the bigger difference in the implied volatility surface due to instability in the inversion method used for the Black Scholes formula.

*Table 5.1:* Local Volatility

| Measures | values |
|---|---|
| MSE | 1.61E-05 |
| RMSE | 4.01E-03 |
| MAE | 3.19E-03 |
| MAPE | 1.08E-02 |
| $\|\|.\|\|_\infty$ | 9.35E-03 |

*Table 5.2:* Option Prices

| Measures | values |
|---|---|
| MSE | 1.51E-07 |
| RMSE | 3.88E-04 |
| MAE | 2.98E-04 |
| MAPE | 3.91E-03 |
| $\|\|.\|\|_\infty$ | 9.32E-04 |

*Table 5.3:* Implied Volatility

| Measures | values |
|---|---|
| MSE | 1.85E-06 |
| RMSE | 1.36E-03 |
| MAE | 1.09E-03 |
| MAPE | 3.78E-03 |
| $\|\|.\|\|_\infty$ | 3.50E-03 |

## 5.2 Exotic Options

### 5.2.1 Asian Call

In order to test the robustness of the model for pricing exotic options, first we used the model for pricing Asian Call options. Recall that the price of a Asian Call exotic option is given by the following payoff function:

$$C(K,T) = (S_T^\star - K)^+, \tag{5.2}$$

where $S_T^\star$ is the price given by:

$$S_T^\star = \frac{1}{T}\int_0^T S_u du. \tag{5.3}$$

The price $S_T^\star$ can be seen as a medium price of the asset $S$ in a continuous perspective. Therefore, we need to calculate a discrete approximation given by:

$$S_T^\star = \frac{1}{T}\sum_{i=1}^{I}(S_i\Delta t) = \frac{1}{I}\sum_{i=1}^{I} S_i. \tag{5.4}$$

where $\Delta t$ is the discrete time interval, $I$ is the total number of partitions and $S_i$ is the price of the underlying asset at time $i$.

To calculate the synthetic prices, we generated $10^6$ prices paths using the local volatility function given by (4.15). It was calculated the option prices for two different type of maturities $\tau = \{0.5, 1\}$ and strikes varying from 0.8 to 1.2 by 0.05.

The simulated prices for the exotic option using the "true" local volatility function was calculated with the following expression:

$$C_j = \mathbb{E}^{\mathbb{Q}}[(S_{T_j}^\star - K_j)^+] = \frac{1}{N}\sum_{i=1}^{N}(S_{T_j}^{\star i} - K_j)^+, \tag{5.5}$$

where $N$ is the total number of simulations, $S_{T_j}^{\star i}$ is the $i$-th simulated price at time $T_j$ calculated using (5.4), $S_{T_j}^i$ is the $i$-th simulated price at time $T_j$, $K_j$ and $T_j$ is the strike and maturity of option $j$, respectively.

To estimate the prices of the Asian Call options with the proposed model, we used a Monte Carlo simulation with the discrete SDE given by (4.14), 500.000 paths, starting price $S_0 = 1$ and a local volatility function given by the estimated model with 16 trees. The estimated payoff was calculated using (5.9)..

In Figure 5.18 it is possible to check the fitted values for the adjusted model. Moreover, some error measurements are show in Table 5.4.

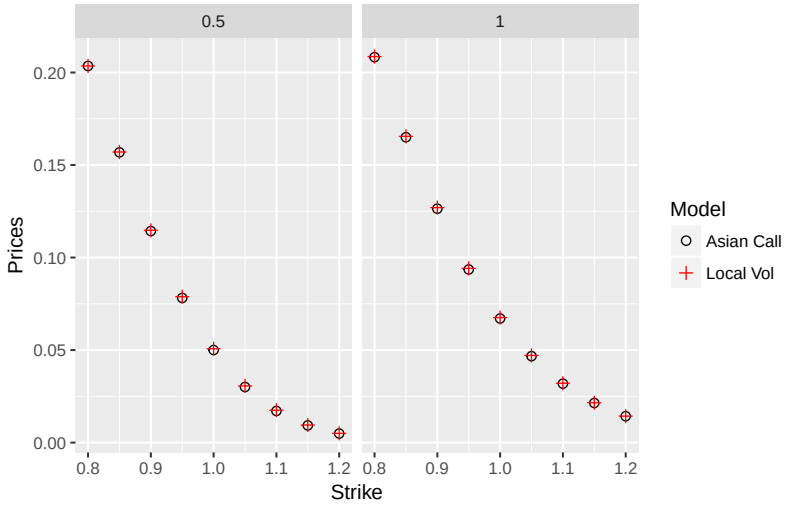


*Figure 5.18:* Asian Calls fitted values

*Table 5.4:* Error for Asian Call prices

| Measures | values |
|---|---|
| MSE | 1.71E-07 |
| RMSE | 4.14E-04 |
| MAE | 3.60E-04 |
| MAPE | 8.91E-03 |
| MAX | 6.64E-04 |

### 5.2.2 Float Strike Call

Another type of exotic option tested in the work is the Float Strike Call. For this option, the payoff function is given by:

$$C(T) = (S_T - S_T^\star)^+, \tag{5.6}$$

where $S_T^\star$ is the price given by:

$$S_T^\star = \frac{1}{T}\int_0^T S_u du. \tag{5.7}$$

In this case, the option does not have a strike parameter, since it is given by (5.7), which is random and depends of the path of the price. The bigger the maturity, the bigger the uncertainty of $S_T^\star$ and the bigger the price of the option.

Again, we constructed the "true" simulated prices using $10^6$ paths and local volatility function given by (4.15). For each maturity, the simulated option price using the "true"

local volatility function was calculated using:

$$C_j = \mathbb{E}^{\mathbb{Q}}[(S_{T_j} - S^{\star}_{T_j})^+] = \frac{1}{N}\sum_{i=1}^{N}(S^i_{T_j} - S^{\star i}_{T_j})^+, \tag{5.8}$$

where $N$ is the total number of simulations, $S^i_{T_j}$ is the $i$-th simulated price at time $T_j$, $T_j$ is the maturity of option $j$ and $S^{\star i}_{T_j}$ is the $i$-th price calculated using (5.4).

To estimate the price given by our model, we used 500.000 simulated with (4.14) and local volatility function given by the model with 16 trees. Then, to estimated prices for the Float Strike options we used (5.8). In Figure 5.19 is exposed the comparison of the fitted values with the synthetic observed prices.

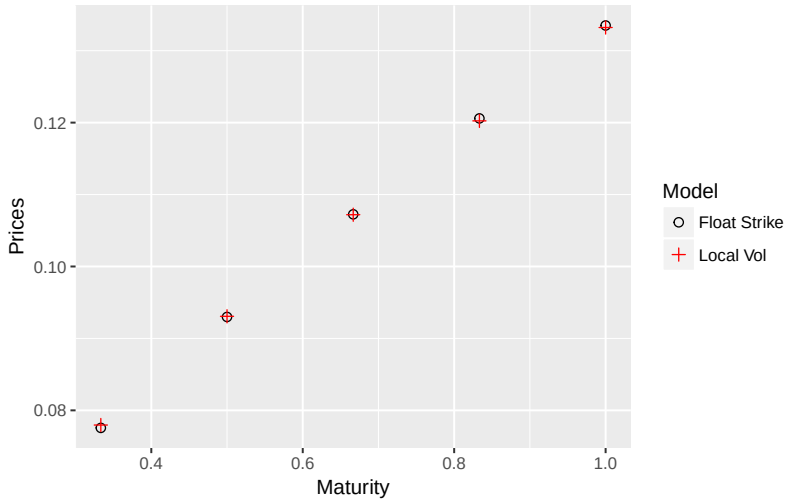


*Figure 5.19:* Float Strike Call fitted values

In Table 5.5 one can check the error metrics of the fitted values.

*Table 5.5:* Error for Float Strike Call prices

| Measures | values |
|---|---|
| MSE | 7.32E-08 |
| RMSE | 2.70E-04 |
| MAE | 2.33E-04 |
| MAPE | 2.31E-03 |
| MAX | 4.00E-04 |

### 5.2.3 Barrier

Barrier options commonly have in addition to strike and maturity the barrier price, that works as a another boundary condition in addition to the payoff function at maturity.

The most common types are quickly described below:

- Up and Out - Asset prices starts below the barrier price. If the price hits the barrier the payoff is zero;
- Up and In - Asset prices starts below the barrier price. If the price does not hit the barrier the payoff is zero;
- Down and Out - Asset prices starts above the barrier price. If the price hits the barrier the payoff is zero;
- Up and In - Asset prices starts above the barrier price. If the price does not hit the barrier the payoff is zero.

Therefore, Barrier options is another path dependent derivative. In this example we adjusted the price of a call option with strike $K$, maturity $T$ and Up and Out barrier price $B$. Intuitively, one can expect that the lower the price of the barrier, the lower the price of the option, since the higher the probability of the asset price $S$ to hit the barrier price $B$. The payoff will be given by:

$$C_j = \mathbb{E}^{\mathbb{Q}}[(S^{\star}_{T_j} - K_j)^{+}\mathbb{1}(S_t < B)] = \frac{1}{N}\sum_{i=1}^{N}(S^{\star i}_{T_j} - K_j)^{+}\mathbb{1}(S^i_t < B), \tag{5.9}$$

where $\mathbb{1}(S_t < B)$ is the indicator function equals to one if $S_t < B, \forall \quad 0 \leq t \leq T$, and zero otherwise.

We tested the adjusted local volatility for two types of maturities $\tau = \{.5, 1\}$, different types of barrier prices and different strikes. In Figure 5.20 we can see the fitted values for different strikes and barrier prices for $\tau = .5$. In Figure 5.21 we have the prices for different strikes and barrier prices for $\tau = 1$.

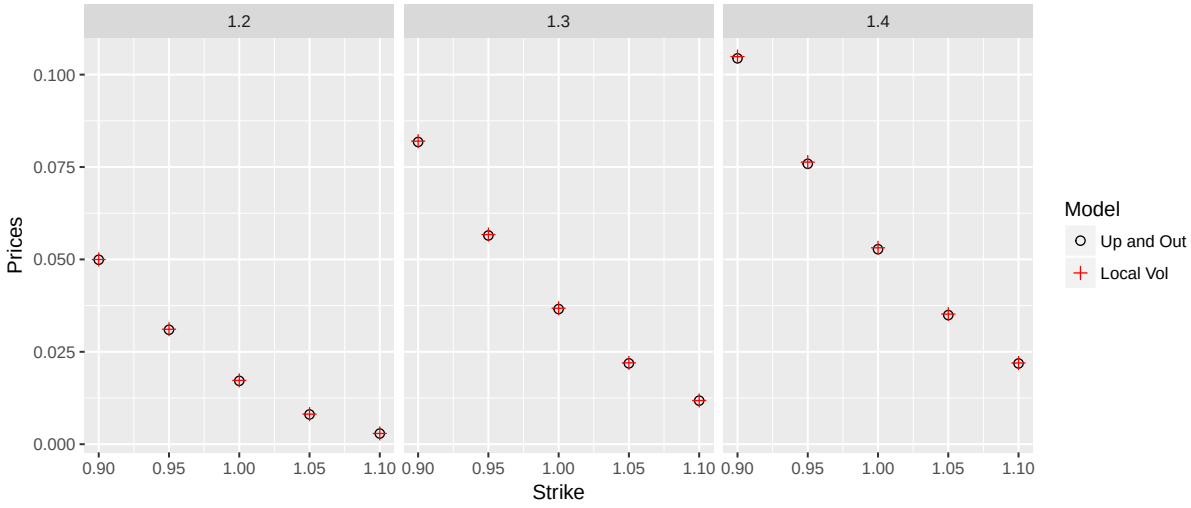


*Figure 5.20:* Up and Out Barrier with 0.5 maturity

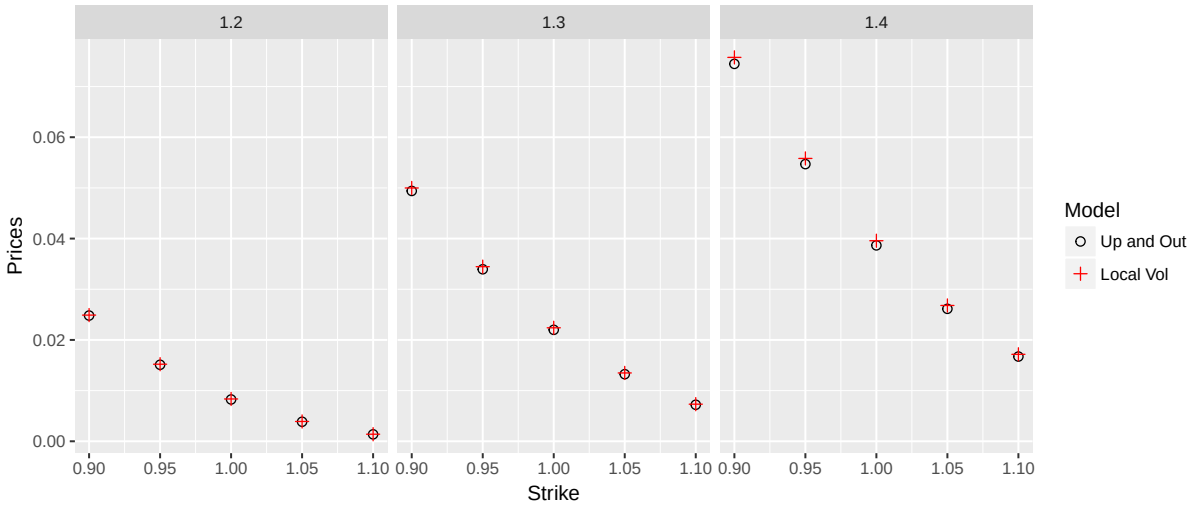


*Figure 5.21:* Up and Out Barrier with 1 maturity

The error measures are exposed in Table 5.6 and 5.7 for $\tau = .5$ and $\tau = 1$ respectively.

*Table 5.6:* Up and Out Maturity = .5

| Measures | values |
|---|---|
| MSE | 5.98E-08 |
| RMSE | 2.45E-04 |
| MAE | 2.01E-04 |
| MAPE | 5.68E-03 |
| MAX | 4.72E-04 |

*Table 5.7:* Up and Out Maturity = 1

| Measures | values |
|---|---|
| MSE | 3.50E-07 |
| RMSE | 5.91E-04 |
| MAE | 4.45E-04 |
| MAPE | 1.68E-02 |
| MAX | 1.28E-03 |

# CHAPTER 6

# Conclusion

The proposed method, which uses gradient boosting and smooth transition trees, was capable of generating correct option prices with almost zero errors. The true local volatility surface was estimated and the implied volatility surface generated by the method had a good prediction performance even for non observed strikes and maturities, with mean absolute error less than 1%.

Furthermore, the estimated local volatility surface could be used to pricing exotic options. Another advantage of the technique is the possibility to compute the convexity by explicit calculating the derivative of the local volatility surface with respect to the strike.

The main drawback of the method is the time taken to solve the PDE with the local volatility function using Monte Carlo. Moreover, sometimes we could observe some instability inverting the options prices in order to get the implied volatility values, specially for short maturities and very OTM options.

It might be useful to solve directly the PDE that connects the local volatility surface with the implied volatility surface as described in (2.26). Although this last approach involves solving a nonlinear PDE, it avoids the necessity of inverting the Black Scholes's Formula and the use of Monte Carlo.

# Bibliography


[1] Fernando Antonio Lucena Aiube, *Modelos quantitativos em finanças com enfoque em commodities*, Bookman Editora, 2013.

[2] Vinicius Albani, Adriano De Cezaro, and Jorge P Zubelli, *Convex regularization of local volatility estimation*, International Journal of Theoretical and Applied Finance **20** (2017), no. 01, 1750006.

[3] William F Ames, *Numerical methods for partial differential equations*, Academic press, 2014.

[4] Marco Avellaneda, Craig Friedman, Richard Holmes, and Dominick Samperi, *Calibrating volatility surfaces via relative-entropy minimization*, Applied Mathematical Finance **4** (1997), no. 1, 37–64.

[5] Louis Bachelier, *Théorie de la spéculation*, Gauthier-Villars, 1900.

[6] Mario Bertero, *Linear inverse and iii-posed problems*, Advances in electronics and electron physics, vol. 75, Elsevier, 1989, pp. 1–120.

[7] Fischer Black and Myron Scholes, *The pricing of options and corporate liabilities*, Journal of political economy **81** (1973), no. 3, 637–654.

[8] Leo Breiman, *Bagging predictors*, Machine learning **24** (1996), no. 2, 123–140.

[9] ———, *Random forests*, Machine learning **45** (2001), no. 1, 5–32.

[10] ———, *Classification and regression trees*, Routledge, 2017.

[11] Joel Correa Da Rosa, Alvaro Veiga, and Marcelo C Medeiros, *Tree-structured smooth transition regression models*, Computational Statistics & Data Analysis **52** (2008), no. 5, 2469–2488.

[12] A De Cezaro, O Scherzer, and JP Zubelli, *Convex regularization of local volatility models from option prices: Convergence analysis and rates*, Nonlinear Analysis: Theory, Methods & Applications **75** (2012), no. 4, 2398–2415.

[13] Emanuel Derman and Iraj Kani, *Riding on a smile*, Risk **7** (1994), no. 2, 32–39.

[14] Daniel J Duffy, *A critique of the crank nicolson scheme strengths and weaknesses for financial instrument pricing*, The Best of Wilmott (2002), 333.

[15] Bernard Dumas, Jeff Fleming, and Robert E Whaley, *Implied volatility functions: Empirical tests*, The Journal of Finance **53** (1998), no. 6, 2059–2106.

[16] Bruno Dupire, *Pricing and hedging with smiles*, Mathematics of derivative securities **1** (1997), no. 1, 103–111.

[17] Herbert Egger and Heinz W Engl, *Tikhonov regularization applied to the inverse problem of option pricing: convergence analysis and rates*, Inverse Problems **21** (2005), no. 3, 1027.

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics New York, 2001.

[19] Jerome H Friedman, *Greedy function approximation: a gradient boosting machine*, Annals of statistics (2001), 1189–1232.

[20] Jim Gatheral, *The volatility surface: a practitioner's guide*, vol. 357, John Wiley & Sons, 2011.

[21] Steven L Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, The review of financial studies **6** (1993), no. 2, 327–343.

[22] Norbert Hilber, Oleg Reichmann, Christoph Schwab, and Christoph Winter, *Computational methods for quantitative finance: Finite element methods for derivative pricing*, Springer Science & Business Media, 2013.

[23] George J Jiang and Yisong S Tian, *The model-free implied volatility and its information content*, The Review of Financial Studies **18** (2005), no. 4, 1305–1342.

[24] Nabil Kahalé, *An arbitrage-free interpolation of volatilities*, Risk **17** (2004), no. 5, 102–106.

[25] Siem Jan Koopman, Borus Jungbacker, and Eugenie Hol, *Forecasting daily variability of the s&p 100 stock index using historical, realised and implied volatility measurements*, Journal of Empirical Finance **12** (2005), no. 3, 445–475.

[26] Roger W Lee, *Implied volatility: Statics, dynamics, and probabilistic interpretation*, Recent advances in applied probability, Springer, 2005, pp. 241–268.

[27] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean, *Boosting algorithms as gradient descent*, Advances in neural information processing systems, 2000, pp. 512–518.

[28] Attilio Meucci, *Risk and asset allocation*, Springer Science & Business Media, 2009.

[29] David M Pooley, Peter A Forsyth, and Ken R Vetzal, *Numerical convergence properties of option pricing pdes with uncertain volatility*, IMA Journal of Numerical Analysis **23** (2003), no. 2, 241–267.

[30] Paul Wilmott, *Paul wilmott introduces quantitative finance*, John Wiley & Sons, 2007.

[31] Tong Zhang, Bin Yu, et al., *Boosting with early stopping: Convergence and consistency*, The Annals of Statistics **33** (2005), no. 4, 1538–1579.

[32] Robert Zvan, Kenneth R Vetzal, and Peter A Forsyth, *Pde methods for pricing barrier options*, Journal of Economic Dynamics and Control **24** (2000), no. 11-12, 1563–1590.

# Appendices

# .1 Fokker-Planck's Equation

Let define a probabilistic space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\mathcal{F}_t$ is a filtration for the process $X_t$ and let $X_t$ be a stochastic process defined by the following SDE:

$$dX_t = f(X_t, t)dt + \sigma(X_t, t)dW_t, \tag{1}$$

where $W_t$ is a standard brownian motion under $\mathbb{P}$. Them, using Ito's formula, we have that for any suitable function $v(X_t, t)$ that:

$$\begin{aligned} v(X_T, T) - v(X_t, t) = &\int_t^T \left( \frac{\partial v}{\partial s}(X_s, s) + f(X_s, s)\frac{\partial v}{\partial X}(X_s, s) + \sigma^2(X_s, s)\frac{1}{2}\frac{\partial^2 v}{\partial X^2}(X_s, s) \right) ds \\ &+ \int_t^T \sigma(X_s, s)\frac{\partial v}{\partial X}(X_s, s)dW_s. \end{aligned} \tag{2}$$

Taking the expectation under $\mathbb{P}$ and using $v$ such that $v(x, t) = 0$ and $v(x, s) \to 0$ as $s \to T$ uniformly in $x$ we have that:

$$0 = \mathbb{E}\left[ \int_t^T \left( \frac{\partial v}{\partial s}(X_s, s) + f(X_s, s)\frac{\partial v}{\partial X}(X_s, s) + \frac{1}{2}\sigma^2(X_s, s)\frac{\partial^2 v}{\partial X^2}(X_s, s) \right) ds | \mathcal{F}_t \right]. \tag{3}$$

Therefore, writing in terms of transition density $\rho$ we have that:

$$0 = \int_t^T \int_{-\infty}^{\infty} \left( \frac{\partial v}{\partial s}(x, s) + f(x, s)\frac{\partial v}{\partial x}(x, s) + \frac{1}{2}\sigma^2(x, s)\frac{\partial^2 v}{\partial x^2}(x, s) \right) \rho dx ds. \tag{4}$$

Integrating by parts and choosing a suitable $v$ function with compact support, we have that:

$$\begin{aligned} \int_t^T \int_{-\infty}^{\infty} \frac{\partial v}{\partial s}(x, s)\rho dx ds &= \int_{-\infty}^{\infty} \left( v\rho \Big|_t^T - \int_t^T v(x, s)\frac{\partial \rho}{\partial s} ds \right) dx \\ &= \int_{-\infty}^{\infty} \int_t^T -v(x, s)\frac{\partial \rho}{\partial s} ds dx, \end{aligned} \tag{5}$$

$$\int_t^T \int_{-\infty}^{\infty} f(x,s)\frac{\partial v}{\partial x}(x,s)\rho dxds = \int_t^T \left( f(x,s)v\rho\Big|_{\infty}^{\infty} - \int_{-\infty}^{\infty} v(x,s)\frac{\partial f(x,s)\rho}{\partial x}dx \right) ds \quad (6)$$
$$= \int_t^T \int_{-\infty}^{\infty} -v(x,s)\frac{\partial (f\rho)}{\partial x}dxds,$$

$$\int_{-\infty}^{\infty}\int_t^T \frac{1}{2}\sigma^2(x,s)\frac{\partial^2 v}{\partial x^2}(x,s)\rho dxds = \int_t^T \left( \frac{1}{2}\sigma^2(x,s)\frac{\partial v}{\partial x}\rho\Big|_{-\infty}^{\infty} - \int_{\infty}^{\infty} \frac{1}{2}\frac{\partial v}{\partial x}(x,s)\frac{\partial(\sigma^2\rho)}{\partial x}dx \right) ds$$
$$= \int_t^T - \left( \frac{1}{2}v(x,s)\frac{\partial(\sigma^2\rho)}{\partial x}\Big|_{\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{1}{2}v(x,s)\frac{\partial^2(\sigma^2\rho)}{\partial x^2} \right) ds$$
$$= \int_t^T \int_{-\infty}^{\infty} \frac{1}{2}v(x,s)\frac{\partial^2(\sigma^2\rho)}{\partial x^2}dxds. \quad (7)$$

Substituting (5), (6) and (7) in (4) we have:

$$\int_t^T \int_{-\infty}^{\infty} -v(x,s)\frac{\partial \rho}{\partial s} - v(x,s)\frac{\partial(f\rho)}{\partial x} + \frac{1}{2}v(x,s)\frac{\partial^2(\sigma^2\rho)}{\partial x^2}dxds = 0, \quad (8)$$

for any suitable $v$. Therefore:

$$\frac{\partial \rho}{\partial s} = -\frac{\partial(f\rho)}{\partial x} + \frac{1}{2}\frac{\partial^2(\sigma^2\rho)}{\partial x^2}, \quad (9)$$

which is the Fokker-Planck's equation.