binaryOption

Dongxu Li

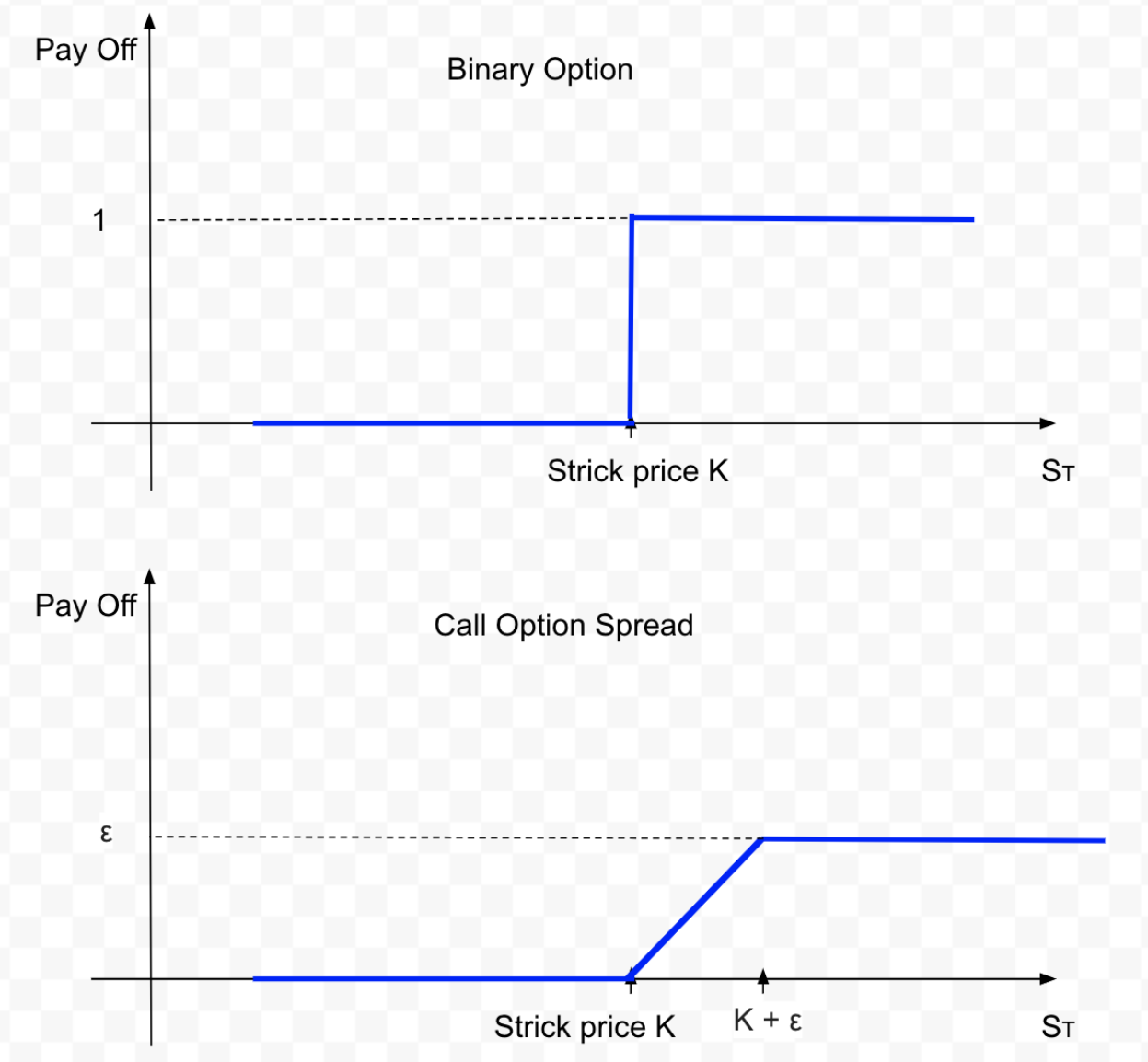
19/12/2018

Binary Option Pricing

Binary option is a financial exotic option in which the payoff is either some fixed monetary amount or nothing at all. The mathematics interpretation of a binary call would be: At time 0, we setup a call contract, the underlying asset price is S_0 and at the maturity date, the asset price is S_T. If $S_T > S_0$, we get 1 dollar in return, otherwise, we gets nothing.

First, from the classic black–Scholes model framework:

Let's look at the payoff of a binary call option and a call spread constructed by European Option:



It's easy to see that we could replicate the payoff of Binary Option by using $\frac{1}{\epsilon}$ amount of the spreads constructed by long a call with strick price K and short a call with strick $K+\epsilon$. And then we push ϵ to zero.

$$C_{Binary} = \lim_{\epsilon \to 0} \frac{C_{European}(K-\epsilon) - C_{European}(K)}{\epsilon}$$

Thus, the value of a binary call is the negative of the derivative of the price of a European call with respect to strike price:

$$C_{Binary} = -\frac{dC_{European}}{dK}$$

Plus it's very easy to get the Call price of the European Option by applying the classic black–Scholes formula.

Then the next step would be going back to the assumption of the black–Scholes formula that the underlying asset is a geometric brownian motion and can be expressed by stochastic differential equation:

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t$$

σ is the parameter we could get by using the sample standard deviation of our data.

Finally we could obtain the function of European Call as function of interest free rate r, volatility σ, strick price K, initial price of the asset S_0 and time to maturity T:

$$C_{European}(r, S_0, \sigma, K, T)$$

is explicitily determined by black-sholes formula.

In conclusion, it seems that the only thing we need to get from data is the volatility which is fairly easy to obtain.

Second, directly from a probability approch:

The replication of the binary option is fairly reasonable, but under the standard black–scholes model, the assumption is usually too strong. The model ignore the volatility skew. And it also assume the underlying asset is follow a specific diffusion model. Here I come up with a different idea to transform the original problem to a slightly different one. Our goal is to pricing a derivative with payoff at maturity, we care about the value: (since the time T is rather small in value, thus we ignore the risk free rate effect)

$$C_{Binary} = \mathrm{E}[\mathbf{1}_{\{S_T > S_0\}}] = \mathrm{P}(\{S_T > S_0\}) = \mathrm{P}(\{(\frac{S_T}{S_0} - 1) > 0\})$$

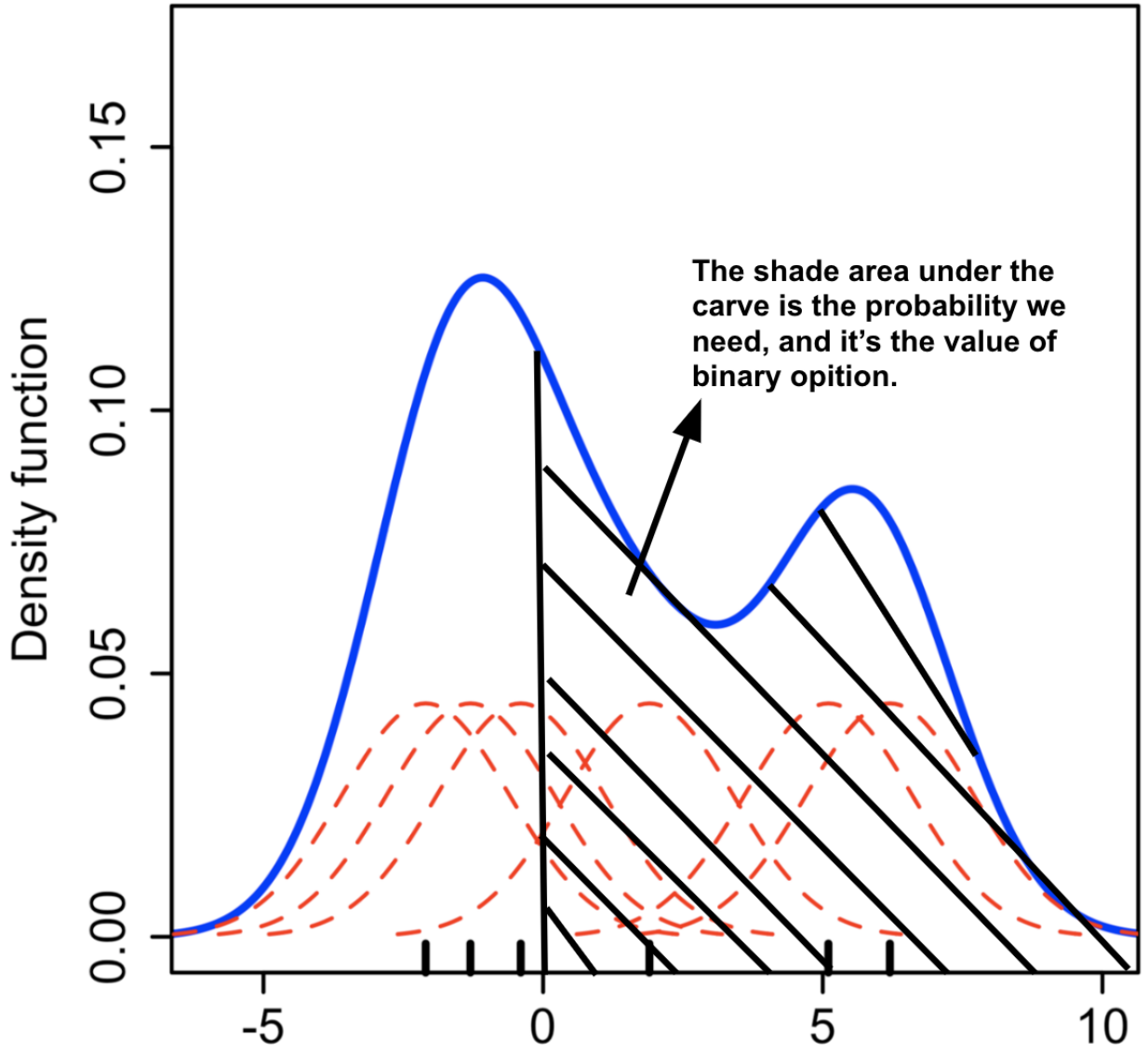
Then We could treat $\frac{S_T}{S_0} - 1$ as a random variable, and using the data we garthered to estimate the probability density function$f$(pdf) of this random variable. $\hat{f}$ is the function we are trying to estimate. The method we used here to eastimate the pdf is called Kernel density estimation. For more details explaination, please go to Kernel density estimation wiki:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big),$$

Here h is the called the bandwidth, which is treated as a hyperparameter. And function $K(.)$ here is called the Kernel function.

kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. Different Kernel functions could be used which makes this method very flexible. The assumption of KDE is that all sample values are from univariate independent and identically distributed sample. But we could go further to explore the Multivariate kernel density estimation.

After we have obtained the density funcion, we could integrate the function again the value larger than 0, and obtain the option price: This figure is the pdf of random variable $\frac{S_T}{S_0} - 1$.



Third, regression approach (or neural network approach).

Finally, since we want to estimate a probabiliy, I think maybe try a basic neural network approch with an output layer of a single neuron, and the neuron contain a sigmoid activation funtion. Basically, we are trying to find mapping from the historical price return to a probability output of future stock price direction. Think of our neural network as a mapping or function F, then the target is:

$$F(S_t, S_{t-1}, S_{t-2}..., S_{t-k}) \to \mathbf{1}_{\{S_T > S_t\}}$$

In the following, I did some rough numerical experiments, I collect about 30 days minute-by-minute USD/JPY price data from 2017.01.02 to 2017.01.31. The data is divided in to the training set and the test set.

Numerical Experiment for regression approach:

Here I used a 30 minutes time window to conduct the numerical experiment. The sample data is as below:

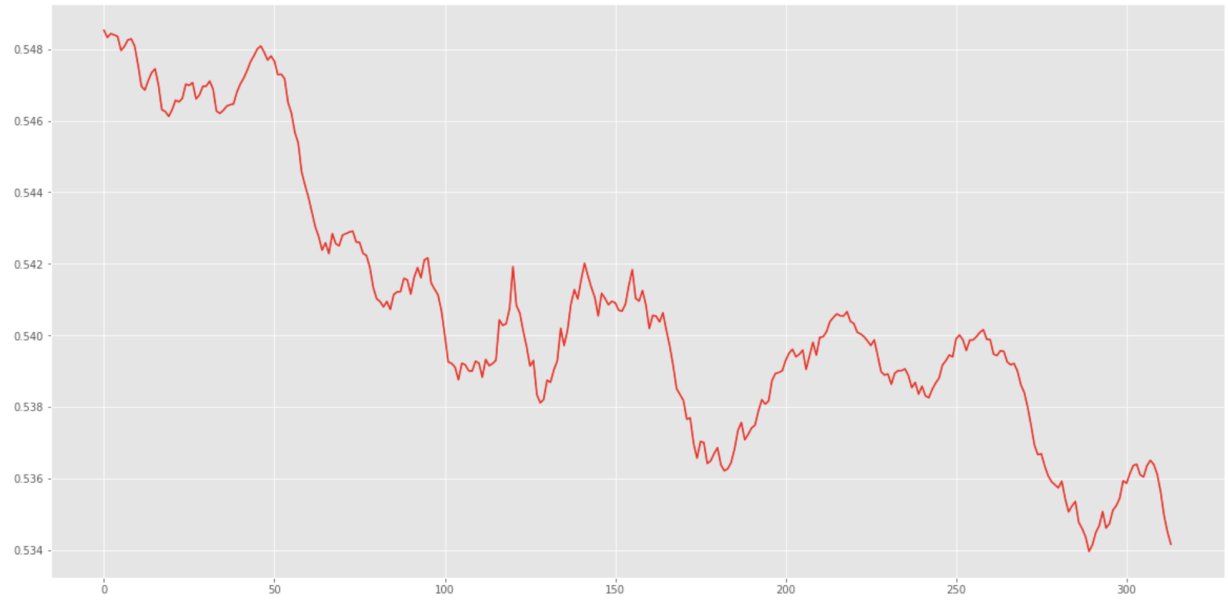
	Date	Time	Open	High	Low	Close
0	2017.01.02	02:00	116.858	116.870	116.858	116.870
1	2017.01.02	02:01	116.875	116.915	116.875	116.901
2	2017.01.02	02:02	116.901	116.901	116.901	116.901
3	2017.01.02	02:03	116.901	116.906	116.897	116.900
4	2017.01.02	02:04	116.900	116.903	116.900	116.903
5	2017.01.02	02:05	116.901	116.901	116.895	116.895
6	2017.01.02	02:06	116.893	116.897	116.893	116.897
7	2017.01.02	02:07	116.900	116.900	116.900	116.900
8	2017.01.02	02:10	116.894	116.900	116.894	116.900
9	2017.01.02	02:11	116.900	116.900	116.897	116.897

The table shape is (31501, 6), here we just used the close price. The function we are trying to estimate is:

$$F(S_t, S_{t-1}, S_{t-2}..., S_{t-30}) \to \mathbf{1}_{\{S_{30+t} > S_t\}}$$

Data processing part and tryout of regression are contained in the jupyter notebook.

There I define a neural network with 4 hidden layer, and each larger with specific number of neurons with specific activation functions. After the training, we plot our prediction below, value 1 indicates $S_{30+t} > S_t$, value 0 indicates $S_{30+t} < S_t$. The red line is our predicted probability of getting $S_{30+t} > S_t$.



The prediction is not perfect, the hyperparameter needs proper adjustment. This is more of a heuristic or very rough idea.

I took a brief look at the price in Binary.com, and find that the price is the fair price added by a little price premium. I assume the price different is due to the friction and may include the service fee.