Implied Volatility Surface Modeling

Kevin Xu (muyangx) Lynn Wang (yuwang3) Michael Wang (zizhangw)
Ming Yang (mingy)

Carnegie Mellon University, Tepper School of Business

May 4, 2019

Abstract

The implied volatility surface (IVS) is widely studied in the industry. In this project, we will apply various time-series and machine learning models in the forecast of IVS, which include linear regression, ARIMA, decision tree regression, random forest regression, linear GAM, VARMAX and support vector regression. We then explore the potential of creating a feature using the deviation from put-call parity. This model yields an MSE value slightly large than other models. Although the model is not impressive enough in terms of prediction accuracy, it is able to capture volatility spikes, which can be a really useful feature. Our model has demonstrated the potential of DPCP in modeling the volatility surface and recommended improvements for future practitioners.

1 Introduction

The Black-Scholes formula is famous for pricing options but the real-world uses Black-Scholes to map option prices to implied volatility (IV).

$$C(S_t, t) = N(d_1)S_t - N(d_2)PV(K) \tag{1}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right] \tag{2}$$

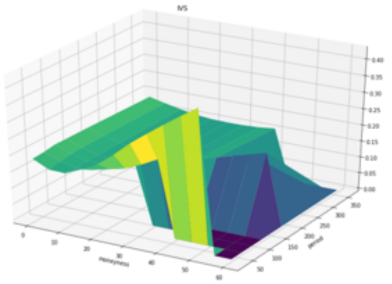
$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{3}$$

$$PV(K) = Ke^{-r(T-t)} \tag{4}$$

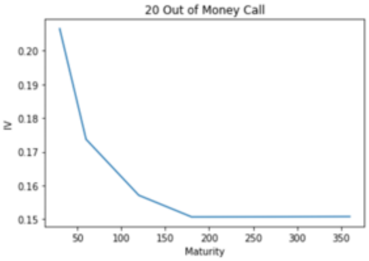
We can see that by inputting options price C, underlying price S, strike price K, and time to maturity T into Black-Scholes formula, we can solve for σ, the IV. The mapping of strike price (moneyness), time to maturity, and implied volatility is the implied volatility surface (IVS). It is shown in literature that the IVS contains rich information about future market volatility, and market makers, traders, and asset managers rely heavily on the IVS to calibrate pricing models.[1]

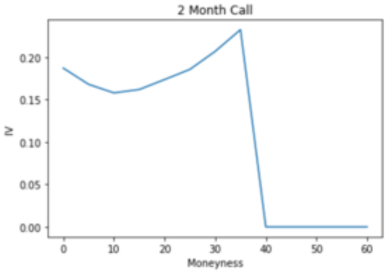
2 Data Set

Throughout this project we used 3 years of end of day SPX options data from CBOE (2016.1-2019.1).Key Entries in the data are: SPX (S), Underlying of the Options is the SP 500 Index. Strike Price(K), Time to Maturity (T), Moneyness: (K-S)/S for Calls and (S-K)/S for Puts, Implied Vol (IV): Solved through B-S with S, T, K, C as input



IVS





The above figures show the implied volatility surface of 2019-01-03. Looking at the 2d plots, we can see that the implied volatility decreases gradually from over 0.20 to 0.15 for 20 out of money calls. On the other hand, looking at the 2 month call we can see that for moneyness 0-40, there is a volatility skew between 0.15 and 0.2, and iv drops to zero as moneyness goes beyond 40. This means that investors do not think the SP 500 index will rise more than 40 percent in two months, and therefore, the value for deeply out of money calls are zero.

3 Methodology

Our prediction problem therefore becomes utilizing historical options data to predict future points on the IVS.

3.1 Multiple Linear Regression

We start with the simple model — Multiple Linear Regression. For our problem, the advantages of Multiple Linear Regression Model are as follows. 1) It is simple to interpret. 2) It is simple to tune and train. Most importantly, the assumption of a linear relationship between our response variable (atm iv) versus predictor variable (atm iv lagged and otm/itm iv) does make intuitive sense. With those in mind, let's take a look at the results.

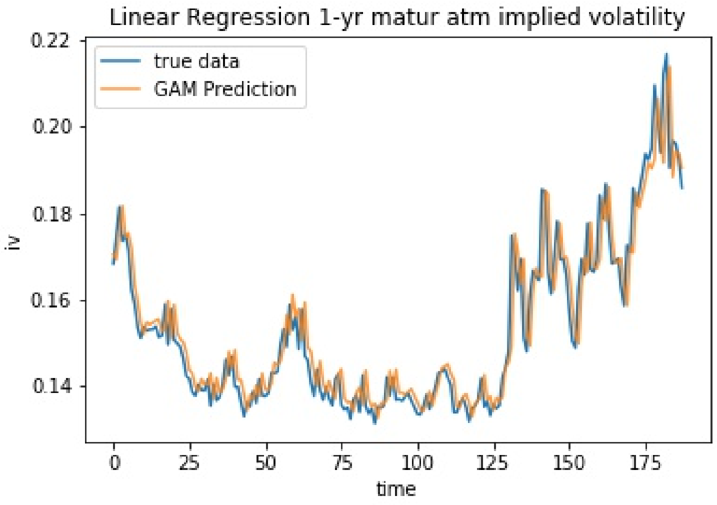


Figure 1: Linear Regression predicted vs. true 1-year maturity ATM call option IV

A decent performance for a rather simple model. Out of sample MSE for this model is 4.204e-05.

3.2 ARIMA

Uni-variate (single vector) ARIMA is a forecasting methods that predicts short term future values of a series using only the information of itself. This is our baseline model for the following reasons: 1) This is one of the common approaches for time-series data. 2) The model is essentially a regression model through OLS. 3) The predictive power of the model only originates from the single variable.

$$y_t = \mu + \psi_1 y_{t-1} + \psi_2 y_{t-2} + ... + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \epsilon_t \tag{5}$$

To calibrate the appropriate ARIMA model for Y, we first determine the order of differencing (i) so that we have a stationary time-series; i equals to one in our case. Since there is no obvious seasonality, we can move forward with the model. We calibrate the parameters in the model using the standard auto-arima function, which uses the lowest AIC and BIC.

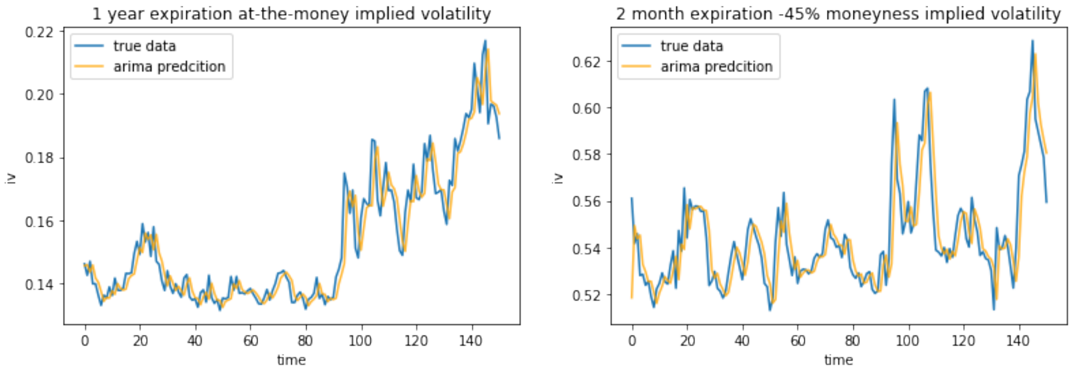


Figure 2: ARIMA prediction vs. true 1-yr atm and 2mn -45 moneyness IV

The model as expected does not have strong predictive power. We can see from the parameters that we use ARIMA-(0,1,1) for the plot above with θ approximately 1. The resulted MSE on average for different moneyness and maturity is 7.045e-05.

3.3 Decision Tree Regression

Decision Tree Regression or Regression Tree is a model that predicts the continuous value of a target variable based on several input variables using tree structures. The advantages of using a tree model include 1) it is simple to understand and interpret. 2) it performs relatively well with large datasets. 3) it has built-in feature selection. Additional irrelevant feature will be less used so that they can be removed on subsequent runs. This is particularly important as we are dealing with high-dimensional data. Based on all the reasons above, Regression Tree serves as a great starting point to explore all the possibilities for our purposes.

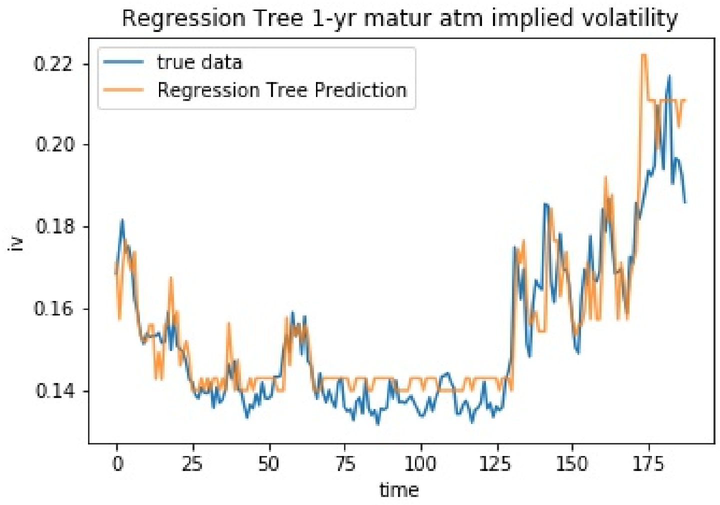


Figure 3: Regression Tree predicted vs. true 1-year maturity ATM call option IV

Since there isn't much gain in terms of MSE after depth goes beyond 10, we choose max depth 10 for our regression Tree. Using 75% of 755 days as training data and the rest being the test data, we obtain the following results. We can see it is not as impressive as the previous models and has room for improvement, but it is a good place to start nonetheless. Out of sample MSE for this model is 7.681e-05.

3.4 Random Forest

In this section, we explore Random Forests (RF). RF correct for decision trees' habit of overfitting to their training set. We have Feature Importances Plot to carefully select features. Unsurprisingly, the most useful predictors are the implied volatility itself (lag by 1) and the ones with almost identical maturity and very close moneyness.

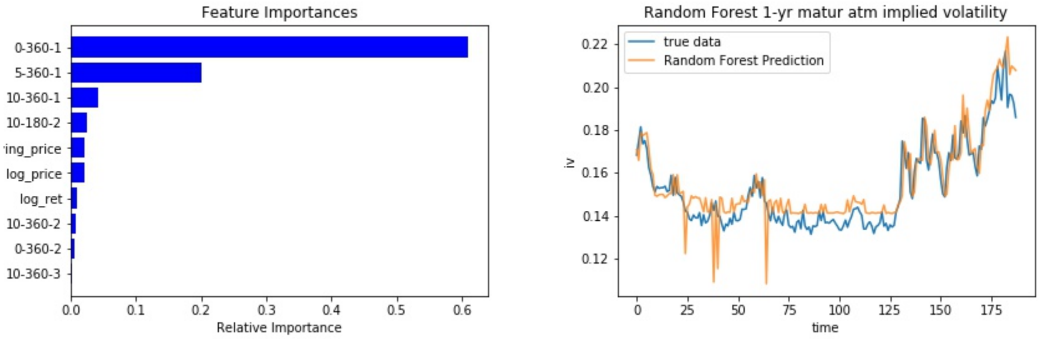


Figure 4: Random Forest predicted vs. true 1-year maturity ATM call option IV

For points half-way and beyond, the performance of random forest model is pretty good. However, for test points in the earlier to middle time period, we generally overestimate a bit with a few exceptions of drastically underestimating. Out of sample MSE for this model is 8.553e-05.

3.5 Linear GAM

A generalized additive model (GAM) is a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions. Therefore, it is a fairly reasonable model to check out for our purposes.

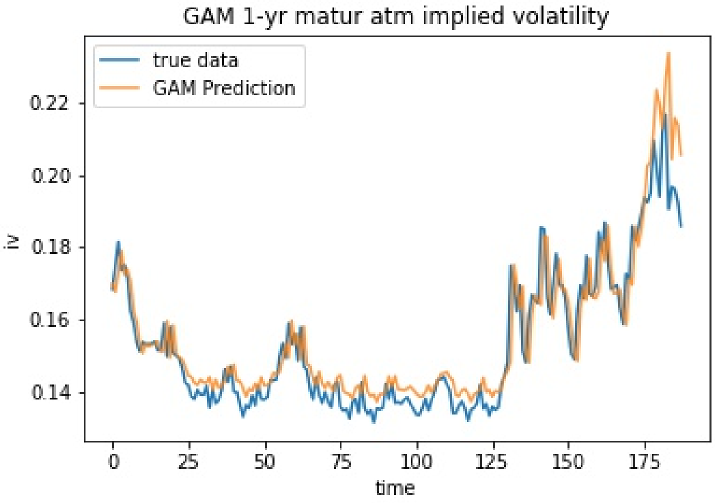


Figure 5: GAM predicted vs. true 1-year maturity ATM call option IV

GAM exhibits the similar prediction problem as our random forest model. However, regarding the test data, it has a better performance. Out of sample MSE for this model is 6.237e-05.

3.6 VARMAX

Vector Autoregressive Moving-Average with eXogenous regressors (VARMAX) is a multivariate version of the ARMA model that also includes the modeling of exogenous variables. This is an ideal method we would like to apply, because 1) it well captures past patterns in the time-series endogenous variables and 2) it takes into consideration the impact of contemporary exogenous variables. The VARMAX model is generically specified as:

$$y_t = \nu + A_1 y_{t-1} + ... + A_p y_{t-p} + B x_t + \epsilon_t + M_1 \epsilon_{t-1} + ... + M_q \epsilon_{t-q} \tag{6}$$

where y_t is a k_endog * 1 vector.

Our endogenous variable is a 40-dimension vector containing implied volatility of 40 options on the same day. The exogenous variable is time-t standardized log-return of the underlying asset (SPX).

Due to limitation of computation power, we were not able to select parameters using cross validation, so we chose (p,q) to be (1,1) according to results of baseline model ARIMA. We trained the following two VARMAX(1,1) models.

1) All 755 days data are used as training set. The in-sample MSE is 1.1563e-04. Prediction of implied volatility vs. true implied volatility of 1-year maturity at-the-money call option is shown above.

2) The first 500 days data (about 2/3 of all data) are used as training set. We make a two-day forecast and get an out-of-sample MSE of 5.3854e-05. Forecasting is limited in a short time window because the Python package VARMAX only supports making forecasts using previous predicted values, instead of taking in new data on a rolling basis for the time-series. Therefore, we reckon that forecasts in a long time scale should have a much larger error than the result if we could train and predict on a rolling basis.

The performance of this model is worse or barely better than our baseline model, we put up several possible explanations: 1) Choice of parameters (p,q) is limited to (1,1) here. If we could select parameter using cross validation, results will be improved. 2) If the model is trained and updated on a rolling basis, it will fit better to recent pattern and gives better predictions for the next day.

3.7 SVR

Support Vector Machine (SVM) is a supervised model used not only for classification, but also regression. For this particular project, we decided to explore the predictive power of Support Vector Regression (SVR)

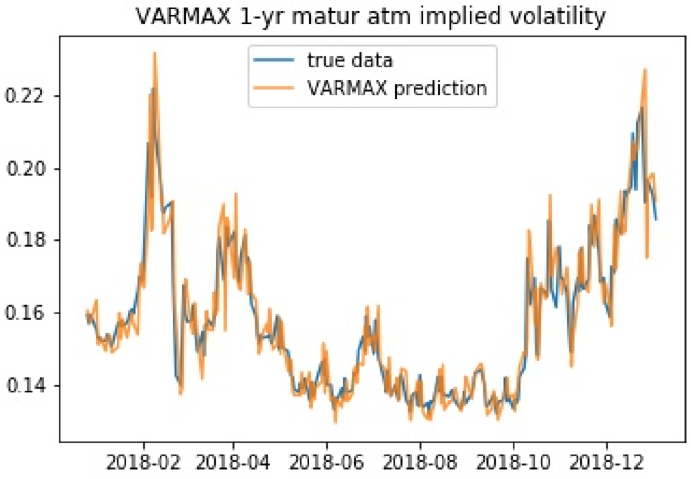


Figure 6: VARMAX predicted vs. true 1-year maturity ATM call option IV

for the following reasons. 1) SVR is considered a suitable model for high dimensional data. When we are measuring how the different iv from different periods affects the iv level today for online learning, we tend to have a significant amount of features resulting in high dimensional data. 2) The selection of kernels in the model allows us to capture complex relationship. 3) The model is also robust to outliers, which are quite common for financial data. 4) The model resistant to overfitting as well as co-linearity in our data, which is quite strong.

$$min\frac{1}{2}w^T * w \tag{7}$$

$$y_i - w_i^T * x_i - b \leq \epsilon + \xi_i \tag{8}$$

$$w_i^T * x_i + b - y_i \leq \epsilon + \xi_i^* \tag{9}$$

$$\xi_i, \xi_i^* \geq 0 \tag{10}$$

By time-series adjusted k-fold, we calibrate the model using a rolling window of 5 days. We use a linear kernel with $\epsilon = 0.0002$ and $C = 8.0$.

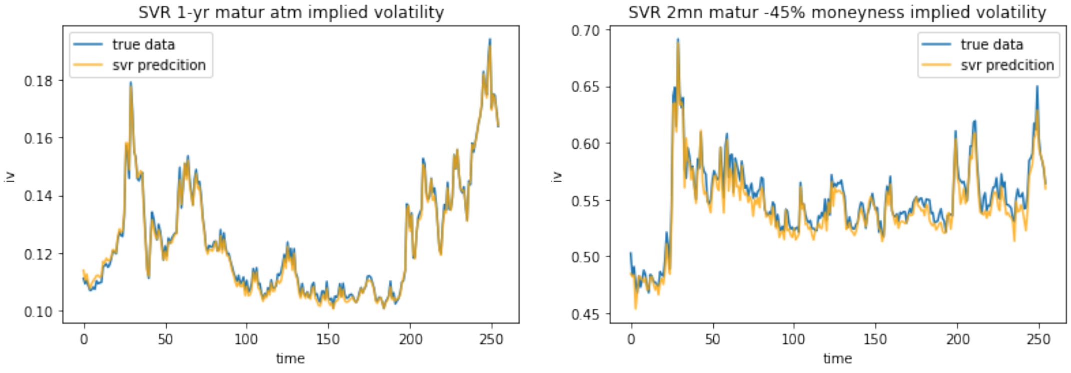


Figure 7: SVR prediction vs. true 1-yr atm and 2mn -45 moneyness IV

The model presents strong predictive power with impressive accuracy. It also addresses the issue most of our other models have by being able to predict spikes quite accurately. However, for some less traded moneyness and maturity, there tends to be noise in our data, which is a situation where SVR does not perform to well. The average out of sample MSE for this model is 9.203 e-06

4 Feature Exploration

Apart from the features in the original data set, we did some exploration into other features that can be fitted into models shown above. Here we introduce the feature we explored, DPCP, and the results obtained.

4.1 Theory

Relationship between prices of European put and call options with the identical strike price and expiry are described in put-call parity:

$$C_t - P_t = S_t - K * B(t,T) \tag{11}$$

We can reformulate the put-call parity as the following:

$$IV_t^C = IV_t^P \tag{12}$$

We define the deviation from put-call parity (DPCP) as:

$$DPCP_t = IV_t^P - IV_t^C \tag{13}$$

We can expect this has a mean-reversion pattern, and can be used to predict the change in implied volatility.

4.2 Model and Forecast

We first model the IV of contract j (both call and put) at time t as a function of moneyness and maturity, here the moneyness refers to delta-moneyness instead of moneyness in the original data set. Delta-moneyness equals to delta-0.5 for call options and delta+0.5 for put options:

$$IV_{j,t} = f(M_{j,t}, \tau_{j,t}|\beta_t) + \epsilon_{j,t} \tag{14}$$

We then have the DPCP of the ATM options:

$$DPCP_t = IV_{j,t}^P - IV_{j,t}^C = \epsilon_{j,t}^P - \epsilon_{j,t}^C \tag{15}$$

We expect an equilibrium-correction relationship between $\epsilon_{j,t}^P$ and $\epsilon_{j,t}^C$, which use DPCP as mean drift:

$$\Delta\epsilon_{j,t}^P = q_1 * DPCP_{j,t-1} + \eta_{j,t}^P \tag{16}$$

$$\Delta\epsilon_{j,t}^C = q_2 * DPCP_{j,t-1} + \eta_{j,t}^C \tag{17}$$

The proposed equilibrium-correction model for IVS is thus:

$$IV_{j,t}^P = f(M_{j,t}, \tau_{j,t}|\beta_t) + \epsilon_{j,t-1}^P + q_1 * DPCP_{j,t-1} + \eta_{j,t}^P \tag{18}$$

$$IV_{j,t}^C = f(M_{j,t}, \tau_{j,t}|\beta_t) + \epsilon_{j,t-1}^C + q_2 * DPCP_{j,t-1} + \eta_{j,t}^C \tag{19}$$

By fitting IVS to moneyness and maturity, we can get $\hat{\beta}_t$, and construct 1-day ahead forecasts by:

$$\hat{\beta}_{t+1} = \gamma + \sum_{k=0}^{p} \phi_k * \hat{\beta}_{t-k} + u_t \tag{20}$$

The forecast computation of equilibrium-correction model is:

$$IV_{j,t+1}^P = f(M_{j,t+1}, \tau_{j,t+1}|\hat{\beta}_{t+1}) + \epsilon_{j,t}^P + q_1 * DPCP_{j,t} \tag{21}$$

$$IV_{j,t+1}^C = f(M_{j,t+1}, \tau_{j,t+1}|\hat{\beta}_{t+1}) + \epsilon_{j,t}^C + q_2 * DPCP_{j,t} \tag{22}$$

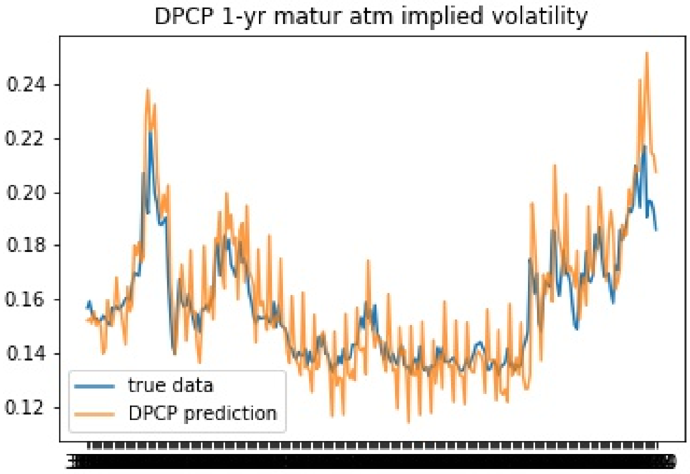


Figure 8: DPCP based predicted vs. true 1-year maturity ATM call option IV

4.3 Data, Analysis, and Results

We only have IV for at-the-money options that have same underlying, maturity, and strike price. Thus we only do predictions on the at-the money options' implied volatility.

We fit a linear regression to model each day (time t)'s IV as a function of delta-moneyness (M) and maturity(τ):

$$f(M_{j,t}, \tau_{j,t}|\beta_t) = \beta_{t,0} + \beta_{t,1}M_{j,t} + \beta_{t,2}\tau_{j,t} + \beta_{t,3}M_{j,t}^2 + \beta_{t,4}M_{j,t}\tau_{j,t} \tag{23}$$

Then we calculate the change in error terms in the above model, $\Delta\epsilon_{j,t}$ for put and call options separately. This model applies to all days, so we split the data set into training (500 days) and testing (255 days), and train a linear regression model to fit them as a function of DPCP one step before:

$$\Delta\epsilon_{j,t} = q * DPCP_{j,t-1} + \eta_{j,t} \tag{24}$$

We then construct one-day ahead forecast of $\hat{\beta}$ using VAR(5) model. We use AIC based lag order selection embedded in VAR package, limiting maximum lag to be 5, and get an optimal parameter to be 5. We made forecast on a rolling basis, i.e., we update the predictors as one day ends and $\hat{\beta}$ for that day is fitted.

$$\hat{\beta}_{t+1} = \gamma + \sum_{k=0}^{p} \phi_k * \hat{\beta}_{t-k} + u_t \tag{25}$$

Next, we apply the models on testing data set and construct forecasts according to this equilibrium-correction model, and get an MSE of 6.2488e-04. Prediction of implied volatility from this model vs. true implied volatility of 1-year maturity at-the-money call option is shown below. We can see from MSE and the figure that this model does not perform well, but it is just a prototype with a lot of improvements we can make in the future. However, one important feature is that it is able to capture the volatility spikes. When there are volatility spikes, we can get a good price and make more profit due to lack of liquidity in the market. Thus, capturing the volatility spikes is a desirable feature of a model.

Here are several further improvements we can make: 1) Apply more sophisticated models, rather than linear regression, to model IV and $\Delta\epsilon_{j,t}$ 2) Calibrate the time-series model applied to make more precise one-day forecast of $\hat{\beta}$ 3) Adding the DPCP feature as a predictor or exogenous variable to the models elaborated above to see if there is improvement in the performance.

5 Reference

[1]Cristian Homescu 2011 Implied volatility surface: construction methodologies and characteristics
[2]Francesco Audrino,Dominik 2009 Colangelo Semi-parametric forecasts of the implied volatility surface using regression trees
[3]Chuong Luong and Nikolai Dokuchaev 2018 Forecasting of Realised Volatility with the Random Forests Algorithm
[4]Yaxiong Zeng, Diego Klabjan 2018 Online Adaptive Machine Learning Based Algorithm for Implied Volatility Surface Modeling
[5]Xun Gong, Michel van der Wel, Dick van Dijk Forecasting the Implied Volatility Surface Using Put-Call Parity Deviations

6 Appendix

The source code of this project could be found at: `https://github.com/KeTIme/Vol_Modeling`. This repository contains joint work of all authors. Contact: Kevin Xu (muyangx@andrew.cmu.edu), Lynn Wang (yuwang3@andrew.cmu.edu), Michael Wang (zizhangw@andrew.cmu.edu), Ming Yang (mingy@andrew.cmu.edu)