A Machine Learning Method for High-Frequency Data Forecasting*

Erick López, Héctor Allende, and Héctor Allende-Cid

Department of Informatics,
Federico Santa María Technical University,
Avda. España 1680, Valparaíso, Chile
{elopez,hallende,vector}@inf.utfsm.cl
http://www.inf.utfsm.cl

Abstract. In recent years several models for financial high-frequency data have been proposed. One of the most known models for this type of applications is the ACM-ACD model. This model focuses on modelling the underlying joint distribution of both duration and price changes between consecutive transactions. However this model imposes distributional assumptions and its number of parameters increases rapidly (producing a complex and slow adjustment process). Therefore, we propose using two machine learning models, that will work sequentially, based on the ACM-ACD model. The results show a comparable performance, achieving a better performance in some cases. Also the proposal achieves a significatively more rapid convergence. The proposal is validated with a well-known financial data set.

Keywords: Financial High-Frequency Data, Time Series, ACM-ACD Model, Machine Learning, Forecasting.

1 Introduction

With recent technological advances, there has been an increasing demand for computational models, that are able to detect and quantify patterns, as well as, forecast data from high-frequency processes. An example is the financial market, where the price of an asset constantly changes in very short periods and is irregularly spaced over time, presenting a series of distinctive characteristics [1]. In recent years, a class of models has been proposed that considers these characteristics for modelling the durations and price changes between two consecutive transactions. These class of models are called ACM-ACD, and were proposed by Russell and Engle [3]. The ACM-ACD method, models the underlying joint distribution of durations and price changes in two parts: first it models the marginal distribution of the duration and then, it models the conditional distribution of price changes, given the duration. This approach proposes an autoregressive model for the durations and price changes, however, imposes

* This work was supported in part Research Grant Fondecyt (Chile) 110854 and MECESUP FSM 1101.

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 621–628, 2014.

distributional assumptions and is highly parametric. In [4], the authors show that the ACM-ACD model is sensitive to initial values of its parameters, is slow in terms of convergence and often fails (the parameter estimation process does not converge). In this work, based on the approach of the ACM-ACD model, we propose a model that uses Super Vector Machines and Artificial Neural Networks models, that work sequentilly. They follow the main idea of Russell's model. The advantage of using these types of models is that they work without distributional assumptions, have less parameters, and have faster convergence, thus directly affecting the response time, a very necessary characteristic, specially in Financial applications. The paper is structured as follows: In the next section we define the problem. In section 3 and 4, we briefly present the state of the art model ACM-ACD and the proposed model, respectively. In the following section we present the results with well-known real financial data. In the last section, we present some concluding remarks and delineate future work.

2 Problem Definition

From a statistical point of view, the ocurrence of an event (time) is a random variable from a stochastic point process. When the time comes with an additional feature (called mark), the double sequence is a realization from a marked point process.

Let t_i be the time at which the ith observation occurs and let $\tau_i = t_i - t_{i-1}$ be the duration between two consecutive observations. The marks observed at the ith event are denoted by $y_i \in \mathbb{R}^k$ which is a k-dimensional vector from a sample space Ξ. Therefore, the data can be viewed as $\{(\tau_i, y_i), i = 1, \ldots, T\}$, where the ith observation has an underlying joint conditional distribution on the past of (τ_i, y_i) given by:

$$(\tau_i, y_i)|\mathcal{A}_{i-1} \sim f(\tau_i, y_i|\breve{\tau}_{i-1}, \breve{y}_{i-1}; \theta_i) \tag{1}$$

where $\mathcal{A}_{i-1} = \{(\breve{\tau}_{i-1}, \breve{y}_{i-1})\}$ denotes the past of (τ_i, y_i) and θ_i are parameters that are potentially different from observation to observation. The underlying joint density (1) can be expressed as the product of the marginal density $q(\tau_i|\breve{\tau}_{i-1}, \breve{y}_{i-1}; \theta_\tau)$ (with parameters θ_τ) that explain the durations and the conditional density $g(y_i|\tau_i, \breve{\tau}_{i-1}, \breve{y}_{i-1}; \theta_y)$ (with parameters θ_y) for marks given both the duration and the past filtration of (τ, y).

Different statistical approaches have been proposed to address this issue, however most of them impose distributional assumptions and are overparameterized, producing a complex and slow adjustment process.

3 A Framework to Model the Durations and Marks

On the financial market context, τ is the duration between trades and y is the price change (the first difference: $y_i = p_i - p_{i-1}$), which is a discrete data.

3.1 A Statistical Model

One of the most important approaches for modeling the duration dynamics was introduced by Engle and Russell [2]. The main idea of the ACD model, that these authors proposed, is to model the durations $\{\tau_i\}_{t=1,\ldots,n}$ in terms of a multiplicative error model $\tau_i = \psi_i \varepsilon_i$ (see [1]), where ψ_i denotes a function of the past durations (and possible covariates), and ε_i defines an i.i.d. random variable of a standard exponential distribution with $E[\varepsilon_i] = 1$.

Later, Russell and Engle [3] proposed a model for the price change (from now on called marks), and modeled it as a multinomial random variable (because of its discrete nature) that could take k possible states, called autoregressive conditional multinomial of order (p, q, r) or simply ACM(p, q, r)

$$h(\pi_i) = c + \sum_{j=1}^{p} A_j (x_{i-j} - \pi_{i-j}) + \sum_{j=1}^{q} B_j h(\pi_{i-j}) + \sum_{j=1}^{r} \chi_j \log(\tau_{i-j+1}) \qquad (2)$$

where $h(\cdot)$ is the inverse logistic function, c is a (k-1) dimensional vector of constants, A_j and B_j denote the jth (k-1)×(k-1) parameter matrices, χ_j is a (k-1) parameter vector; p, q, r are unknown parameters. Besides x_i is a (k-1) vector indicating the discrete price change y_i, which takes the jth column of the (k-1)×(k-1) identity matrix if the jth state occurred; and π_i is a (k-1) vector of conditional probability (on information available at time t_{i-1}) associated with the states, which the jth element of π_i corresponds to the probability that the jth element of x_i takes the value 1.

Since the logarithm appears on the left hand side of equation 2, it seems natural to take the logarithm of the duration (look the right hand side), where the model depends on the log of the contemporaneous duration and on the first (r-1) lags of the log duration. Therefore, the dynamics for the durations are modelled by a variant of the ACD model: the log-ACD model, defined as

$$\log(\psi_i) = \omega + \sum_{j=1}^{u} \alpha_j \varepsilon_{i-j} + \sum_{j=1}^{v} \beta_j \log(\psi_{i-j}) + \sum_{j=1}^{w} \left(\rho_j y_{i-j} + \zeta_j y_{i-j}^2\right) \qquad (3)$$

where $\omega, \alpha_j, \beta_j, \rho_j, \zeta_j, u, v$ and w are scalar parameters.

Combining both expressions (2) and (3), the ACM-ACD model of order $(p, q, r) \times (u, v, w)$ is defined.

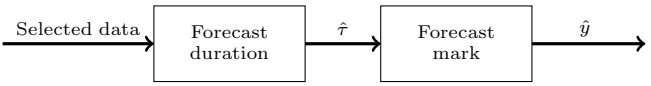
3.2 Machine Learning Models

Machine Learning is the area that is concerned of how to construct systems with the ability to automatically improve their performance in a given task using data about their behavior in the environment. Within the different paradigms of Machine Learning, the approach called "supervised learning" has as goal to learn a mapping $f : X \to Y$ from a set of input instances X, to a set of output instances Y.

In our problem, as the decomposition of joint density function induces a marginal model for the durations and a conditional model for the mark given the duration, where the durations are continuous and the marks are discrete, it is reasonable to try to model the duration with an Feedforward Artificial Neural Network model (FANN) of a single hidden layer [6] or a Support Vector Regression model (SVR) [5]; and with the obtained duration, model the mark with a Support Vector Machine model (SVM) [5], since the forecast of a new state is similar to classify a new mark given its history.

4 A Linear Ensemble Method to Model Jointly the Durations and Marks

We propose as a forecasting strategy, the combination of two machine learning models (based in the idea of ACM-ACD model [3]) that work sequentially. The first will forecast the durations and the second model, taking the duration forecasted in the previous step, plus its history $(\check{\tau}, \check{y})$, will forecast the mark.



Given the strategy presented above, we generate two forecasting models: νSVM-νSVR and νSVM-FANN, with the goal of comparing the forecasting performance of these two models with the ACM-ACD models, in order to compare the advantages and disadvantages of using machine learning models with high-frequency data.

Also, this proposal consideres this daily scenario: All price movements in one day are stored in a batch. Then (1) the model receives the batch at the end of the day; (2) the dataset is pre-processed and stored in a repository (database); (3) the model itself trains using the history contained in dataset and after that it begins to forecast; (4) waiting for the next batch, the model compares the forecast values versus true values (that came with the new batch) to measure its performance. The above four points are repeated every valid day (Monday to Friday).

The pre-processing steps consider: (a) a review on all points to detect incoherent durations, i.e., negative durations; (b) following Engle [2] the first half hour and last half hour of every day are deleted to reduce the effect of overnight information and market opening pressure, since the underlying process that data generating is different in those time intervals; (c) based on Brownlees [7] there exists the possibility of finding outliers in the data set, thereby, a filter is applied to delete the identified outliers.

The input that the machine learning models receive are vectors, which must be built selecting different points of the history (τ_i, y_i). To achieve this, a cross correlation function or the partial correlation function are used following the algorithm 1.

Algorithm 1. Lag Selection

1: $\{(\tau_i, y_i)\}$
2: $\{(\log(\tau_i), y_i)\} \leftarrow \{y_i^2\}$ # Add the squared mark and modify the duration by its logarithm.
3: $lag.max \leftarrow 0,9 \cdot \text{long}(batch)$ # Calculate the lag maximum.
4: $O^{acf} \leftarrow acf(\log(\tau), y^2)$ # Calculate the autocorrelation from 1 to $lag.max$ lag.
5: $\lambda \leftarrow Z_{0.975}/\sqrt{\text{long}(batch)}$ # Calculate the threshold for selecting the # significative lags.
6: $l^{up} \leftarrow O^{acf} > \lambda$
7: $l^{down} \leftarrow O^{acf} < -\lambda$
8: $index_1 \leftarrow [l^{up} \text{ or } l^{down}]$

Finally, to measure the performance, we will use different metrics with the purpose of comparing different factors of the forecasting process. Let N be number of forecasted points, then: (1) given that the duration forecasts are continuous values, the associated error be will calculated with the mean squared error $Error_\tau = \frac{1}{N}\sum_{i=1}^{N}(\log(\tau_i) - \log(\hat{\tau}_i))^2$, where τ_i is the i^{th} real duration and $\hat{\tau}_i$ its forecasted value; (2) the mark forecasts generated are discrete values, moreover nominal values within a small finite set, whereby, to measure the forecasting performance, we use the percent error $Error_y = \frac{1}{N}\sum_{i=1}^{N}(1 - \mathbb{1}(y_i, \hat{y}_i))$ where $\mathbb{1}(a, b) = 1$, if $a = b$, or 0 otherwise.

However, the percent error is not sufficient to explain if the model captured the underlying dynamics, since the high frequency data is characterized by high kurtosis, implying that if we forecast always the modal class we will have always a high performance. Therefore, we will use some metrics from multi-class classification problems for comparing its dynamics [9,8].

Let tp_i the number of correctly recognized examples that belong to the class i, tn_i the number of correctly recognized examples that do not belong to the class i, fp_i the number of examples that were incorrectly assigned to the class i, and fn_i the number of examples that were not recognized to belong to the class i, when really belonged. Given above, Table 1 shows the performance measures used in this work.

5 Experiments and Results

The data used in this work is the IBM stock data from January 1994, with 21 traded days and 34541 recorded movements. After pre-processing the data, the size decreased to 24818 points (ticks).

We will consider 4 scenarios: (1) the price increased or decreased one tick over time, (2) the price increased or decreased one tick or unchanged from their previous value, (3) the price up or down one tick, up or down two ticks over time, (4) the price up or down one tick, up or down two ticks, or unchanged from their previous over duration. We will analyze the 4 scenarios separately.

The νSVM-νSVR model was fitted by tuning the following parameters: $\nu \in [0, 1]$ for the SVM model, and $\nu \in [0, 1]$ and $C \in \{1, 2, \ldots, 10\}$ for the SVR model. In both cases we used a radial basis kernel, $k(a, b) = \exp(-\gamma \cdot \|a - b\|^2)$, where γ is 1/(input vector dimension).

Table 1. Performance measures used for measured the capture of underlying dynamic

$A_\mu = \frac{\sum_{i=1}^{k} tp_i}{tp_1 + fn_1 + fp_1 + tn_1}$,	Overall effectiveness of a classifier.
$A_M = \frac{1}{k}\sum_{i=1}^{k} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}$,	The average per-class effectiveness of a classifier.
$P_M = \frac{1}{k}\sum_{i=1}^{k} \frac{tp_i}{tp_i + fp_i}$,	The average per-class rate of actually occurred price changes when predicted.
$S_M = \frac{1}{k}\sum_{i=1}^{k} \frac{tp_i}{tp_i + fn_i}$,	The average per-class effectiveness of a classifier to indentify class labels.
$r_1 = \frac{\sum_{i=1\|y_i\neq 0}^{k} tp_i}{\sum_{i=1\|y_i\neq 0}^{k}(tp_i + fn_i)}$,	The fraction of correctly predicted class labels among the real price changes without considered when the price not change, i.e., class 0.
$r_2 = \frac{\sum_{i=1\|y_i\neq 0}^{k} tp_i}{\sum_{i=1\|y_i\neq 0}^{k}(tp_i + fp_i)}$,	The fraction of correctly predicted class labels among the correctly predicted price changes without considered when the price not change, i.e., class 0.

For the νSVM-FANN model, we tuned $\nu \in [0, 1]$ for the SVM model with the described above. The FANN model parameters were the number of neurons for a single hidden layer determined by the pyramid rule [10], $2\sqrt{N_i N_o}$, where N_i is the number of network inputs and N_o is the number of its outputs (in our case one). We used the resilient backpropagation algorithm with weight backtracking to train.

The ACM-ACD model was fitted using the Levenberg-Marquardt algorithm because of its convergence properties, however, the number of maximum iterations was only set to 50, because of its large time to converge. The tuning parameters used were the following: $p, q \in \{1, 2, 3\}$, $r \in \{1, 2, 3\}$, $u, v \in \{1, 2, 3\}$, $w \in \{1, 2, 3\}$. The method to estimate parameters was the maximum likelihood or minimum MSE.

In the proposed models (νSVM-νSVR and νSVM-FANN), the input vectors are formed by the cross autocorrelation function (ccf) or partial autocorrelation function (pacf) following the algorithm 1.

Let be k the number of states (indicating the ticks used, i.e., the different scenarios), Table 2 shows the best results obtained for each k value. The results show that the proposals based in machine learning achieve a better or comparable performance compared with the ACM-ACD model. We observe that in the three first scenarios the proposal outperformed the other models, however, with $k = 3$ the ACM-ACD model achieved a better performance in P_M, S_M, r_1, r_2. This scenario has a majority class, $y_i = p_i - p_{i-1} = 0$ (fact characteristic of financial high-frequency series), therefore there is a high kurtosis. In fact, when $k = 5$ our proposal is outperformed. In our case, these scenarios show a imbalanced multiclass classification problem and our proposal does not consider this. The multiclass task used was the "one-against-one" approach, which is not robust in the presence of unbalanced data and the ACM-ACD model was proposed

Table 2. Best Performance and Error for each scenario

Scenario $k = 2$	A_μ	A_M	P_M	S_M	r_1	r_2	$Error_\tau$	$Error_y$
νSVM(0.7)-νSVR(5,1)-ccf	0,5455	0,5455	0,5455	0,5455	0,5455	0,5455	6,0807	0,4545
νSVM(0.7)-FANN-ccf	0,5463	0,5463	0,5463	0,5463	0,5463	0,5463	7,1584	0,4537
Scenario $k = 3$	A_μ	A_M	P_M	S_M	r_1	r_2	$Error_\tau$	$Error_y$
νSVM(0.1)-νSVR(2,1)-ccf	0,5997	0,7332	0,3212	0,3316	0,0153	0,5022	10,8332	0,4003
ACM(3,3,3)-ACD(1,1,3)-lik-50	0,5924	0,7282	0,3533	0,3376	0,0371	0,5322	46,5466	0,4076
Scenario $k = 4$	A_μ	A_M	P_M	S_M	r_1	r_2	$Error_\tau$	$Error_y$
νSVM(0.01)-νSVR(1,0.4)-ccf	0,5395	0,7698	0,2698	0,2810	0,5395	0,5395	5,9835	0,4605
Scenario $k = 5$	A_μ	A_M	P_M	S_M	r_1	r_2	$Error_\tau$	$Error_y$
ACM(2,2,2)-ACD(2,2,2)-lik-50	0,5931	0,8373	0,2042	0,2012	0,0293	0,5199	18,7665	0,4069
ACM(2,2,3)-ACD(3,3,3)-error-50	0,6155	0,8462	0,1851	0,1999	0,0003	0,6667	11,7453	0,3845

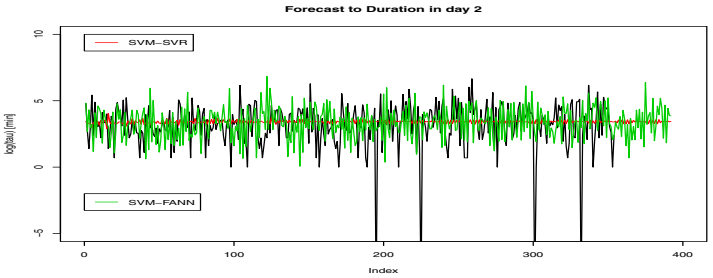


Fig. 1. Forecast of Logarithm of the Durations for day 2 using νSVM(0.7)-νSVR(5,1)-ccf (red line) and νSVM(0.7)-FANN-ccf (green line), when $k = 2$

considering this characteristics. So, in those scenarios the ACM-ACD model is expected to behave better.

On the other hand, when $k = 2$ the SVM-FANN model outperforms the SVM-SVR model in every performance measure. This scenario shows that a smaller error does not necessarely yield a better performance. Indeed, the details in the forecast process show that SVM-SVR is weaker capturing the dynamics compared with SVM-FANN (see figure 1).

Respect to the processing time performance measures, in all cases the proposal performs better that ACM-ACD model, although we used a fast algorithm as Levenberg-Marquardt to estimate parameters of the latter model.

6 Conclusions

This work presents a comparative evaluation between a Machine Learning framework versus a classical statistical approach (ACM-ACD model). We observed that our proposal achieved comparable results with the Russell's model, however, it is clear the potential of the latter model considering problems where there

is high kurtosis. As was expected, our proposal is significantly more quick, but was in some cases outperformed, in terms of capturing the underlying dynamics, showing the superiority of ACM-ACD in those scenarios.

Considering the other scenarios, our proposal achieved a better performance in all indicators, showing a better effectiveness, precision and sensitivity. It is necessary to highlight that the different results obtained with the SVM-SVR and SVM-FANN models in terms of $Error_\tau$ is not sufficient to detect and capture the durations dynamics. The SVM-SVR model achieved a better result in terms of the error, but the SVM-FANN model captured better its dynamics. Finally, despite that the ACM-ACD model used the algorithm Levenberg-Marquardt, our proposal achieved better running times, a very important issue in financial applications. Future work, will deal with the problem of the imbalancement in the data (high kurtosis).

References

1. Engle, R.F., Russell, J.R.: Analysis of High Frequency Financial Data. In: Ait-Shahlia, Y. (ed.) Handbook of Financial Econometrics (2004)
2. Engle, R.F., Russell, J.R.: Autoregressive conditional duration: A new model for irregularly spaced transaction data. Econometrica 66(5), 1127–1162 (1998)
3. Russell, J.R., Engle, R.F.: A discrete-state continuous-time model of financial transactions prices and times: The autoregressive conditional multinomial-autoregressive conditional duration model. Journal of Business and Economic Statistics 23, 166–180 (2005)
4. Zhang, Q., Cai, C.X., Keasey, K.: Forecasting using high-frequency data: a comparison of asymmetric financial duration models. Journal of Forecasting 28(5), 371–386 (2009)
5. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2001)
6. Hornik, K., Stinchcombe, M.B., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
7. Brownlees, C.T., Gallo, G.M.: Financial econometric analysis at ultra-high frequency: Data handling concerns. Computational Statistics and Data Analysis 51(4), 2232–2245 (2006)
8. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management 45(4), 427–437 (2009)
9. Zuccolotto, P.: Forecasting tick-by-tick price movements. Statistica & Applicazioni 2(1), 37–52 (2004)
10. Palit, A.K., Popovic, D.: Compuational Intelligence in Time Series Forecasting: Theory and Engineering Applications. Advances in Industrial Control. Springer (2005)