"Science Stays True Here"
Journal of Mathematics and Statistical Science, 149-167 | Science Signpost Publishing

Kernel Modification Effects for Support Vector Machine Applied to Limit Order Book of Nikkei 225 Futures

Hayato Kijima

Graduate School of Information Science, Toho University, Miyama 2-2-1, Chiba 274-8510, Funabashi, Japan

Hideyuki Takada

Department of Information Science, Toho University,Miyama 2-2-1, Chiba 274-8510, Funabashi, Japan

Abstract

Market participants place their limit/market orders by taking into account both the trajectory and current status of the limit order book. This behavior is based on the policy that the shape of the limit order book is quite informative for predicting future direction of a traded asset. In this paper, we employ Support Vector Machine to learn future mid-price directions and apply conformal transformation of the kernel function in order to improve its accuracy. Our empirical studies are based on Nikkei 225 futures and show that the conformal transform methods improved the precision more than 3% in average compared to the standard Gaussian RBF kernel. We further investigate numerically how the precision is improved by controlling parameter involved in the conformal transform.

Keywords: *Limit Order Book, Support Vector Machine, Riemannian metric,Conformal transformation*.

Introduction

Dynamics of the Limit Order Book (LOB - described below) has currently addressed with the growing availability of ultra high frequency data records. In fact, understanding the LOB dynamics would provide effective strategy to save the transaction costs for investors and also provide liquidity at efficient

---

Corresponding author: Hideyuki Takada, Department of Information Science, Toho university, Miyama 2-2-1, Funabashi, Chiba 274-8510, Japan. E-mail: hideyuki.takada@is.sci.toho-u.ac.jp.

price to market makers. Therefore it is important to investigate the stochastic dynamics of the LOB.

Market participants place limit/market orders taking into account both the trajectory and current status of the LOB. This behavior is based on the policy that the shape of the LOB is quite informative for predicting the future direction of a traded asset. In fact, many researchers tried to predict future price directions via machine learning techniques such as Logistic regression analysis in Ban Zheng [11] and Support Vector Machine as Kercheval and Zhang [6] and Deng, Sakurai and Shioda [3]. The other machine learning techniques such as Artificial Neural Network, Random forest and Naive-Bayes classifier and Support Vector Regression are applied to predict stock price for relatively long period (daily) without the information of the LOB as in Patel, Shah, Thakkar and Kotecha [7]. For the recent comprehensive treatmentfor LOB, from mathematical finance point of view,would be found in [5] Gould et al.

In this paper, we employ Support Vector Machine (SVM - described below) combined with conformally transformed Gaussian RBF kernel function to predict the mid-price dynamics in the LOB. For this purpose, as Fletcher and Shawe-Tayjor [4] studied, we first see the problem of the LOB dynamics so as to fit the SVM formulation. More precisely, shape of the LOB at each time is treated as a training data and its outcome that is observable in the future (direction of the mid-price movement) is treated as a corresponding label. We expect that if we can train the SVM with many training data effectively, then the SVM can predict future direction of the mid-price successfully. However, future direction of the mid-price would also depend on the balance between supply and demand in each moment, thereby even perfectly same shape of the LOB may produce opposite outcome. Because of such an inconsistency, one may think that high accuracy would not be expected for making predictions.

In order to improve accuracy of the SVM, we adopted kernel modification method pioneered by Amari and Wu [1]. They considered the feature space as a Riemannian manifold, which can be realized as a curved surface embedded in high dimensional Euclidian space, and express the distance of two distinct points in feature space via Riemannian metric. Since the Riemannian metric and the kernel function are related to each other, we can discuss about modification of the kernel functions within this framework. The issue is how to modify the kernel function preserving computational tractability. In order to focus on the effect of kernel modification, we don't discuss about sequential updating for control parameters of SVM.

As some existing researches such as Wu and Amari [10] and Williams, Li, Feng and Wu [8] report, in the realm of natural science, SVM with conformally transformed RBF kernel function exhibit more than 3% higher performance than standard RBF kernel. Main contribution of our study is to show the same aspect in the area of financial market possibly containing many kinds of noise. To the best of our knowledge, our empirical study is the first to apply the conformally transformed kernel to finance area, especially to the LOB.

In this paper, we first describe the data structure we used and methodology for applying SVM to the LOB. Next, theoretical backgrounds of the paper are descrived in detal and then explores the results for high frequency data of Nikkei 225 Futures.

Data and Methodology

Our empirical analyses are based on the historical data of the LOB of Nikkei 225 futures listed in Osaka exchange in Japan. Historical LOB data, we rely on, are provided by Osaka exchange via Rakuten Securities for her customer without any charge. This LOB data is comprised of agreed prices and event records such as volume of limit sell/buy orders for each price level with approximately 20 milliseconds time scale. Instead of these original data sets, we use adequately processed data such as second-scale historical data obtained by extracting every second from the original data. When there is no renewal in the LOB, our processed data remains the same. Example of the LOB data are shown in Table 1 illustrating the best-bid price at time $t =$ 09:02:42.379 is 1012 δ = 10120 yen and the best-ask price at the same time is 1013 δ = 10130 yen. From these records, we can read that the market sell order of size 233 was executed at $t =$ 09:02:47.913 and bid-ask spread appeared next 644 milliseconds. After that, no limit buy order at 1012 δ arrived and limit sell order of size 422 at the price 1012 δ newly reached hence the mid-price changed.

time	midprice	1008δ	1009δ	1010δ	1011δ	1012δ	1013δ	1014δ	1015δ	1016δ
09:02:42.379	10125	···	···	798	487	237	806	1041	843	···
09:02:43.923	10125	···	···	798	488	233	807	1041	843	···
09:02:47.913	10120	···	···	798	488	0	807	1041	843	···
09:02:48.010	10120	···	830	798	355	0	807	1041	843	···
09:02:48:557	10115	···	830	798	355	422	852	1041	···	···
09:02:48:877	10115	···	830	798	355	428	853	1051	···	···

Table 1: Example of the LOB data flow of Nikkei 225 futures as of Dec. 25, 2012. (δ = 10 yen)

Trading time of Nikkei 225 futures is divided into following four sessions; pre-opening session, regular session, pre-closing session and night session. During the pre-opening session starting from 8 AM and ending at 9 AM, and during the pre-closing session from 3:00 PM to 3:15 PM, limit orders are received but matching cannot be executed. During the night session from 4:30 PM to 3 AM, Japanese investors do not trade much and then very few transactions are made. Therefore, for our empirical studies, we use the data of regular session starting every weekday from 9 AM and ending 3:10 PM. Our empirical studies are concentrated to the period between April 2, 2012 and June 30, 2012.

State of the Limit Order Book (LOB)

A single snapshot of the LOB can be handled as a high dimensional vector recorded with timestamp. For more precise description of the LOB, we introduce coordinate system consist of time-axis, price-axis and volume-axis as shown in Figure 1. Here, for each time, we slide the price-axis and take the best-bid price as an origin of the price-axis in this coordinate system. We pay attention to the distance between the requested price of the limit order and mid-price rather than exact price level of the limit order. Thus each coordinate has three components such as time, the distance between the limit order and best-bid price, and the volume size of the limit order. It is assumed that limit orders and market orders can be placed on a fixed price grid $\{1,2,\cdots,N\}$ representing multiples of a price tick denoted by δ, so the state of the LOB can be seen as a discrete function on a discrete line calibrated with unit length of δ.

The upper boundary N is chosen large enough so that it is highly unlikely that orders at prices higher than N will be placed within the time frame of our analysis[1]. State of the LOB at time t is described by the discrete time N-dimensional stochastic process $\boldsymbol{Z}(t) = (Z_1(t),\cdots,Z_N(t))$, where the p-th element $Z_p(t)$ denotes the time t order size waiting for the future market order of price $p\cdot\delta$ to be matched. Practically in Nikkei 225 futures market, investors cannot necessarily observe all the limit orders although more exists. Because of the narrow window of records, only the information close to the best price (generally 10 prices for both sell and buy orders) are available and limit orders placed far from the best prices are not shown for investors.

Let the best-ask price at time t is denoted by $P_{sell}(t) > 0$ and similarly the best-bid price is denoted by $P_{buy}(t) > 0$. We define the number of outstanding sell orders at a distance k (equivalently $k\cdot\delta$ in price) from the best-bid price $P_{buy}(t)$ as $Q_{sell}^{k\delta}(t) \in Z_+$. Thus the quantity $Q_{sell}^{\delta}(t)$ indicates the number of orders of best-ask at time t. Similarly the number of outstanding buy orders at a distance k from the best-ask price is defined as $Q_{buy}^{k\delta}(t) \in Z_+$.

[1]As described in Cont, Stoikov and Talreja [10] since the model is intended to be used on the time scale of days, this finite boundary assumption is reasonable.

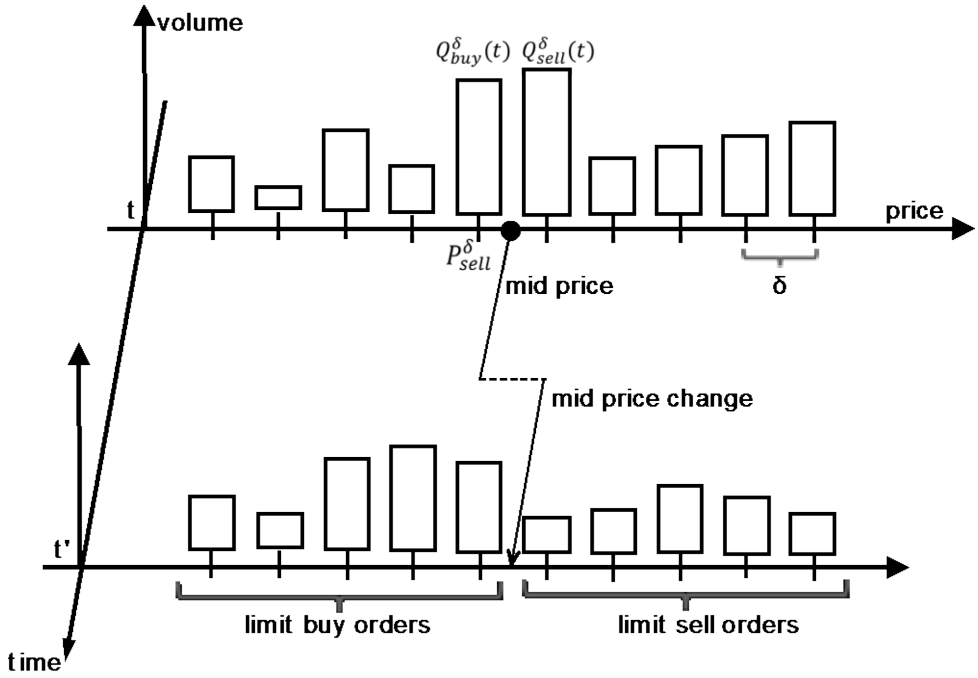

Figure 1: Time evolution of the Limit Order Book

Application of the Support Vector Machine

In this paper, we attempt to classify super-short-term (seconds) future price direction with SVM and focus on the improvement of its accuracy. We expect that the shape of the LOB is predictably effective for the direction of the next mid-price movements. As discussed above, the shape of the LOB means that the sequence of volume size of limit orders while absolute price levels are not contained. For simple explanation of our methodology, let us first consider the best-ask price and the best-bid price. When the volume of the best-ask price is relatively larger than that of the best-bid price, it is natural to consider that the market has a high likelihood of downtrend. This simple intuition would be applicable to the general circumstances considering all available limit orders. In fact, Fletcher and Shawe-Tayjor [5] considered not only the snapshot of the LOB, but also the time-derivatives (these are finite difference because of discrete time setting) of the volume size at each price level, which contains past information. In our study, we focus on the effectiveness of modification of the kernel function thus the simple structure of training data is preferred. Of course, our model could be applied to more generalized structure of training data including exponential moving average of the price, standard deviation of the price, the maximum and minimum prices over some period and the number of price increases and decreases over that time period as Fletcher and Shawe-Tayjor [5] suggested.

We consider discretized time grids $T_i, 0 \leq i \leq M$ to express full available historical data records of the LOB including the mid-price. Let the available shape of the LOB, i.e., the order book volume at time T_i is identified with $2n$-dimensional vector

$$\boldsymbol{x_i} = \left(Q_{buy}^{n\delta}(T_i), Q_{buy}^{(n-1)\delta}(T_i), \dots, Q_{buy}^{\delta}(T_i), Q_{sell}^{\delta}(T_i), \dots, Q_{sell}^{n\delta}(T_i)\right).$$

In case of Nikkei 225 futures, constantly available data is restricted to$1 \leq n \leq 9$, which may sounds odd. In many cases we can see limit sell/buy orders placed in the price corresponding to $n = 10$, however, sometimes at the time of mid-price change, the bid-ask spread, which is strictly larger thanδ, appears in a very short time and then the limit order corresponding to $n = 10$ is pushed out of the records. This is the reluctant reason why we have no other alternative but to handle the data with $1 \leq n \leq 9$.

Let $\pi(T_i) \coloneqq (P_{sell}(T_i) + P_{buy}(T_i))/2$ denote the mid-price at time T_i and define

$$\mathfrak{T}(T_k) = \min\{j + 1 \,|\, \pi(T_{j+1}) \neq \pi(T_j), j \geq k\}$$

as an index of discretized time grid at which the mid-price moved for the first time after T_k. Training data for SVM is a set of sequence of prior trials $\{\boldsymbol{x_i} \in R^{2n}\}_{0 \leq i \leq m}$ with $m \leq M$ that have already been classified into two classes. In our case, classification means that each $\boldsymbol{x_i}$ has been assigned a label $y_i \in \{+1, -1\}$ depending on the direction of the first mid-price movement as follows.

$$y_i = \begin{cases} +1 & if \;\; \pi\left(T_{\mathfrak{T}(T_i)}\right) > \pi\left(T_{\mathfrak{T}(T_i)-1}\right) \\ -1 & if \;\; \pi\left(T_{\mathfrak{T}(T_i)}\right) < \pi\left(T_{\mathfrak{T}(T_i)-1}\right) \end{cases}$$

That is, if the first mid-price movement after T_i was upward direction, then we set $y_i = +1$ and in an opposite case we set $y_i = -1$. Nonlinear SVM maps the input data $\boldsymbol{x} \in I = R^{2n}$ into a higher dimensional feature space $F = R^N, 2n \leq N$ by a nonlinear mapping Φ By choosing an adequate mappingΦ, the data points become mostly linearly separable in the feature space. Trained SVM finds maximum margin hyper plane in a feature space F as a final decision boundary defined by

$$f(\boldsymbol{x}) = \sum_{i \in SV} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b$$

where K is called kernel function and summation runs over all the support vectors. Here, parameters α_i are derived by solving quadratic programing problem

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y_i y_j \, K(\boldsymbol{x_i}, \boldsymbol{x_j})$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \qquad \sum_{i=1}^{m} \alpha_i y_i = 0$$

withprespecified parameter C, which controls the trade-off between margin and misclassification error. It is well known that $\Phi(\boldsymbol{x}) \in F$ is not necessarily known to derive separating boundary and we only need to know the inner product of vectors $\Phi(\boldsymbol{x})$ in the feature space by virtue of $K(\boldsymbol{x}, \boldsymbol{x}') = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}')$. Thus the trained SVM with m data will return predicted direction $y_{m+1} = h(\boldsymbol{x}_{m+1}) \coloneqq sign(f(\boldsymbol{x}_{m+1})) \in \{+1, -1\}$ for new trial data $\boldsymbol{x}_{m+1}$.

Geometric reformulation of the Kernel

In order to improve the performance of SVM classifiers, Amari and Wu [6] and Wu and Amari [7] proposed conformal transformation of kernel functions based on the understanding that a good kernel should enlarge the separation between the two classes. Furthermore, Williams, Li, Feng and Wu [8] studied more robust method in the sense that the additional free parameter is only one and then computational algorithm is kept simple.

From geometrical point of view, the mapped data $\boldsymbol{z} = \Phi(\boldsymbol{x})$ generally lie on a $2n$ dimensional surface S in F. Hear we omitted the subscript i in $\boldsymbol{x_i}$ indicating the time recorded for simplicity. If we assume that Φ has all continuous derivatives, S can be seen as an embedded submanifold possessing Riemannian metric. The Riemannian metric enable one to measure the distance of two distinct points on S via line integral along the shortest curve (geodesic) connecting these two points. Here, the line element, denoted by ds, can be expressed as

$$(ds)^2 = \sum_{1 \le i,j \le 2n} g_{ij} \, dx^i dx^j$$

where g_{ij} is called Riemannian metric and superscript i in x^i denote the i-th element of vector $\boldsymbol{x}$.

Applying Φ, $2n$-dimensional vector $\boldsymbol{dx} = (dx^1, \cdots, dx^{2n})$ is mapped to

$$\boldsymbol{dz} = \Phi(\boldsymbol{x} + \boldsymbol{dx}) - \Phi(\boldsymbol{x}).$$

Then the line element ds is expressed in feature space as

$$(ds)^2 = \|ds\|^2 = \|\Phi(\boldsymbol{x} + \boldsymbol{dx}) - \Phi(\boldsymbol{x})\|^2.$$

From Taylor's theorem for $2n$-variate function $R^{2n} = I \ni \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}) = (\Phi^1(\boldsymbol{x}), \cdots, \Phi^N(\boldsymbol{x})) \in F$, the line element ds is re-expressed with the kernel function K as follows

$$(ds)^2 = \sum_{k=1}^{N} \left\{ \sum_{j=1}^{2n} \frac{\partial \Phi^k(\boldsymbol{x})}{\partial x^j} \right\}^2$$

$$= \sum_{k=1}^{N} \sum_{1 \le i,j \le 2n} \frac{\partial \Phi^k(\boldsymbol{x})}{\partial x^i} \frac{\partial \Phi^k(\boldsymbol{x})}{\partial x^j} dx^i dx^j$$

$$= \sum_{1 \le i,j \le 2n} \left(\frac{\partial^2 K(\boldsymbol{x}, \boldsymbol{x}')}{\partial x^i \partial x'^j} \right)_{\boldsymbol{x}=\boldsymbol{x}'} dx^i dx^j .$$

Therefore the Riemannian metric induced on S can be written as

$$g_{ij}(\boldsymbol{x}) = \left(\frac{\partial^2 K(\boldsymbol{x}, \boldsymbol{x}')}{\partial x^i \partial x'^j} \right)_{\boldsymbol{x}=\boldsymbol{x}'} .$$

It shows how a local area in I is magnified (by the factor $g_{ij}(\boldsymbol{x})$) in F under the mapping $\Phi(\boldsymbol{x})$.

Conformal Transformation

In general, inner product of vectors defines the length of vector, the angle between two vectors and the orthogonality. Angle and orthogonality are invariant under the multiplication of inner product by a positive number and we say that multiplied inner product is conformal to the original inner product. Similarly, since a metric is an inner product on tangent space of manifold, when some metric on manifold is multiplied by a positive function on manifold, we say that the multiplied metric is conformal to the original one.

A conformal transformation preserves both angles and the shapes of infinitesimally small figures, but not necessarily their size. The original idea of Amari and Wu [6] and Wu and Amari [7] of conformal transformation of the kernel function is to enlarge the magnification factor $g_{ij}(\boldsymbol{x})$ around the separating boundary but reduce it around other points far from the boundary by modifying the kernel K as

$$\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}') = D(\boldsymbol{x}) D(\boldsymbol{x}') K(\boldsymbol{x}, \boldsymbol{x}')$$

with a properly defined positive function $D(\boldsymbol{x})$. We have

$$\tilde{g}_{ij}(\boldsymbol{x}) = D(\boldsymbol{x})^2 g_{ij}(\boldsymbol{x}) + D_i'(\boldsymbol{x}) D_j'(\boldsymbol{x}) K(\boldsymbol{x}, \boldsymbol{x}) + 2 D_i'(\boldsymbol{x}) D(\boldsymbol{x}) K_i'(\boldsymbol{x}, \boldsymbol{x})$$

where $K_i^{'}(\boldsymbol{x},\boldsymbol{x}) = \frac{\partial K(\boldsymbol{x},\boldsymbol{x}^{'})}{\partial x_i}\Big|_{\boldsymbol{x}=\boldsymbol{x}^{'}}$ and $D_i^{'}(\boldsymbol{x}) = \frac{\partial D(\boldsymbol{x})}{\partial x_i}$. In order to increase the soft margin around the separating boundary in F, $D(\boldsymbol{x})$ should be chosen in a way such that $\tilde{g}_{ij}(\boldsymbol{x})$ has greater values around the separating boundary. Next we list the form of $D(\boldsymbol{x})$ proposed by many authors.

The form of $\boldsymbol{D}(\boldsymbol{x})$

Some existing proposed functions are listed bellow. Amari and Wu [6] considered

$$D(\boldsymbol{x}) = \sum_{k\in SV} C_k \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x_k}\|^2}{2\tau^2}\right)$$

whereτ is a free parameter and summation runs over all support vectors. Hear C_k should be chosen carefully depending on support vectors. Wu and Amari [7] considered

$$D(\boldsymbol{x}) = \sum_{k\in SV} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x_k}\|^2}{\tau_k^2}\right)$$

where τ_k is defined by

$$\tau_k^2 = \frac{1}{M}\sum_{m=1}^{M} \|\boldsymbol{x_m}-\boldsymbol{x_k}\|^2.$$

The summation runs over M support vectors $\{\boldsymbol{x_m}\}_{m=1,\cdots,M}$ that are nearest to $\boldsymbol{x_k}$. Subsequently, Wu and Chang [9] extended this idea to undesirable case where the training dataset is imbalanced, by applying adaptively tuned τ_k according to the spacial distribution of support vectors in feature space F. This is achieved by taking τ_k as

$$\tau_k^2 = \mathrm{AVG}_{i\in\{\|\Phi(\boldsymbol{x_i})-\Phi(\boldsymbol{x_k})\|^2<\ell,\ y_i\neq y_j\}}(\|\Phi(\boldsymbol{x_i})-\Phi(\boldsymbol{x_k})\|^2)$$

where the average comprises all the support vectors in $\Phi(\boldsymbol{x})$'s neighborhood within the radius of ℓ but having a different class label. Hear, ℓ is the average distance of the nearest and the farthest support vector from $\Phi(\boldsymbol{x_k})$.

In this paper, as many authors selected ([6] [7] [8] [9]), we adopt the Gaussian radial basis function

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \tag{12}$$

as a primary kernel and as Williams, Li, Feng and Wu [8], we assume that the conformal function $D(\boldsymbol{x})$ has the form of

$$D(\boldsymbol{x}) = e^{-\kappa f(\boldsymbol{x})^2} \tag{13}$$

where $f(\boldsymbol{x})$ is given by the decision boundary (2) and κ is positive constant. We call f as the first-pass solution of primary Gaussian RBF kernel. This takes its maximum of the separating region where $f(\boldsymbol{x}) = 0$, and decays to $e^{-\kappa}$ at the margins of the separating region where $f(\boldsymbol{x}) \pm 1$.

One can see particularly that the conformal transformation under (13) achieves the expected change of magnifications depending on the distance from the separating boundary as Williams, Li, Feng and Wu [8] suggested as follows. First, we remember that the volume form with respect to the metric g_{ij} is given by $\sqrt{\det(g(\boldsymbol{x}))}dx^1 \cdots dx^{2n}$, where $g(\boldsymbol{x})$ is the matrix whose (i,j)-element is $g_{ij}(\boldsymbol{x})$. Here we call the term $\sqrt{\det(g(\boldsymbol{x}))}$ as the *magnification factor*, which represent how a local area is magnified in F under the mapping Φ. When K is the Gaussian RBF kernel (12),

$$g_{ij}(\boldsymbol{x}) = \frac{1}{\sigma^2}\delta_{ij}$$

and then the magnification factor is

$$\sqrt{\det(g(x))} = \frac{1}{\sigma^n}.$$

Furthermore, transformed Riemannian metric $\tilde{g}_{ij}$ is given by

$$\tilde{g}_{ij}(x) = \frac{D(x)^2}{\sigma^2}\delta_{ij} + D_i(x)D_j(x)$$

and then the magnification factor is

$$\det(\tilde{g}(\boldsymbol{x})) = \frac{D(\boldsymbol{x})^{2n}}{\sigma^{2n}} + \frac{D(\boldsymbol{x})^{2n-2}}{\sigma^{2n-2}}\sum_{i=1}^{2n} D_i(\boldsymbol{x})^2.$$

By definition, $D_i(\boldsymbol{x}), i = 1,2,\cdots,2n$ are the components of $\nabla D(\boldsymbol{x}) = D(\boldsymbol{x})\nabla log D(\boldsymbol{x})$, one then obtain the ratio of the transformed to the original magnification factors as

$$\sqrt{\frac{\det(\tilde{g}(\boldsymbol{x}))}{\det(g(\boldsymbol{x}))}} = D(\boldsymbol{x})^{2n}\sqrt{1+\sigma^2\|\nabla logD(\boldsymbol{x})\|^2}.$$

Substituting (13), one obtain

$$\sqrt{\frac{\det(\tilde{g}(\boldsymbol{x}))}{\det(g(\boldsymbol{x}))}} = \exp(-n\kappa f(\boldsymbol{x})^2)\sqrt{1+4\kappa^2\sigma^2 f(\boldsymbol{x})^2\|\nabla f(\boldsymbol{x})\|^2}.$$

Qualitative meaning of this equation would be summarized as follows. Dominating term $\exp(-n\kappa f(x)^2)$ in the right handside achieves the greatest magnification when $f(x)=0$ and decreases rapidly as $f(x)$ moves away from 0. Thus we can enlarge the magnification factor around the separating boundary but reduce it around other points far from the boundary. The other term $\sqrt{1+4\kappa^2\sigma^2 f(x)^2\|\nabla f(x)\|^2}$ in the right hand side could be understood as follows. Along contours of constant $f(x)$, the magnification is greatest where the contours are closest. In this paper, we call this SVM equipped with conformally transformed kernel function (6) with (13) as *modified kernel SVM*.

Computational algorithm for modified kernel SVM

In order to train the modified kernel SVM, first we need to train the SVM equipped with the primary kernel (12) as usual and then reuse the results $f(x)$ as the first-pass solution to get the second-pass solution of the modified kernel SVM. In each supervised training stage, training data should be chosen large enough to obtain reliable decision boundary. On the one hand, old information of the mid-price and the LOB would be less effective for prediction rather than the most recent information. In our empirical study, the training of SVM is refreshed every day and then, for each day, we chose the time interval with running interval $[T_m, T_{m+k}], (m = 0,1,2,\cdots)$ for training data, where $T_0 =$ 09:00:00 (opening time) and T_k =10:00:00 are fixed. Since we had processed the original data into second-scale data, $T_{m+1} - T_m =$ 1second.

Here, we must note that the label y_ℓ, which will be apparent at time $t > T_\ell$, associated with training data x_ℓ should be known by the time T_m to take into account for training at T_m. Otherwise one knows the answer before it happens. Therefore, if we are in a time T_m, we can use the data during $[T_{m-k}, T_\ell]$, where $\ell = max\{i \in \{1,2,\cdots,M\} | T_{\mathfrak{T}(T_i)} \leq T_m\}$ for learning at T_m. In what follows, we abbreviate the term *training data* $[T_m, T_{m+k}]$ so as to identify the training data and its corresponding time interval. So if we say *training data* $[T_m, T_{m+k}]$, it indicates the available training data during the time interval

$[T_m, T_{m+k}]$. Computational algorithm of our newly proposed prediction model of the modified kernel SVM is summarized as follows.

Algorithm

Step 0. Set the control parameters C, σ, κ and the time parameter T_0 =09:00:00 and T_k =10:00:00. Training data is set to $[T_0, T_k]$ and $m = 0$.

Step 1. Train SVM with a primary RBF kernel K defined by (12).

Get α_i, b and the set of support vectors that determine the decision boundary (2).

Step 2. Given $f(x)$, train the modified kernel SVM defined by (6) and (13) for the same data.

Get the new decision boundary $\tilde{f}(x)$.

Step 3. Based on $\tilde{f}(x)$ and newly observed current state of the LOB denoted by x_{m+k+1},

Calculate the prediction $y_{m+k+1} = \tilde{h}(x_{m+k+1}) = sign(\tilde{f}(x_{m+k+1}))$ and store the result.

Step 4. Increment the time parameter by 1 second; $m \leftarrow m + 1$.

Set the training data as $[T_m, T_{m+k}]$.

Go to Step 1 until T_{m+k} reaches 15:10:00, the ending time of the regular session.

To proceed above algorithm in a real market, total computational time of Step 1 and 2 should be sufficiency smaller than one tick, i.e., 1 second. In our Matlab computational environment of iMac with 4GHz Intel Core i7, 32GB 1867MHz DDR3, total computational time of Step 1 and 2 is 0.17 second in average.

Next, we visually illustrate rough mechanism how the modified kernel SVM can manage the problem of which the standard SVM can't resolve. We consider the case $n = 1$ for buy and sell orders to plot everything in 2 dimensional input space. Figures 2 and 3 plot the 2 dimensional vector s$x_i = (P^{\delta}_{buy}(T_i), P^{\delta}_{sell}(T_i))$, which are used as training data (green * and red +), and also a newly classified one plot indicating price prediction made by trained SVM. Suppose that the current time is t =14:23:53 as of April 17th 2012 and we want to predict the next mid-price direction, which become visible at $t' > t$ as$y = +1$. Figures 2 shows that the prediction made by the standard SVM was $y = -1$ illustrated by light orange plus symbol (+), which is incorrect. Figures 3 shows that the prediction made by the modified kernel SVM with $\kappa = 0.3$ was $y = +1$ illustrated by blue asterisk (*), which is correct.

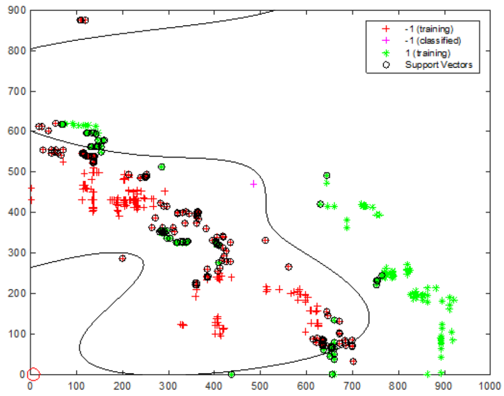

Figure 2: Prediction via the standard SVM

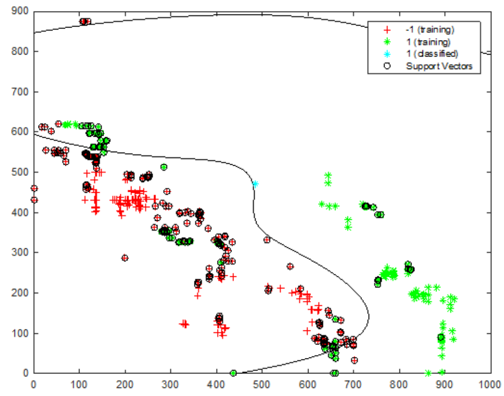

Figure 3: Prediction via the modified SVM

The above example illustrates the case that the trained data set is sparse around the trial data, which we want to classify. Contrary to the above illustrations, next Figures 4 and 5 shows the case that the trained data set is dense around the trial data.

Suppose that the current time is t =10:00:09 as of March 2nd 2012 and we want to predict the next mid-price direction, which become visible at $t' > t$ as $y = -1$. Figure 4 shows that the prediction made by the standard SVM was $y = +1$ illustrated by blue asterisk (*), which is incorrect. Figure 5 shows that the prediction made by the modified kernel SVM with $\kappa = 0.3$ was $y = -1$ illustrated by light orange plus symbol (+), which is correct. In both cases, we can see that via modification of the kernel function, misclassifications decided by the standard SVM is corrected

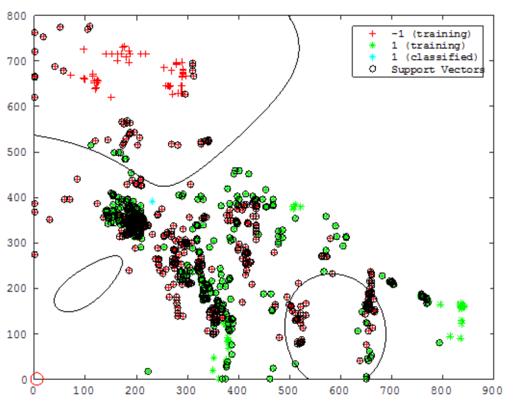

Figure 4: Prediction via the standard SVM

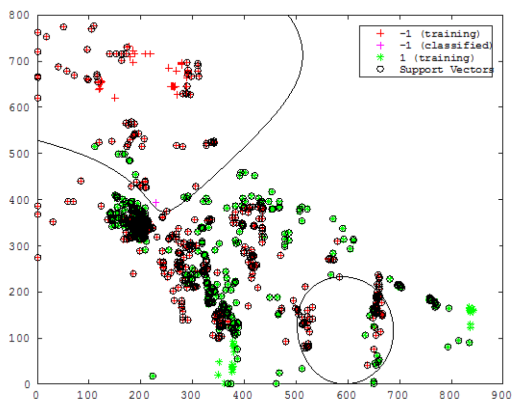

Figure 5: Prediction via the modified SVM

Thus the modified kernel seems to have great potential for improvement and next we illustrate the detail of our empirical studies. Since the trajectory of the LOB is an aggregated view of many kinds of market participants, pros and cons of future direction of the mid-price are batted back and forth, even exactly the same shape of the LOB have the potential of opposite direction. This would be a one of the reason why performance of SVM is relatively low rather than the applications to natural sciences. In our empirical studies in the area of finance, we show that the conformal modification possesses nice performance.

Results and Discussion

We need to determine evaluation method in order to compare the newly proposed prediction model of the modified kernel SVM with the standard one that is treated as a benchmark. Our objective is to make sure of the existence of nice parameter κ that controls $D(x)$ so as to outperform the benchmark. We rely on a confusion matrix for evaluation but we don't want to get into the detail for tuning methodology of the control parameters C and σ. Therefore we calculate the case with $C \in \{1, 10, 100\}$ and $\sigma = 1$ and take average of these three cases in terms of **Precision** defined bellow.

Let $\sigma_{1,1}$ be a number that count the event of actual upward direction of the mid-price movements that were correctly predicted as upward direction (true positive), while a number $\sigma_{-1,1}$ count the event of actual upward direction that were incorrectly predicted as down ward direction (false negative). Similarly, let $\sigma_{-1,-1}$ be a number that count the event of actual down ward direction of the mid-price movements that were correctly predicted as down ward direction (true negative), while a number $\sigma_{1,-1}$ count the event of actual down ward direction that were incorrectly predicted as up ward direction (false positive). These numbers constitute the confusion matrix as given in Table 2, which is often employed in the field of machine learning and specifically the problem of statistical classification.

Table 2: Confusion matrix

		Realization		
		1	-1	Total
Prediction	1	$\sigma_{1,1}$	$\sigma_{1,-1}$	$\sigma_{1,1} + \sigma_{1,-1}$
	-1	$\sigma_{-1,1}$	$\sigma_{-1,-1}$	$\sigma_{-1,1} + \sigma_{-1,-1}$
Total		$\sigma_{1,1} + \sigma_{-1,1}$	$\sigma_{1,-1} + \sigma_{-1,-1}$	$\sigma_{1,1} + \sigma_{-1,1} + \sigma_{1,-1} + \sigma_{-1,-1}$

And then the concepts of precision, recall and accuracy are defined respectively as follows.

$$\mathbf{Precision}_{(+1)} = \frac{\sigma_{1,1}}{\sigma_{1,1} + \sigma_{1,-1}}, \qquad \mathbf{Recall}_{(+1)} = \frac{\sigma_{1,1}}{\sigma_{1,1} + \sigma_{-1,1}}$$

$$\mathbf{Precision}_{(-1)} = \frac{\sigma_{-1,-1}}{\sigma_{-1,1} + \sigma_{-1,-1}}, \qquad \mathbf{Recall}_{(-1)} = \frac{\sigma_{-1,-1}}{\sigma_{-1,-1} + \sigma_{1,-1}}$$

$$\mathbf{Accuracy} = \frac{\sigma_{1,1} + \sigma_{-1,-1}}{\sigma_{1,1} + \sigma_{-1,1} + \sigma_{1,-1} + \sigma_{-1,-1}}$$

Roughly speaking, precision and recall are related as trade-off. One can extract some model parameters such as C and σ based on some tuning methodology. One of these methodology would be achieved by solving numerically the next optimization problem; maximizing **Precision** with respect to C and σ under some constraint on **Recall** such as **Recall**>0.6. However, as we simply focus on the effect of the conformal transformation, we do not involved with the detail of tuning methodology for C and σ. Therefore, in the rest of our study, we evaluate mainly in terms of **Precision**.

The effect of the conformal transforms

The most intrigued results of the empirical analysis would be the following; how much the modified kernel SVM outperforms the benchmark? Next figures (6) and (7) show the performance of the predictions in terms of $\mathbf{Precision}_{(+1)}$ and $\mathbf{Precision}_{(-1)}$ respectively for each days. For illustration, we selected $n = 9$ and then the training data for each time T_i is 18 dimensional vector $\boldsymbol{x}_i$. For the training of modified kernel, i.e., in order to get the second-pass solution, we fixed$\kappa = 1.5$. We calculated separately for three cases $(C, \sigma) = (1,1)$, $(10,1)$, $(100,1)$and take average of these cases in order to eliminate the possibilities of misleading that may have been caused by inappropriate specification of C andσ. Both two graphs show that the precisions obtained by the modified kernel SVM (solid line) are mostly located above the benchmark (dashed line). As Table 3 shows, $\mathbf{Precision}_{(+1)}$ and $\mathbf{Precision}_{(-1)}$ outperform the benchmark more than 3% in average between April 2, 2012 and June 30, 2012.

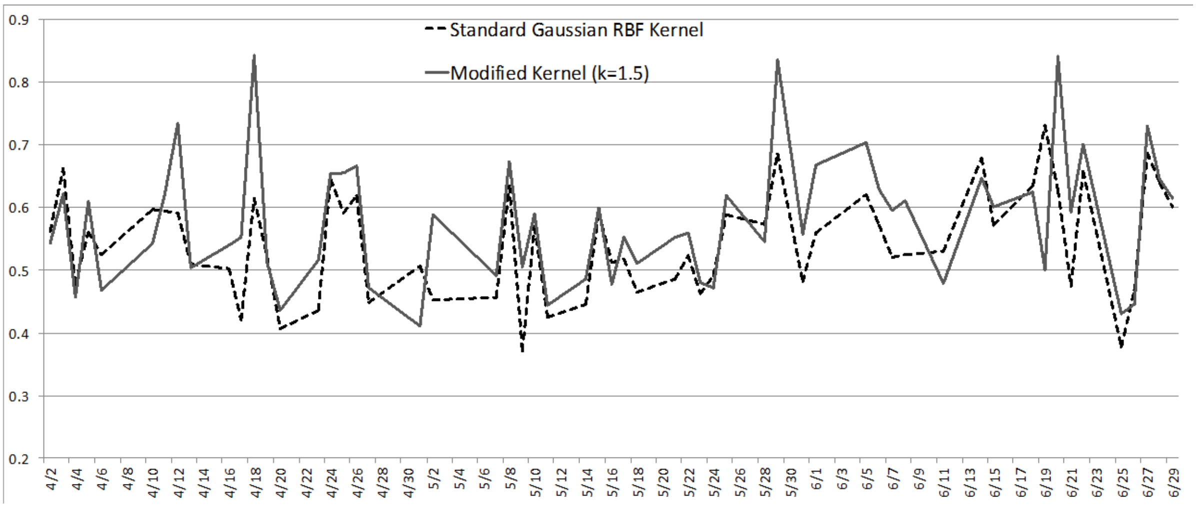

Figure 6: Transition of $\mathbf{Precision}_{(+1)}$ from April 2, 2012 and June 30, 2012

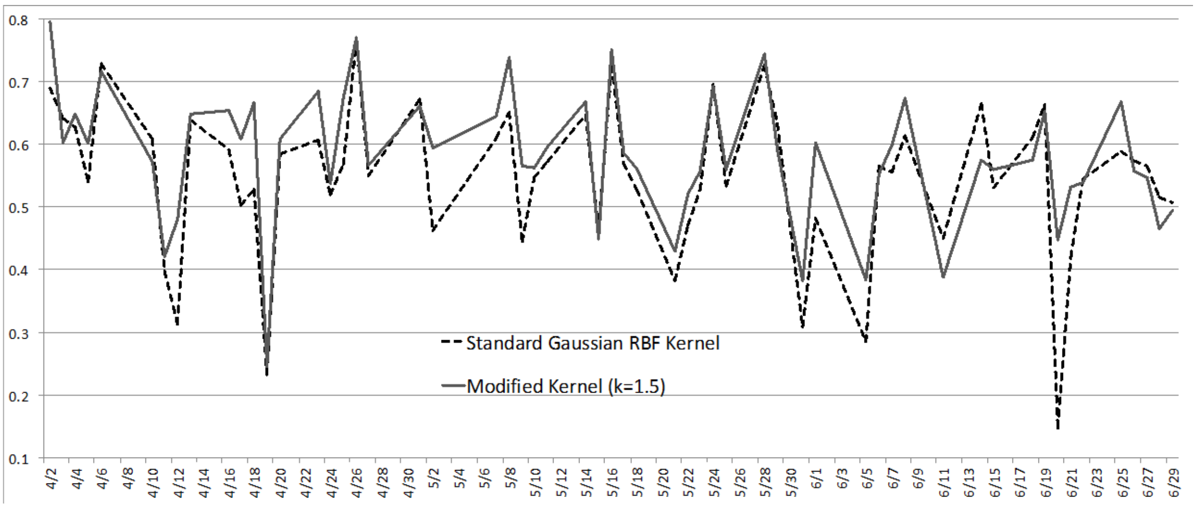

Figure 7: Transition of $\mathbf{Precision}_{(-1)}$ from April 2, 2012 and June 30, 2012

Table 3: $\mathbf{Precision}_{(+1)}$ and $\mathbf{Precision}_{(-1)}$ in average between April 2, 2012 and June 30, 2012

	Benchmark	Modified SVM		Benchmark	Modified SVM
$\mathbf{Precision}_{(+1)}$	54.340%	57.757%	$\mathbf{Precision}_{(-1)}$	54.566%	58.220%

The effect of κ

In the previous subsection, we compared the performance of modified kernel SVM with the benchmark under prespecified parameter κ in (13), which plays an important rule for improvement. Next figures from 8 to 13 show how the $\mathbf{Precision}_{(+1)}$ and $\mathbf{Precision}_{(-1)}$ vary with respect to κ for each month. As is clear from (13), $\kappa = 0$ corresponds to the standard SVM and then by viewing these figures we can see how much the precisions improved when we increased κ. Three lines black, orange and blue in each figure corresponds to the dimension n of training data.

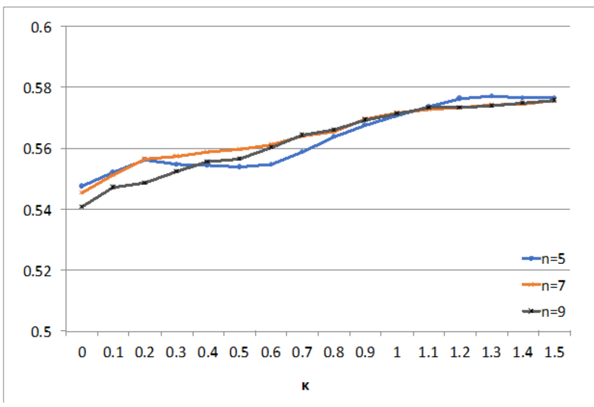

Figure 8: Average of $\mathbf{Precision}_{(+1)}$ in April 2012

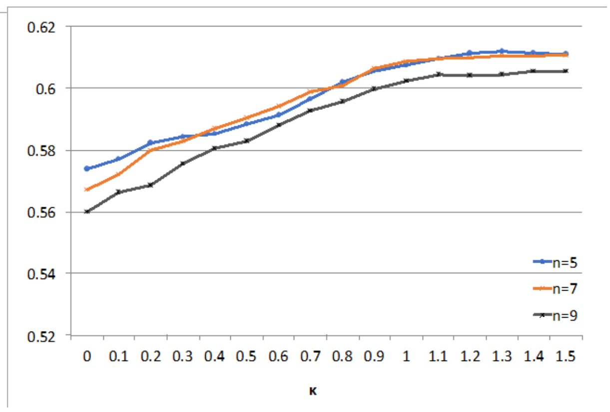

Figure 9: Average of $\mathbf{Precision}_{(-1)}$ in April 2012

Figure 8 shows that the $\mathbf{Precision}_{(+1)}$in April 2012 increase with the increasing κ for the three cases $n = 5,6,7$ in much the same way, while the Figure 9 shows, the level of the$\mathbf{Precision}_{(-1)}$in April 2012 vary depending on n, i.e., $n = 5,6$ exceed $n = 9$. Thus we can't conclude that $n = 9$is the most informative.

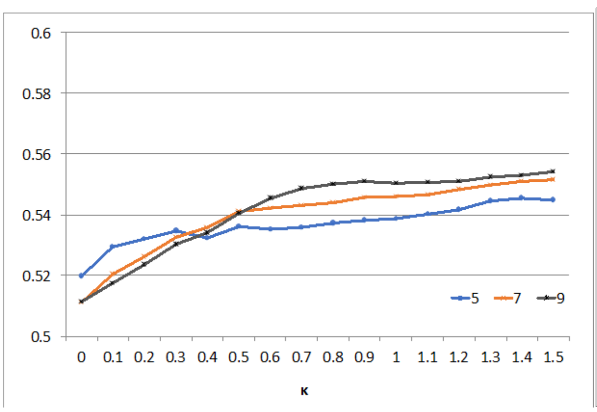

Figure 10: Average of $\mathbf{Precision}_{(+1)}$ in May 2012

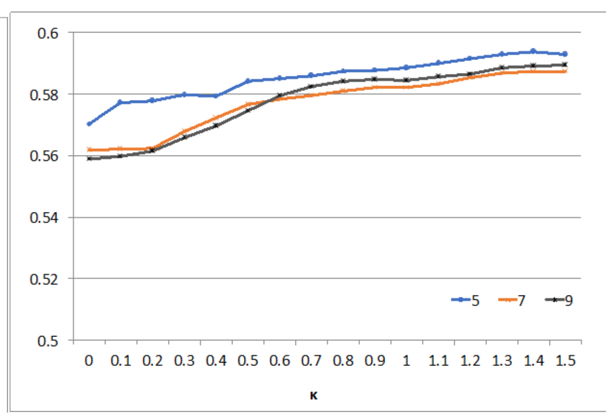

Figure 11: Average of $\mathbf{Precision}_{(-1)}$ in May 2012

Figure 10 shows that increasing κ is more effective to larger n, while Figure 11 shows that it is not necessarily correct. It is interesting to note that, as Figure 12 and 13 show, $n = 5$ case is less sensitive for increasing κ rather than $n = 7,9$ cases.

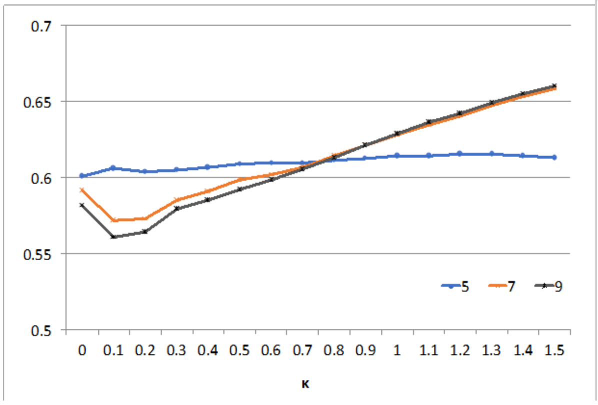

Figure 12: Average of $\mathbf{Precision}_{(+1)}$ in June 2012

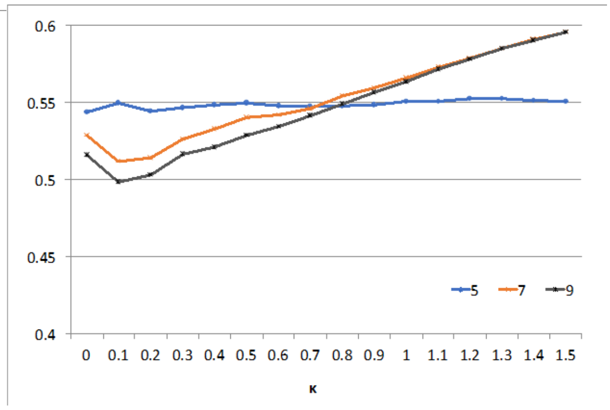

Figure 13: Average of $\mathbf{Precision}_{(-1)}$ in June 2012

Conclusions

As some existing works support, SVM would be a one of the best machine learning technique for predicting mid price direction by using LOB. In this paper, we found effectiveness of the SVM equipped with conformally transformed Gaussian RBF kernel to improve precision and studied the sensitivity with respect to the parameter κ which control the degree of the conformal transform. Since the computational load is twice as much as the standard Gaussian RBF kernel, conformal transform method would be widely applied to high frequency data analysis in finance.

Acknowledgment

The authers are grateful to participants of the 18^{th} SIG-Financial Informatics conference in Tokyo.

References

[1]. Shun-ichiAmari and Si Wu (1999) Improving support vector machine classifiers by modifying kernel functions, Neural Networks, 12, 783-789.

[2]. Rama Cont, Sasha Stoikov and Rishi Talreja (2010) A stochastic model for order book dynamics. Operations Research, Vol. 58, No. 3, May-June 2010, 549-563.

[3]. Shangkun Deng, Akito Sakurai and Kei Shioda (2011) Prediction of Foreign Exchange Market States with

Support Vector Machine, 10th International Conference on Machine Learning and Applications and Workshops, Vol. 1, 327-332.

[4]. Tristan Fletcher and John Shawe-Taylor (2013) Multiple Kernel Learning with Fisher Kernels for High Frequency Currency Prediction, Computational Economics, Vol. 42, Issue 2, 217-240.

[5]. Martin D. Gould, Mason A. Porter, Stacy Williams, Mark Macdonald, Daniel J. Fenn and Sam D. Howison (2013) Limit order books, Quantitative Finance, Vol. 13, No. 11, 1709-1742.

[6]. Alec N. Kercheval and Yuan Zhang (2015) Modellinghigh-frequency limit order book dynamics with support vector machines, Quantitative Finance, Vol. 15, Issue 8, 1315-1329.

[7]. Jigar Patel, Sahil Shah, PriyankThakkar and K Kotecha (2015) Predicting stock and stock price in- dex movement using Trend Deterministic Data Preparation and machine learning techniques, Expert Systems with Applications, Vol 42, 259-268.

[8]. Peter Williams, Sheng Li, Jianfeng Feng and Si Wu (2005) Scaling the kernel function to improve performance of the Support Vector Machine Advances in Neural Networks, Volume 3496 of the series Lecture Notes in Computer Science, 831-836.

[9]. Gang Wu and Edward Y. Chang (2003) Adaptive feature-space conformal transformation for imbalanced-data learning, Proceedings of the Twentieth International Conference on Machine Learn- ing (ICML 2003), Washington DC.

[10]. Si Wu and Shun-ichiAmari (2002) Conformal transformation of kernel functions: A data-dependent way to improve the performance of support vector machine classifiers, Neural Processing Letter, 15, 59-67.

[11]. Ban Zheng, Eric Moulines and Frederic Abergel (2013) Price Jump Prediction in a Limit Order Book, Journal of Mathematical Finance, Vol. 3, 242-255.

Published: Volume 2017, Issue 6 / June 25, 2017