Weighted Kernel Regression for Predicting Changing Dependencies

Steven Busuttil and Yuri Kalnishkan

Computer Learning Research Centre and Department of Computer Science,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom.
{steven,yura}@cs.rhul.ac.uk

Abstract. We want to make predictions in the online mode of learning for data where the dependence of the outcome y on the signal x can change with time. Standard regression techniques give all training examples the same weight; however, it is clear that older examples are less representative of the current dependency. Therefore, we require methods that consider the information content of examples to decay with time. We propose two methods for doing this: one naive and another, which is based on the Aggregating Algorithm (AA). Surprisingly these two techniques are computationally similar. To measure the empirical performance of these new methods, we perform experiments on options implied volatility data provided by the Russian Trading System Stock Exchange (RTSSE). In these experiments our methods perform better than the proprietary state-of-the-art technique currently used at the RTSSE.

1 Introduction

Consider the case where we are given signal-outcome (x, y) pairs (or examples) where the dependency of y on x can change with time. An example of this is when we want to predict financial option implied volatility which depends on the strike price and the time to maturity (for more information see [1]). Predicting the implied volatility of options is our main example and inspiration, but the methods we propose are not limited to it. Standard regression techniques, like Ridge Regression, treat all training examples equally, while we want recent examples to be given more importance in the same spirit of GARCH. GARCH [1, Chap. 19]) is a technique used in finance that uses weights that increase exponentially. It is usually applied to historical volatility.

In Sect. 3 we propose two new methods for achieving this. One is based on previous work in regression, which we modify to include weights, and the other is an application of the Aggregating Algorithm (AA) [2]. The Aggregating Algorithm optimally merges experts' predictions. In this case we apply it to all the linear predictors that can change with time. Kernels are then used to introduce nonlinearity. Empirical results in Sect. 4 on options implied volatility data show that our methods perform well.

2 Background

2.1 Online Linear Regression

We can define online regression by the following protocol. At every moment in time $t = 1, 2, \dots$, the value of a signal $\mathbf{x}_t \in X$ arrives. Statistician (or Learner) S observes $\mathbf{x}_t$ and then outputs a prediction $\gamma_t \in \mathbb{R}$. Finally, the outcome $y_t \in \mathbb{R}$ arrives. The set X is a signal space which is assumed to be known to Statistician in advance. We will be referring to a signal-outcome pair as an example.

The performance of S is measured by the sum of squared discrepancies between the predictions and the outcomes (known as square loss). Therefore on trial t Statistician S suffers loss $(y_t - \gamma_t)^2$. The losses incurred over several trials sum up to the overall loss. Thus after T trials, the total loss of S is

$$\mathrm{L}_T(S) = \sum_{t=1}^{T} (y_t - \gamma_t)^2 \ .$$

Clearly, a smaller value of $\mathrm{L}_T(S)$ means a better predictive performance.

In linear regression we are interested in the case where $X = \mathbb{R}^n$. Let us model the data by the linear equation[1]

$$y_t = \langle \mathbf{w}, \mathbf{x}_t \rangle + \varepsilon_t \ , \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ and $\varepsilon_t \in \mathbb{R}$ is some noise.

The method of Least Squares (LS) was derived independently by Legendre and Gauss in 1805 and 1809 respectively. At time T it aims to find a solution to (1) (i.e., a $\mathbf{w}_{\mathrm{L}}$) that minimises the overall sum of square losses over the previously seen examples

$$\mathcal{L}_T(\mathrm{LS}) = \sum_{t=1}^{T-1} (y_t - \langle \mathbf{w}_{\mathrm{L}}, \mathbf{x}_t \rangle)^2 \ .$$

This translates to solving the system of linear equations

$$\mathbf{w}_{\mathrm{L}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \ ,$$

where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{T-1})'$ and $\mathbf{y} = (y_1, \dots, y_{T-1})'$.

Least Squares runs into problems when some features in $\mathbf{X}$ are highly correlated because the matrix $\mathbf{X}'\mathbf{X}$ becomes close to singular, resulting in unstable solutions. Ridge Regression (RR), first introduced to statistics in [3], differs from Least Squares in that at time T its objective is to minimise

$$\mathcal{L}_T(\mathrm{RR}) = a\|\mathbf{w}_{\mathrm{R}}\|^2 + \sum_{t=1}^{T-1} (y_t - \langle \mathbf{w}_{\mathrm{R}}, \mathbf{x}_t \rangle)^2 \ ,$$

[1] As usual, all vectors are identified with one-column matrices and $\mathbf{B}'$ stands for the transpose of matrix $\mathbf{B}$.

where a is a fixed positive real number. RR's solution is

$$\mathbf{w}_{\mathrm{R}} = (a\mathbf{I} + \mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \ ,$$

where $\mathbf{I}$ is the identity matrix[2]. Apart from stabilising the solution (since $a > 0$ the matrix $(a\mathbf{I} + \mathbf{X}'\mathbf{X})$ is positive definite and therefore nonsingular), this technique also includes regularisation in that it favours a $\mathbf{w}_{\mathrm{R}}$ with smaller elements. This reduces the complexity of the solution, decreasing the risk of overfitting the training data, and consequently generalises better.

2.2 The Aggregating Algorithm (AA)

In this section we will be giving an overview of the Aggregating Algorithm (AA) mostly following [2, Sects. 1 and 2]. Let Ω be an outcome space, Γ be a prediction space and Θ be a (possibly infinite) pool of experts. We consider the following game between Statistician (or Learner) S, Nature, and Θ:

```
for $t = 1, 2, \dots$ do
    Every expert $\theta \in \Theta$ makes a prediction $\gamma_t^{(\theta)} \in \Gamma$
    Statistician $S$ observes all $\gamma_t^{(\theta)}$
    Statistician $S$ outputs a prediction $\gamma_t \in \Gamma$
    Nature outputs $\omega_t \in \Omega$
end for
```

Given a fixed loss function $\lambda : \Omega \times \Gamma \to [0, \infty]$, Statistician aims to suffer a cumulative loss $\mathrm{L}_T(S) = \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t)$ that is not much larger than the loss $\mathrm{L}_T(\theta) = \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t^{(\theta)})$ of the best expert $\theta \in \Theta$. The AA takes two parameters, a prior probability distribution P_0 in the pool of experts Θ and a learning rate $\eta > 0$. Let $\beta = e^{-\eta}$.

We will first describe the Aggregating Pseudo Algorithm (APA) that does not output actual predictions but generalised predictions. A generalised prediction $g : \Omega \to \mathbb{R}$ is a mapping giving a value of loss for each possible outcome. At every step t, the APA updates the experts' weights such that those that suffered large loss during the previous step have their weights reduced and vice-versa:

$$P_t(d\theta) = \beta^{\lambda(\omega_t, \gamma_t^{(\theta)})} P_{t-1}(d\theta), \quad \theta \in \Theta \ .$$

At time t, the APA chooses a generalised prediction by

$$g_t(\omega) = \log_\beta \int_\Theta \beta^{\lambda(\omega, \gamma_t^{(\theta)})} P_{t-1}^*(d\theta) \ ,$$

where $P_{t-1}^*(d\theta) = P_{t-1}(d\theta)/P_{t-1}(\Theta)$. This guarantees that for any learning rate $\eta > 0$, prior P_0, and $T = 1, 2, \dots$ (see [2, Lemma 1])

$$\mathrm{L}_T(\mathrm{APA}) = \log_\beta \int_\Theta \beta^{\mathrm{L}_T(\theta)} P_0(d\theta) \ . \tag{2}$$

[2] In general, we will not be specifying the rank of identity matrices.

To get a prediction from the generalised prediction $g_t(\omega)$ (note that we use ω since we do not yet know the real outcome of step t, ω_t) the AA uses a substitution function Σ mapping generalised predictions into Γ. A substitution function may introduce extra loss; however, in many cases perfect substitution is possible. We say that the loss function λ is η-mixable if there is a substitution function Σ such that

$$\lambda(\omega_t, \Sigma(g_t(\omega))) \leq g_t(\omega_t) \tag{3}$$

on every step t, all experts' predictions and all outcomes. The loss function λ is mixable if it is η-mixable for some $\eta > 0$.

Suppose that our loss function is η-mixable. Using (3) and (2) we can obtain the following upper bound on the cumulative loss of the AA:

$$\mathrm{L}_T(\mathrm{AA}) \leq \log_\beta \int_\Theta \beta^{\mathrm{L}_T(\theta)} P_0(d\theta) \ .$$

The Square Loss Game. In this paper we are concerned with the (bounded) square loss game (see [2, Sect 2.4]), where $\Omega = [-Y, Y]$, $Y \in \mathbb{R}$, $\Gamma = \mathbb{R}$, and $\lambda(\omega, \gamma) = (\omega - \gamma)^2$. The square loss game is η-mixable if and only if $\eta \leq 1/(2Y^2)$. A perfect substitution function for this game is

$$\gamma = \frac{g(-Y) - g(Y)}{4Y} \ . \tag{4}$$

The Aggregating Algorithm for Regression. The AA was applied to the problem of linear regression resulting in the Aggregating Algorithm for Regression (AAR). AAR merges all the linear predictors that map signals to outcomes [2, Sect. 3] (a Gaussian prior is assumed on the pool of experts). This results in the following solution to the regression problem

$$\mathbf{w}_{\mathrm{A}} = (a\mathbf{I} + \widetilde{\mathbf{X}}'\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}'\widetilde{\mathbf{y}} \ , \tag{5}$$

where $\widetilde{\mathbf{X}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)'$ and $\widetilde{\mathbf{y}} = (y_1, y_2, \ldots, y_{T-1}, 0)'$. It can be shown (see [4]) that at time T the AAR solution $\mathbf{w}_{\mathrm{A}}$ minimises

$$\mathcal{L}_T(\mathrm{AAR}) = a\|\mathbf{w}_{\mathrm{A}}\|^2 + \langle \mathbf{w}_{\mathrm{A}}, \mathbf{x}_T \rangle^2 + \sum_{t=1}^{T-1} (y_i - \langle \mathbf{w}_{\mathrm{A}}, \mathbf{x}_t \rangle)^2 \ .$$

The main property of AAR is that it is optimal in the sense that the total loss it suffers is only a little worse than that of any linear predictor. By the latter we mean a strategy that predicts $\theta'\mathbf{x}_t$ on every trial t, where $\theta \in \mathbb{R}^n$ is some fixed vector. The set of all linear predictors may be identified with $\mathbb{R}^n$.

It is interesting to note that AAR's bound does not make any assumptions on the probability distribution of the data. From (5) it is clear that in computational terms AAR is similar to Ridge Regression but with the signal-outcome pair $(\mathbf{x}_T, 0)$ added to its training set, where $\mathbf{x}_T$ is the new signal for which a prediction is to be made. This makes predictions shrink towards 0, with the goal of making them even more resistant to overfitting (it is assumed that the mean of the outcomes is 0).

2.3 Kernel Methods

The use of linear methods like RR and AAR in the real world is limited since they can only model simple linear dependencies. The kernel trick (first used in this context in [5]) is now a widely used technique which can make a linear algorithm operate in feature space without the inherent complexities. For a function $k : X \times X \to \mathbb{R}$ to be a kernel it has to be symmetric, and for all ℓ and all $\mathbf{x}_1, \ldots, \mathbf{x}_\ell \in X$, the kernel matrix $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$, $i, j = 1, \ldots, \ell$ must be positive semi-definite (have nonnegative eigenvalues). Equivalently, a kernel function k takes two vectors and returns their dot product in some feature space, $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where ϕ is a (nonlinear) transformation to feature space.

Through kernel functions it is therefore possible to perform linear regression in feature space which would be equivalent to performing nonlinear regression in input space. Accordingly, RR and AAR have been reduced into a formulation known as dual variables (see [6] and [7] respectively), where all the signals appear only in dot products. This makes transforming the linear models into nonlinear ones simply a matter of replacing the dot products with a kernel function. The resulting methods, which we shall call Kernel Ridge Regression (KRR) and the Kernel Aggregating Algorithm for Regression (KAAR), respectively calculate the prediction γ for a new example $\mathbf{x}_T$ as follows:

$$\gamma_{\text{KRR}} = \mathbf{y}'(a\mathbf{I} + \mathbf{K})^{-1}\mathbf{k} \ , \tag{6}$$

where $\mathbf{k} = (k(\mathbf{x}_i, \mathbf{x}_T))$ and $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$, $i, j = 1, \ldots, T-1$, and,

$$\gamma_{\text{KAAR}} = \widetilde{\mathbf{y}}'(a\mathbf{I} + \widetilde{\mathbf{K}})^{-1}\widetilde{\mathbf{k}} \ ,$$

where $\widetilde{\mathbf{y}} = (\mathbf{y}', 0)'$, $\widetilde{\mathbf{K}} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$, $i, j = 1, \ldots, T$, and $\widetilde{\mathbf{k}} = \left(\mathbf{k}', k(\mathbf{x}_T, \mathbf{x}_T)\right)'$.

2.4 Controlled KAAR

Controlled KAAR (CKAAR) [4] is a generalisation of both KRR and KAAR. At time T the linear version of CKAAR aims to find a solution $\mathbf{w}_{\text{C}}$ that minimises

$$\mathcal{L}_T(\text{CKAAR}) = a\|\mathbf{w}_{\text{C}}\|^2 + b\langle \mathbf{w}_{\text{C}}, \mathbf{x}_T \rangle^2 + \sum_{i=1}^{T-1} (y_i - \langle \mathbf{w}_{\text{C}}, \mathbf{x}_i \rangle)^2 \ ,$$

where $b \geq 0$. It is clear that when $b = 0$, CKAAR is equivalent to RR and conversely, equivalent to AAR when $b = 1$. Empirical results in [4] suggest that in general, the performance of CKAAR is as good as or better than that of both KAAR and KRR. The linear CKAAR solution to the regression problem is

$$\mathbf{w}_{\text{C}} = (a\mathbf{I} + \widehat{\mathbf{X}}'\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}'\widetilde{\mathbf{y}} \ ,$$

where $\widehat{\mathbf{X}} = (\mathbf{X}', \sqrt{b}\,\mathbf{x}_T)'$ and $\widetilde{\mathbf{y}} = (y_1, y_2, \ldots, y_{T-1}, 0)'$. The kernel version of CKAAR makes a prediction for a new signal $\mathbf{x}_T$ in the following way:

$$\gamma_{\text{CKAAR}} = \widetilde{\mathbf{y}}'(a\mathbf{I} + \widehat{\mathbf{K}})^{-1}\widehat{\mathbf{k}} \ ,$$

where $\widehat{\mathbf{k}} = \left(\mathbf{k}', \sqrt{b}\,k(\mathbf{x}_T, \mathbf{x}_T)\right)'$ and $\widehat{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & \sqrt{b}\,\mathbf{k} \\ \sqrt{b}\,\mathbf{k}' & b\,k(\mathbf{x}_T, \mathbf{x}_T) \end{bmatrix}$.

3 Weighted Kernel Regression

3.1 WeCKAAR

We propose using a modified version of CKAAR, which we call Weighted CK-AAR (WeCKAAR) to make predictions where the underlying dependency of outcomes on signals can change with time. We simply introduce a decaying factor, such that old transactions are given less importance. Therefore, the objective is to find a $\mathbf{w}$ that minimises

$$\mathcal{L}_T(\mathrm{W}) = a\|\mathbf{w}\|^2 + b\langle \mathbf{w}, \mathbf{x}\rangle^2 + \sum_{t=1}^{T-1} d_t(y_t - \langle \mathbf{w}, \mathbf{x}_t\rangle^2) \ , \tag{7}$$

where $d_t \in \mathbb{R}$ are nonnegative weights that increase with t. Let $d_T = b$ and $\mathbf{D} = \mathrm{diag}(d_1, \dots, d_T)$ be the diagonal matrix with elements $d_1 \dots d_T$. Equation (7) can be rewritten as

$$\begin{aligned}\mathcal{L}_T(\mathrm{W}) &= a\|\mathbf{w}\|^2 + \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}\mathbf{w}\right)' \mathbf{D} \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}\mathbf{w}\right) \\ &= a\mathbf{w}'\mathbf{w} + \widetilde{\mathbf{y}}'\mathbf{D}\widetilde{\mathbf{y}} + \mathbf{w}'\widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{X}}\mathbf{w} - 2\mathbf{w}'\widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{y}} \ .\end{aligned}$$

If we differentiate this with respect to $\mathbf{w}$, divide throughout by 2 and make it equal to zero, we get

$$\frac{1}{2}\frac{\partial \mathcal{L}_T(\mathrm{W})}{\partial \mathbf{w}} = a\mathbf{w} + \widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{X}}\mathbf{w} - \widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{y}} = 0 \ .$$

This implies that $\mathbf{w} = \left(\widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{X}} + a\mathbf{I}\right)^{-1} \widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{y}}$.

Dual (Kernel) Form. Using Lemma 1 we can obtain a form of WeCKAAR's prediction where signals appear only in dot products. Accordingly, a prediction for the signal $\mathbf{x}_T$ is

$$\gamma_T = \mathbf{w}'\mathbf{x}_T = \widetilde{\mathbf{y}}'\sqrt{\mathbf{D}}\left(\sqrt{\mathbf{D}}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}'\sqrt{\mathbf{D}} + a\mathbf{I}\right)^{-1} \sqrt{\mathbf{D}}\widetilde{\mathbf{X}}\mathbf{x}_T \ ,$$

where $\sqrt{\mathbf{D}} = \mathrm{diag}(\sqrt{d_1}, \dots, \sqrt{d_T})$. We now apply the kernel trick by replacing dot products with kernel functions k to obtain the kernel version of WeCKAAR:

$$\gamma_T = \widetilde{\mathbf{y}}'\sqrt{\mathbf{D}}\left(\sqrt{\mathbf{D}}\widetilde{\mathbf{K}}\sqrt{\mathbf{D}} + a\mathbf{I}\right)^{-1} \sqrt{\mathbf{D}}\widetilde{\mathbf{k}} \ , \tag{8}$$

where

$$\sqrt{\mathbf{D}}\widetilde{\mathbf{K}}\sqrt{\mathbf{D}} = \begin{bmatrix} d_1 k(\mathbf{x}_1, \mathbf{x}_1) & \sqrt{d_1 d_2} k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \sqrt{d_1 d_T} k(\mathbf{x}_1, \mathbf{x}_T) \\ \sqrt{d_2 d_1} k(\mathbf{x}_2, \mathbf{x}_1) & d_2 k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \sqrt{d_2 d_T} k(\mathbf{x}_2, \mathbf{x}_T) \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{d_T d_1} k(\mathbf{x}_T, \mathbf{x}_1) & \sqrt{d_T d_2} k(\mathbf{x}_T, \mathbf{x}_2) & \cdots & d_T k(\mathbf{x}_T, \mathbf{x}_T) \end{bmatrix} \ .$$

3.2 KAARCh

For our second new method, we apply the Aggregating Algorithm (AA) to the regression problem where the experts can change with time. We call this method the Kernel Aggregating Algorithm for Regression with Changing underlying dependencies (KAARCh). The main idea behind this method is to apply the Aggregating Algorithm to the case where the pool of experts is made up of all linear predictors that can change with time. We assume that outcomes are bounded by Y, therefore, for any t, $y_t \in [-Y, Y]$ (we do not require our algorithm to know Y). We are interested in the square loss, therefore we will be using optimal $\eta = 1/(2Y^2)$ and substitution function (4).

An expert is a sequence $\theta_1, \theta_2, \ldots$, that at time T predicts

$$\mathbf{x}'_T(\theta_1 + \theta_2 + \ldots + \theta_T) \ ,$$

where for any t, $\theta_t \in \mathbb{R}^n$ and $\mathbf{x}_t \in \mathbb{R}^n$. To apply the AA to this problem we need to define a lower triangular block matrix $\mathbf{L}$, and θ which is a concatenation of all the θ_t for $t = 1 \ldots T$, such that

$$\mathbf{L}\theta = \begin{bmatrix} \mathbf{I}\,\mathbf{0} \cdots\cdots \mathbf{0} \\ \mathbf{I}\,\mathbf{I} \ddots \quad \vdots \\ \vdots\ \vdots \ddots \ddots \vdots \\ \mathbf{I}\,\mathbf{I} \cdots\ \mathbf{I}\ \mathbf{0} \\ \mathbf{I}\,\mathbf{I} \cdots\ \mathbf{I}\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{T-1} \\ \theta_T \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_1 + \theta_2 \\ \vdots \\ \theta_1 + \theta_2 + \cdots + \theta_{T-1} \\ \theta_1 + \theta_2 + \cdots + \theta_{T-1} + \theta_T \end{bmatrix} .$$

The matrices $\mathbf{I}$ and $\mathbf{0}$ in $\mathbf{L}$ are the $n \times n$ identity and all-zero matrices respectively. We also need to define $\mathbf{z}_t$ which is $\mathbf{x}_t$ padded with zeros in the following way

$$\mathbf{z}_t = \begin{bmatrix} \underbrace{0 \cdots 0}_{n(t-1)} & \mathbf{x}'_t & \underbrace{0 \cdots 0}_{n(T-t)} \end{bmatrix}' \quad \text{so that } \mathbf{z}'_t \mathbf{L}\theta = \mathbf{x}'_t(\theta_1 + \theta_2 + \ldots + \theta_t) \ .$$

Let $a > 0$ be an arbitrary constant. Consider the prior distribution P_0 in the set $\mathbb{R}^{nT}$ of possible weights θ with the Gaussian density

$$P_0(d\theta) = \left(\frac{a\eta}{\pi}\right)^{nT/2} e^{-a\eta \sum_{t=1}^T \|\theta_t\|^2} d\theta \ .$$

The loss of θ over the first T trials is

$$\mathrm{L}_T(\theta) = \sum_{t=1}^T (y_t - \mathbf{z}'_t \mathbf{L}\theta)^2 = \theta' \mathbf{L}' \left(\sum_{t=1}^T \mathbf{z}_t \mathbf{z}'_t\right) \mathbf{L}\theta - 2\left(\sum_{t=1}^T y_t \mathbf{z}'_t\right) \mathbf{L}\theta + \sum_{t=1}^T y_t^2 \ .$$

Therefore, the loss of the APA is (recall that $\beta = e^{-\eta}$)

$$\begin{aligned} \mathrm{L}_T(\mathrm{APA}) &= \log_\beta \int_{\mathbb{R}^{nT}} \beta^{\mathrm{L}_T(\theta)} P_0(d\theta) \\ &= \log_\beta \int_{\mathbb{R}^{nT}} \left(\frac{a\eta}{\pi}\right)^{nT/2} e^{-\eta\left(\theta'\mathbf{L}'\left(\sum_{t=1}^T \mathbf{z}_t\mathbf{z}'_t\right)\mathbf{L}\theta - 2\left(\sum_{t=1}^T y_t \mathbf{z}'_t\right)\mathbf{L}\theta + \sum_{t=1}^T y_t^2 + \theta' a\mathbf{I}\theta\right)} d\theta \\ &= \log_\beta \int_{\mathbb{R}^{nT}} \left(\frac{a\eta}{\pi}\right)^{nT/2} e^{-\eta\theta'\left(\mathbf{L}'\sum_{t=1}^T \mathbf{z}_t\mathbf{z}'_t\mathbf{L} + a\mathbf{I}\right)\theta + 2\eta\left(\sum_{t=1}^T y_t\mathbf{z}'_t\right)\mathbf{L}\theta - \eta\sum_{t=1}^T y_t^2} d\theta \ . \end{aligned}$$

Given the generalised prediction $g_T(\omega)$ which is the APA's loss with variable $\omega \in \mathbb{R}$ replacing y_T and using substitution function (4), the AA's prediction is

$$\begin{aligned}\gamma_T &= \frac{1}{4Y}\log_\beta \frac{\beta^{g_T(-Y)}}{\beta^{g_T(Y)}} \\ &= \frac{1}{4Y}\log_\beta \frac{\int_{\mathbb{R}^{nT}} e^{-\eta\theta'\left(\mathbf{L}'\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t'\mathbf{L}+a\mathbf{I}\right)\theta+2\eta\left(\sum_{t=1}^{T-1}y_t\mathbf{z}_t'\mathbf{L}-Y\mathbf{z}_T'\mathbf{L}\right)\theta}d\theta}{\int_{\mathbb{R}^{nT}} e^{-\eta\theta'\left(\mathbf{L}'\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t'\mathbf{L}+a\mathbf{I}\right)\theta+2\eta\left(\sum_{t=1}^{T-1}y_t\mathbf{z}_t'\mathbf{L}+Y\mathbf{z}_T'\mathbf{L}\right)\theta}d\theta} \ .\end{aligned}$$

Let $Q_1(\theta) = \theta'\left(\mathbf{L}'\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t'\mathbf{L}+a\mathbf{I}\right)\theta - 2\left(\sum_{t=1}^{T-1}y_t\mathbf{z}_t'\mathbf{L}-Y\mathbf{z}_T'\mathbf{L}\right)\theta$, and $Q_2(\theta) = \theta'\left(\mathbf{L}'\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t'\mathbf{L}+a\mathbf{I}\right)\theta - 2\left(\sum_{t=1}^{T-1}y_t\mathbf{z}_t'\mathbf{L}+Y\mathbf{z}_T'\mathbf{L}\right)\theta$. By Lemma 2

$$\gamma_T = \frac{1}{4Y}\log_\beta \frac{e^{-\eta\min_{\theta\in\mathbb{R}^{nT}}Q_1(\theta)}}{e^{-\eta\min_{\theta\in\mathbb{R}^{nT}}Q_2(\theta)}} = \frac{1}{4Y}\left(\min_{\theta\in\mathbb{R}^{nT}}Q_1(\theta) - \min_{\theta\in\mathbb{R}^{nT}}Q_2(\theta)\right) \ .$$

Finally, by using Lemma 3 we get

$$\gamma_T = \sum_{t=1}^{T-1} y_t\mathbf{z}_t'\mathbf{L}\left(\mathbf{L}'\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t'\mathbf{L}+a\mathbf{I}\right)^{-1}\mathbf{L}'\mathbf{z}_T \ . \tag{9}$$

Dual (Kernel) Form. Equation (9) can be rewritten in matrix notation $\gamma_T = \mathbf{y}'\mathbf{Z}\mathbf{L}\left(\mathbf{L}'\widetilde{\mathbf{Z}}'\widetilde{\mathbf{Z}}\mathbf{L}+a\mathbf{I}\right)^{-1}\mathbf{L}'\mathbf{z}_T$, where $\mathbf{Z} = (\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_{T-1})'$ and $\widetilde{\mathbf{Z}} = (\mathbf{Z}',\mathbf{z}_T)'$. We can get a dual formulation of this by using Lemma 1:

$$\gamma_T = \widetilde{\mathbf{y}}'\left(\widetilde{\mathbf{Z}}\mathbf{L}\mathbf{L}'\widetilde{\mathbf{Z}}'+a\mathbf{I}\right)^{-1}\widetilde{\mathbf{Z}}\mathbf{L}\mathbf{L}'\mathbf{z}_T \ .$$

Since all signals appear in dot products, we can use the kernel trick to introduce nonlinearity. In this case we get

$$\gamma_T = \widetilde{\mathbf{y}}'\left(\bar{\mathbf{K}}+a\mathbf{I}\right)^{-1}\bar{\mathbf{k}} \ , \tag{10}$$

where

$$\bar{\mathbf{K}} = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & k(\mathbf{x}_1,\mathbf{x}_3) & \cdots & k(\mathbf{x}_1,\mathbf{x}_T) \\ k(\mathbf{x}_2,\mathbf{x}_1) & 2k(\mathbf{x}_2,\mathbf{x}_2) & 2k(\mathbf{x}_2,\mathbf{x}_3) & \cdots & 2k(\mathbf{x}_2,\mathbf{x}_T) \\ k(\mathbf{x}_3,\mathbf{x}_1) & 2k(\mathbf{x}_3,\mathbf{x}_2) & 3k(\mathbf{x}_3,\mathbf{x}_3) & \cdots & 3k(\mathbf{x}_3,\mathbf{x}_T) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_T,\mathbf{x}_1) & 2k(\mathbf{x}_T,\mathbf{x}_2) & 3k(\mathbf{x}_T,\mathbf{x}_3) & \cdots & Tk(\mathbf{x}_T,\mathbf{x}_T) \end{bmatrix}, \bar{\mathbf{k}} = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_T) \\ 2k(\mathbf{x}_2,\mathbf{x}_T) \\ 3k(\mathbf{x}_3,\mathbf{x}_T) \\ \vdots \\ Tk(\mathbf{x}_T,\mathbf{x}_T) \end{bmatrix} \ .$$

Implementation Notes. Consider the case where we have no data for some particular point in time t. In this case we should set the tth row and column of $\mathbf{L}$ all equal to 0. This can be done an arbitrary number of times, which will effectively result in $\bar{\mathbf{K}}$ and $\bar{\mathbf{k}}$ having integer coefficients that grow with possibly different steps. Since a kernel multiplied by any scalar is still a kernel, we can use real numbers for these coefficients, representing the actual real-world time at which an example arrives. Therefore, the coefficients $1,\ldots,T$ in $\bar{\mathbf{K}}$ and $\bar{\mathbf{k}}$ can be replaced with nonnegative increasing real numbers $t_1,\ldots,t_T$.

4 Empirical Results

4.1 Options Implied Volatility

The Russian Trading System Stock Exchange (RTSSE) have provided us with data containing the details of options transactions on several underlying assets. Options (see [1] for more detailed information on options) are a type of derivative securities. They give the right to sell (put option) or buy (call option) an asset (like stock) which has current price S_t at some particular strike price K at a particular point in time in the future (at maturity). On a stock market, derivative securities are mainly used for hedging, that is, as an insurance against possible changes in the value of the underlying asset. Given this, what should the price of an option be?

The most popular approach to pricing options is based on the Black-Scholes theory. This assumes that S_t follows an exponential Wiener process with constant volatility σ. The parameter σ cannot be observed directly, but it can be estimated from market data such as the price history (this estimate is called historical volatility). There are different types of options; we are interested in the so called European options. In this case the price at time t of call and put options, which we will denote by c_t and p_t respectively, are calculated by

$$c_t = S_t N(d_1) - K N(d_2), \text{ and } p_t = c_t + K - S_t \ ,$$

where $N(x)$ is the probability density function of the normal distribution with mean 0 and standard deviation 1, and given T which is the time until maturity in years,

$$d_1 = \frac{\ln(S_t/K) + (\sigma^2/2)T}{\sigma\sqrt{T}}, \text{ and } d_2 = d_1 - \sigma\sqrt{T} \ .$$

In practise this model is often violated. Given the current prices of options and the underlying asset we can find σ that satisfies the formulae above. This σ is known as the implied volatility and it exhibits a dependence on the strike price and the time to maturity. The curve showing the dependence of the implied volatility on the strike price is often called the volatility smile (see [1, Chap. 16]). If time to maturity is taken into account too, we get a volatility surface. The existence of volatility smiles and surfaces contradicts the Black-Scholes model. There is no generally recognised theory describing the phenomenon of implied volatility; however, it remains a useful parameter and traders at a stock exchange often use it to quote option prices.

We are interested in using learning theory methods for predicting implied volatilities without assuming any model for its behaviour. In our experiments we treat the implied volatility of a transaction as the outcome and the parameters of the transaction and other market information (such as the current price of the underlying asset) as the signal.

4.2 Experimentation Methodology

As usual, we need to find good values for any tunable parameters of the methods employed. For the weights required by our new methods, specifically, WeCKAAR's $d_1, \ldots, d_T$ and KAARCh's $t_1, \ldots, t_T$, we use a real number representing the (normalised) time at which the transactions occurred. In our experiments we use the spline kernel[3] (see, for example, [8]) which does not require any parameters. Therefore we only need to find values for the parameter a (see (6), (8), and (10)). We do this by the following crude procedure. We apply a sliding window approach (the window size was set to 50) and find a good value for a on the first window; then we make predictions on the next window (therefore, the parameter a is updated every 50 transactions). This is repeated for the whole dataset. WeCKAAR and KAARCh function better if the mean of the outcomes is 0. We achieve this by shifting the outcomes down by the mean of the outcomes of the previous window and later shift the predictions up by this same amount.

4.3 Results

In Fig. 1 we show results on options data for EERU, GAZP and RTSI. EERU are options on futures on shares of Unified Power Systems of Russia, GAZP are options on futures on shares of Gazprom, and RTSI are options on an RTSI index. The results show the cumulative square loss (therefore a smaller value is better) suffered by the proprietary method used at the RTSSE, WeCKAAR, KAARCh and KRR (applied using the same sliding window technique). The proprietary method used at the RTSSE is based on Kalman Filters and some assumptions on the shape of the volatility curve are made. This contrasts with our methods, where we make no assumptions at all (our methods' applicability is not limited to options data) and use general purpose kernels.

5 Discussion

In the experiments carried out, both our methods performed better than the proprietary method used at the RTSSE (and Kernel Ridge Regression). It also seems that, in general, KAARCh may be better than the simpler method WeCKAAR. This is understandable, since KAARCh has a more advanced underlying theory. Recall that the proprietary method used at the RTSSE was specifically designed for this application. Moreover, it is constantly monitored and tuned by experts to predict implied volatility better. Our methods are applicable to a much more general area and still manage to perform better (even with the crude experimentation methodology described). In fact, KAARCh and WeCKAAR can be used wherever it makes sense to assume that the underlying dependency of outcomes on signals can change with time.

Future work includes deriving a theoretical upper bound on the loss of KAARCh and a better investigation of the empirical performance of our methods on financial and other data, possibly using a better experimentation procedure.

[3] The linear, polynomial and RBF kernels were also considered.

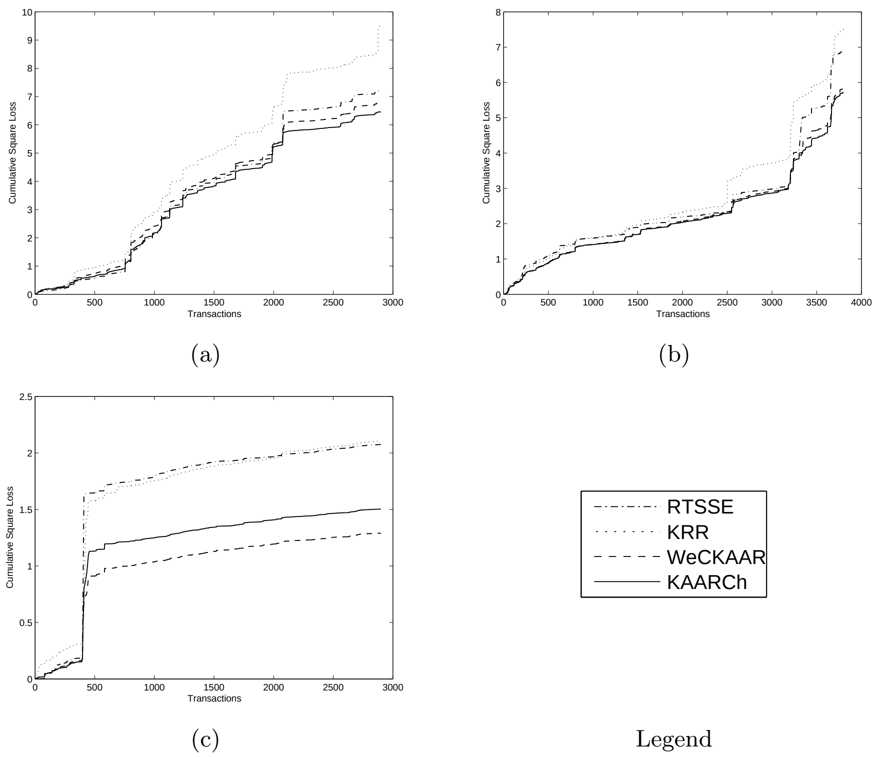

Fig. 1. Predictions on transactions for options on (a) EERU, (b) GAZP, and (c) RTSI.

Acknowledgements. The authors are grateful to Dr Michael Vyugin at the RTSSE for providing the data and sharing his expertise with us. We would also like to thank Prof Volodya Vovk and Prof Alex Gammerman for useful discussions and comments.

References

1. Hull, J.C.: Options, Futures and Other Derivatives. 6th edn. Prentice Hall (2005)
2. Vovk, V.: Competitive on-line statistics. International Statistical Review **69**(2) (2001) 213–248
3. Hoerl, A.E.: Application of ridge analysis to regression problems. Chemical Engineering Progress **58** (1962) 54–59
4. Busuttil, S., Kalnishkan, Y., Gammerman, A.: Improving the aggregating algorithm for regression. In: Proceedings of the 25th IASTED International Conference on Artificial Intelligence and Applications (AIA 2007), ACTA Press (2007) 347–352
5. Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control **25** (1964) 821–837

6. Saunders, C., Gammerman, A., Vovk, V.: Ridge regression learning algorithm in dual variables. In: Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann (1998) 515–521
7. Gammerman, A., Kalnishkan, Y., Vovk, V.: On-line prediction with kernels and the complexity approximation principle. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press (2004) 170–176
8. Schölkopf, B., Smola, A.J.: Learning with Kernels — Support Vector Machines, Regularization, Optimization and Beyond. The MIT Press, USA (2002)
9. Beckenbach, E.F., Bellman, R.: Inequalities. Springer (1961)

A Lemmas

Lemma 1. *Given a matrix* $\mathbf{A}$*, a scalar* a *and* $\mathbf{I}$ *identity matrices of the appropriate size,*

$$(\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1} \ .$$

Proof.

$$\begin{aligned}(\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}\mathbf{A} &= (\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}\mathbf{A}(\mathbf{A}'\mathbf{A} + a\mathbf{I})(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1} \\ &= (\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}(\mathbf{A}\mathbf{A}'\mathbf{A} + a\mathbf{A})(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1} \\ &= (\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}(\mathbf{A}\mathbf{A}' + a\mathbf{I})\mathbf{A}(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1} \\ &= \mathbf{A}(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1}\end{aligned}$$

Lemma 2. *Let* $Q(\theta) = \theta'\mathbf{A}\theta + \mathbf{b}'\theta + c$*, where* $\theta, \mathbf{b} \in \mathbb{R}^n$*,* c *is a scalar and* $\mathbf{A}$ *is a symmetric positive definite* $n \times n$ *matrix. Then*

$$\int_{\mathbb{R}^n} e^{-Q(\theta)} d\theta = e^{-Q_0} \frac{\pi^{n/2}}{\sqrt{\det \mathbf{A}}} \ ,$$

where $Q_0 = \min_{\theta \in \mathbb{R}^n} Q(\theta)$*.*

Proof. Let $\theta_0 \in \arg\min Q$. Take $\xi = \theta - \theta_0$ and $\widetilde{Q}(\xi) = Q(\xi + \theta_0)$. It is easy to see that the quadratic part of $\widetilde{Q}$ is $\xi'\mathbf{A}\xi$. Since $0 \in \arg\min \widetilde{Q}$, the form has no linear term. Indeed, in the vicinity of 0 the linear term dominates over the quadratic term; if $\widetilde{Q}$ has a non-zero linear term, it cannot have a minimum at 0. Since $Q_0 = \min_{\xi \in \mathbb{R}^n} \widetilde{Q}(\xi)$, we can conclude that the constant term in $\widetilde{Q}$ is Q_0. Thus $\widetilde{Q}(\xi) = \xi'\mathbf{A}\xi + Q_0$.

It remains to show that $\int_{\mathbb{R}^n} e^{-\xi'\mathbf{A}\xi} d\xi = \pi^{n/2}/\sqrt{\det \mathbf{A}}$. This can be proved by considering a basis where $\mathbf{A}$ diagonalises (or see [9, Sect. 2.7, Theorem 3]).

Lemma 3. *Let*

$$F(\mathbf{A}, \mathbf{b}, \mathbf{x}) = \min_{\theta \in \mathbb{R}^n}(\theta'\mathbf{A}\theta + \mathbf{b}'\theta + \mathbf{x}'\theta) - \min_{\theta \in \mathbb{R}^n}(\theta'\mathbf{A}\theta + \mathbf{b}'\theta - \mathbf{x}'\theta) \ ,$$

where $\mathbf{b}, \mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{A}$ *is a symmetric positive definite* $n \times n$ *matrix. Then* $F(\mathbf{A}, \mathbf{b}, \mathbf{x}) = -\mathbf{b}'\mathbf{A}^{-1}\mathbf{x}$*.*

Proof. It can be shown by differentiation that the first minimum is achieved at $\theta_1 = -\frac{1}{2}\mathbf{A}^{-1}(\mathbf{b} + \mathbf{x})$ and the second minimum at $\theta_2 = -\frac{1}{2}\mathbf{A}^{-1}(\mathbf{b} - \mathbf{x})$. The substitution proves the lemma.