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. Consider the online regression problem where the dependence of the outcome y_t on the signal $\mathbf{x}_t$ changes with time. Standard regression techniques, like Ridge Regression, do not perform well in tasks of this type. We propose two methods to handle this problem: WeCKAAR, a simple modification of an existing regression technique, and KAARCh, an application of the Aggregating Algorithm. Empirical results on artificial data show that in this setting, KAARCh is superior to WeCKAAR and standard regression techniques. On options implied volatility data, the performance of both KAARCh and WeCKAAR is comparable to that of the proprietary technique currently being used at the Russian Trading System Stock Exchange (RTSSE).

1 Introduction

Consider the online regression problem where the dependence of the outcome y_t on the signal $\mathbf{x}_t$ changes with time. An example of this is the prediction of financial options implied volatility described in Sect. 4.2. Standard regression techniques, like Ridge Regression, treat all training examples equally. In time series theory there is a method called GARCH (see, for example, [1, Chap. 19]), which assigns exponentially decreasing weights to old examples. This method is used to estimate historical volatility in finance. We would like to extend this idea to the more general problem of online regression.

In Sect. 3 we present two methods as a solution to this problem: WeCKAAR and KAARCh. WeCKAAR is a simple method that adds decaying weights to an existing regression technique. KAARCh is a new method based on the Aggregating Algorithm (AA). The AA (see [2]) allows us to merge experts from large pools to obtain optimal strategies. To get KAARCh, the AA is used to merge all predictors that can change with time.

We report the empirical performance of these methods in Sect. 4; first on an artificial dataset, and then on options implied volatility data. These results show that when dealing with changing dependencies, KAARCh is an improvement on standard and weighted regression techniques. In addition, the performance of WeCKAAR and KAARCh on options implied volatility data provided by the Russian Trading System Stock Exchange (RTSSE) is comparable to that of the specially designed proprietary technique currently being used.

2 Background

In the online regression framework at every moment in time $t = 1, 2, \ldots$, the value of a signal $\mathbf{x}_t \in X$ arrives[1]. Statistician (or Learner) S observes $\mathbf{x}_t$ and then outputs a prediction $\gamma_t \in \mathbb{R}$, before 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 after T trials sum up to the cumulative square loss at time T,

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

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

2.1 Linear and Kernel Regression

If $X \subseteq \mathbb{R}^n$ we can consider simple linear regressors of the form $\mathbf{w} \in \mathbb{R}^n$. Given a signal $\mathbf{x} \in X$, such a regressor makes a prediction $\mathbf{w}'\mathbf{x}$. Linear methods are easy to manipulate mathematically but their use in the real world is limited since they can only model simple dependencies. The kernel trick (first used in this context in [3]) 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). For every kernel there exists a unique reproducing kernel Hilbert space (RKHS) F such that k is the reproducing kernel of F. In fact, there is a mapping $\phi : X \to F$ such that kernels can be defined as $k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$. We will be referring to any function in the RKHS F as D. Intuitively $D(\mathbf{x})$ is a decision rule in F that produces a prediction for the object $\mathbf{x}$. We will be measuring the complexity of D by its norm $\|D\|$ in F. For more information on kernels and RKHS see, for example, [4] and [5].

2.2 Ridge Regression (RR)

Ridge Regression (RR), introduced to statistics in [6], is a popular regression technique that at time T aims to find a $\mathbf{w}_{\mathrm{R}}$ that minimises

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

[1] As usual, all vectors are identified with one-column matrices and $\mathbf{A}'$ stands for the transpose of matrix $\mathbf{A}$. We will not be specifying the size of simple matrices like the identity matrix $\mathbf{I}$ when this is clear from the context.

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, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_{T-1})'$ and $\mathbf{y} = (y_1, \ldots, y_{T-1})'$. The kernel version of RR, called Kernel Ridge Regression (KRR) (see [7]) calculates the prediction for a new example $\mathbf{x}_T$ by

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

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

2.3 The Aggregating Algorithm for Regression (AAR)

The Aggregating Algorithm (AA) (see [2]), allows us to merge strategies (or experts) from large pools to obtain optimal strategies. Typically, such an optimal strategy performs nearly as good as the best expert in the pool in terms of the cumulative loss. The AA was applied to the problem of linear regression resulting in the AA for Regression (AAR) [2, Sect. 3] (also known as the Vovk-Azoury-Warmuth forecaster, see [8, Sect. 11.8]). Using a Gaussian prior, AAR merges all the static linear predictors that map signals to outcomes. AAR's solution to the regression problem is $\mathbf{w}_{\mathrm{A}} = (a\mathbf{I}+\widetilde{\mathbf{X}}'\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}'\widetilde{\mathbf{y}}$, where $\widetilde{\mathbf{X}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)'$ and $\widetilde{\mathbf{y}} = (y_1, \ldots, y_{T-1}, 0)'$. It can be shown (see [9]) that this solution minimises

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

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. In [10] AAR was kernelised to get Kernel AAR (KAAR) which makes a prediction at time T by

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

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

2.4 Controlled KAAR (CKAAR)

Controlled KAAR (CKAAR) [9] is a generalisation of both KRR and KAAR. At time T the linear version of CKAAR aims to find a solution $\mathbf{w}_{\mathrm{C}}$ that minimises

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

where $b \geq 0$. It is clear that when $b = 0$, CKAAR is equivalent to RR and equivalent to AAR when $b = 1$. Empirical results in [9] 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 is $\mathbf{w}_{\mathrm{C}} = (a\mathbf{I} + \widehat{\mathbf{X}}'\widehat{\mathbf{X}})^{-1}\widehat{\mathbf{X}}'\widetilde{\mathbf{y}}$, where $\widehat{\mathbf{X}} = (\mathbf{X}', \sqrt{b}\,\mathbf{x}_T)'$. The kernel version of CKAAR makes a prediction at time T by

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

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

3 Methods

We are interested in making predictions in online regression where the dependency of y_t on $\mathbf{x}_t$ changes with time. We present two solutions to this problem: a simple method named WeCKAAR and our new method KAARCh. It is interesting that the prediction formulae of these two methods are very similar.

3.1 WeCKAAR

Weighted CKAAR (WeCKAAR) is a simple modification of CKAAR that employs a decaying factor such that old examples are given less importance. The objective of WeCKAAR is to find a $\mathbf{w}$ that minimises

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

where $d_t \in \mathbb{R}$ are nonnegative weights that increase with t. Let $d_T = b$ and $\mathbf{D} = \operatorname{diag}(d_1, \ldots, d_T)$ be the diagonal matrix with elements $d_1 \ldots d_T$. It can be shown by differentiation that the minimum of (2) is achieved when $\mathbf{w} = \left(\widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{X}} + a\mathbf{I}\right)^{-1} \widetilde{\mathbf{X}}'\mathbf{D}\widetilde{\mathbf{y}}$. If we use the identity $(\mathbf{A}\mathbf{A}' + a\mathbf{I})^{-1}\mathbf{A} = \mathbf{A}(\mathbf{A}'\mathbf{A} + a\mathbf{I})^{-1}$ (see, for example, [10, Sect. 3.1]) to obtain the dual form of this and introduce kernels, WeCKAAR's prediction for the signal $\mathbf{x}_T$ becomes

$$\gamma_T = \widetilde{\mathbf{y}}'\sqrt{\mathbf{D}}\left(\sqrt{\mathbf{D}}\widetilde{\mathbf{K}}\sqrt{\mathbf{D}} + a\mathbf{I}\right)^{-1}\sqrt{\mathbf{D}}\widetilde{\mathbf{k}} \ , \tag{3}$$

where $\sqrt{\mathbf{D}} = \operatorname{diag}(\sqrt{d_1}, \ldots, \sqrt{d_T})$, and

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

3.2 KAARCh

The main idea behind our new method, the Kernel Aggregating Algorithm for Regression with Changing dependencies (KAARCh), is to apply the Aggregating Algorithm (AA) to the case where the pool of experts is made up of all linear predictors that can change with time. More formally, an expert in this case is a sequence $\theta_1, \theta_2, \ldots$, that at time T predicts $\mathbf{x}'_T(\theta_1 + \theta_2 + \ldots + \theta_T)$, where $\mathbf{x}_T \in \mathbb{R}^n$ and for every t, $\theta_t \in \mathbb{R}^n$.

Due to space limitations we are only going to give an overview of the main theoretical results achieved (for details see [11]). Let $a_1, \ldots, a_T$ be positive constants. Applying the AA to the pool of experts described above with a Gaussian prior and introducing kernels, we get KAARCh which makes a prediction by

$$\gamma_T = \widetilde{\mathbf{y}}'\left(\bar{\mathbf{K}} + \mathbf{I}\right)^{-1}\bar{\mathbf{k}} \ , \tag{4}$$

where $\bar{\mathbf{K}} = \left(\left(\sum_{t=1}^{\min(i,j)} \frac{1}{a_t}\right) k(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j}$, and $\bar{\mathbf{k}} = \left(\left(\sum_{t=1}^{i} \frac{1}{a_t}\right) k(\mathbf{x}_i, \mathbf{x}_T)\right)_i$, for $i, j = 1, \ldots, T$.

The main property of KAARCh is that its cumulative loss is less or equal to that of a wide class of experts plus a term of the order $o(T)$. Informally, this class is comprised of all the predictors that do not change very rapidly with time. We assume that outcomes are bounded by Y, therefore, for any t, $y_t \in [-Y, Y]$ (however, we do not require our algorithm to know Y).

Theorem 1. *Let k be a kernel on a space X, let D_t be decision rules in the RKHS induced by k and let $D = (D_1, D_2, \ldots, D_T)'$. Then for any point in time T and any $a_t > 0$, $t = 1, \ldots, T$,*

$$\mathrm{L}_T(\mathrm{KAARCh}) \leq \inf_D \left(\mathrm{L}_T(D) + \sum_{t=1}^{T} a_t \|D_t\|^2 \right) + Y^2 \ln \det \left(\bar{\mathbf{K}} + \mathbf{I} \right) \ .$$

Let us bound the norm of D_1 by d and assume that T is known in advance. If each $\|D_t\|$, for $t = 2, \ldots, T$, is small, we can find $a_1, \ldots, a_T$ such that the extra terms become of the order $o(T)$.

Corollary 1. *Under the conditions of Theorem 1, let T be known in advance. For every positive d and ε, if $\|D_1\| \leq d$ and, for $t = 2, \ldots, T$, $\|D_t\| \leq \frac{d}{T^{0.5+\varepsilon}}$, we can choose $a_1, \ldots, a_T$ such that*

$$\mathrm{L}_T(\mathrm{KAARCh}) \leq \mathrm{L}_T(D) + O\left(T^{\max(0.5,(1-\varepsilon))}\right) = \mathrm{L}_T(D) + o(T) \ .$$

This result can also be achieved if we assume that there are only a few nonzero D_t, for $t = 2, \ldots, T$. In this case, the nonzero D_t can have greater flexibility.

Implementation Notes. For simplicity, we may take all equal $a_1, a_2, \ldots, a_T = a$. In this case, (4) becomes

$$\gamma_T = \widetilde{\mathbf{y}}' \left(\breve{\mathbf{K}} + a\mathbf{I} \right)^{-1} \breve{\mathbf{k}} \ , \tag{5}$$

where

$$\breve{\mathbf{K}} = \begin{bmatrix} 1k(\mathbf{x}_1, \mathbf{x}_1) & 1k(\mathbf{x}_1, \mathbf{x}_2) & 1k(\mathbf{x}_1, \mathbf{x}_3) & \cdots & 1k(\mathbf{x}_1, \mathbf{x}_T) \\ 1k(\mathbf{x}_2, \mathbf{x}_1) & 2k(\mathbf{x}_2, \mathbf{x}_2) & 2k(\mathbf{x}_2, \mathbf{x}_3) & \cdots & 2k(\mathbf{x}_2, \mathbf{x}_T) \\ 1k(\mathbf{x}_3, \mathbf{x}_1) & 2k(\mathbf{x}_3, \mathbf{x}_2) & 3k(\mathbf{x}_3, \mathbf{x}_3) & \cdots & 3k(\mathbf{x}_3, \mathbf{x}_T) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1k(\mathbf{x}_T, \mathbf{x}_1) & 2k(\mathbf{x}_T, \mathbf{x}_2) & 3k(\mathbf{x}_T, \mathbf{x}_3) & \cdots & Tk(\mathbf{x}_T, \mathbf{x}_T) \end{bmatrix}, \breve{\mathbf{k}} = \begin{bmatrix} 1k(\mathbf{x}_1, \mathbf{x}_T) \\ 2k(\mathbf{x}_2, \mathbf{x}_T) \\ 3k(\mathbf{x}_3, \mathbf{x}_T) \\ \vdots \\ Tk(\mathbf{x}_T, \mathbf{x}_T) \end{bmatrix}.$$

Recalling that a scalar multiplied by a kernel is still a kernel, and making allowances such that steps in time can be skipped (for instance there is no data available for some steps), the coefficients $1, \ldots, T$ in $\breve{\mathbf{K}}$ and $\breve{\mathbf{k}}$ can be replaced with any increasing, positive real numbers $t_1, \ldots, t_T$, representing the real-world time at which examples arrive.

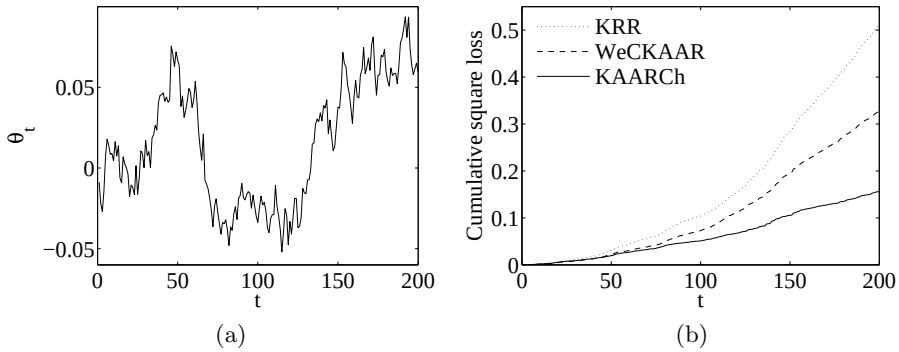

Fig. 1. The behaviour of θ_t with time (a), approximating Brownian motion, and the cumulative loss suffered by KRR, WeCKAAR and KAARCh on the artificial dataset (b).

4 Empirical results

In this section we measure the empirical performance of our methods on an artificial dataset and on a real-world dataset on options implied volatility.

4.1 Artificial Dataset

Let $w_1, \ldots, w_T \in \mathbb{R}$ be T normally distributed random variables with mean 0 and variance σ^2, and $\theta_t = \sum_{i=1}^{t} w_i$. Drawing $\mathbf{x}_t \in \mathbb{R}$ from the interval $[0, 1]$ using a uniform distribution, we generate a dataset by the equation $y_t = \theta_t' \mathbf{x}_t$. In our experiments, we set $T = 200$ and $\sigma = 0.01$, and repeated the procedure 20 times on such randomly generated datasets. The typical behaviour of a resulting θ_t with time can be seen in Fig. 1 (a). In the normal regression setting (where the dependency does not change with time) this graph would simply be a flat line. In Fig. 1 (b) we show the mean over all runs of the cumulative square loss suffered by KRR, WeCKAAR and KAARCh using a linear kernel on these datasets.

4.2 Options Implied Volatility Data

The Russian Trading System Stock Exchange (RTSSE) have provided us with data containing the details of option transactions on several underlying assets. Options are types of derivative securities that give the right to buy or sell assets for a particular strike price in the future (see [1] for more details). The accurate pricing of these options is an important problem. The most popular approach to pricing options is based on the Black-Scholes (B-S) theory. This assumes that the asset price follows an exponential Wiener process with constant volatility σ which cannot be directly observed but can be estimated from historical data. In practice this model is often violated. Given the current prices of options and the underlying asset we can find σ that satisfies the B-S equations. This σ is

Table 1. Results on options implied volatility data. All mean square losses reported are $\times 10^{-3}$, apart from the ones for EERU1206 which are $\times 10^{-2}$.

RTSI1206 (10126 transactions) RTSSE: 2.91				RTSI0307 (8410 transactions) RTSSE: 2.78			
	KRR	WeCKAAR	KAARCh		KRR	WeCKAAR	KAARCh
Poly	36.56	2.19	(**2.16**)	Poly	8.29	2.40	**2.38**
Spline	2.63	(**2.23**)	(2.24)	Spline	3.49	2.29	**2.29**
RBF	3.31	2.33	**2.31**	RBF	3.87	2.33	**2.32**
GAZP1206 (9382 transactions) RTSSE: 1.29				GAZP0307 (10985 transactions) RTSSE: 2.13			
	KRR	WeCKAAR	KAARCh		KRR	WeCKAAR	KAARCh
Poly	1.59	1.54	**1.53**	Poly	3.16	2.45	**2.45**
Spline	5.21	**1.49**	1.49	Spline	2.85	2.47	**2.47**
RBF	1.59	**1.47**	1.48	RBF	3.53	**2.49**	2.49
EERU1206 (13152 transactions) RTSSE: 1.47				EERU0307 (14776 transactions) RTSSE: 4.74			
	KRR	WeCKAAR	KAARCh		KRR	WeCKAAR	KAARCh
Poly	162.43	**1.71**	1.72	Poly	5.49	4.58	**4.52**
Spline	1.92	**1.65**	1.66	Spline	5.07	**4.49**	4.50
RBF	6.36	**1.65**	1.65	RBF	5.83	(**4.46**)	4.49

known as the implied volatility and exhibits a dependence on the strike price and time. There is no generally recognised theory explaining the phenomenon of implied volatility; however, it remains a useful parameter and traders often use it to quote option prices. We are interested in using learning theory methods to predict implied volatility 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 as the signal.

For WeCKAAR's $d_1, \ldots, d_T$ and KAARCh's $t_1, \ldots, t_T$, we used a real number representing the time at which the transactions occurred. The kernels used were the spline, polynomial degree 2, and RBF with $\sigma = 1$ (see, for example, [5]). We employed a sliding window (of size 50) approach. The parameter a (see (1), (3), and (5)) was updated every 50 steps by finding a value that works well on previous examples. Due to computational limitations, we ran experiments on 100 randomly selected segments containing 200 transactions from every dataset.

In Table 1 we give the results obtained on different options data. EERU and GAZP are options on futures of liquid stocks and RTSI is related to options on futures of a popular RTSSE index (the appended numbers specify different transaction periods). The results show the mean square loss suffered by WeCKAAR, KAARCh and KRR, and also that of the proprietary method used at the RTSSE for comparison. To measure the statistical significance of the difference between the results of our methods and that of the RTSSE we used the Wilcoxon signed rank test. When there is no statistical significance in the difference (we use the conventional 5% threshold) the corresponding loss is enclosed in parentheses.

5 Discussion

KAARCh's performance on the artificial dataset is much better than that of WeCKAAR and KRR. We attribute this to KAARCh's superior theoretical properties. Six real-world datasets on options implied volatility were also considered. The results achieved by KAARCh and WeCKAAR on these datasets are always better than those of KRR and very close to those of the RTSSE (and slightly better in half of them). The proprietary method used at the RTSSE was specifically designed for this application and is constantly monitored and tuned by experts to predict better. Therefore, it is remarkable that our methods perform comparably. These results show that our new methods KAARCh and (to a lesser extent) WeCKAAR are capable of handling changing dependencies and, in this context, are an improvement on standard regression techniques.

Acknowledgements. We are grateful to Dr Michael Vyugin at the RTSSE for providing the data and sharing his expertise with us. We also 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. 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
4. Aronszajn, N.: Theory of reproducing kernels. Transactions of the American Mathematical Society **68** (1950) 337–404
5. Schölkopf, B., Smola, A.J.: Learning with Kernels — Support Vector Machines, Regularization, Optimization and Beyond. The MIT Press, USA (2002)
6. Hoerl, A.E.: Application of ridge analysis to regression problems. Chemical Engineering Progress **58** (1962) 54–59
7. 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
8. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press (2006)
9. 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
10. 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
11. Busuttil, S., Kalnishkan, Y.: Online regression competitive with changing predictors. In: Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT 2007). Lecture Notes in Computer Science, Springer (to appear in 2007)