JMLR: Workshop and Conference Proceedings 1–16

Merging Time Series with Specialist Experts

Tim Scarfe TIM@DEVELOPER-X.COM
Yuri Kalnishkan YURA@CS.RHUL.AC.UK
Computer Learning Research Centre and Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, United Kingdom

Abstract

The paper describes an application of specialist experts techniques to prediction with side information. We pick vicinities in the side information domain to create elementary time series, use standard prediction techniques to predict for those elementary series, and then merge the predictions using specialist experts methods. Prediction with expert advice bounds ensure that optimal vicinities are selected dynamically. The algorithm is tested on the problem of predicting implied volatility of options and proves to be a viable alternative to on-line regression.

Keywords: prediction with expert advice, square loss, implied volatility

1. Introduction

We consider the on-line protocol where on each trial $t = 1, 2, \ldots$ the learner observes a signal x_t and attempts to predict an outcome y_t, which is shown to the learner later. The performance of the learner is measured by means of the cumulative loss; throughout the most of the paper we use square loss.

This problem falls in a somewhat grey area between the theory of time series and machine learning. Machine learning mostly studies 'spatial' dependencies between x_t and y_t while the theory of time series concentrates on the evolution of y_t with time.

This problem has been addressed within the context of on-line regression. One option is to apply traditional batch algorithms such as ridge regression in the on-line context; see (Zhdanov and Kalnishkan, 2013) for a comparison of losses of ridge regression in the on-line and batch modes. An alternative is to develop special on-line regression algorithms. For example, see (Vovk, 2001), (Azoury and Warmuth, 2001), and Section 11.8 of (Cesa-Bianchi and Lugosi, 2006) for aggregating algorithm regression also known as the Vovk-Azoury-Warmuth predictor. An important area is the development of regression algorithms targeted at changing dependencies; see (Herbster and Warmuth, 2001) for tracking algorithms, (Busuttil and Kalnishkan, 2007a) for aggregating algorithm regression with changing dependencies, and (Chernov and Zhdanov, 2010b) for regression under discounted loss.

In this paper we propose an alternative to regression methods. We develop an approach to on-line prediction combining spatial dependencies with temporal evolution based on the use of specialist experts. Signals x are grouped into a finite number of vicinities and time series of outcomes y_t are formed according to what vicinities x_t belong to. The predictions made for time series are then merged using a prediction with expert advice technique. This

paper uses recent specialist expert techniques developed in (Chernov et al., 2010; Chernov and Vovk, 2009) to handle the vicinities with greater flexibility.

The method is tested on the problem of predicting applied volatility of options in datasets provided by the Russian Trading Systems Stock Exchange. Very simple methods of predicting time series are used, namely, predict the last element and exponentially weighted moving average. Still the results turn out to be comparable to a much more sophisticated kernel ridge regression technique. The experiments in this paper may be seen as a follow-up to (Busuttil and Kalnishkan, 2007b), where the same datasets were used.

The paper is organised as follows. Section 2 describes the prediction framework and some necessary algorithms from prediction with expert advice. Section 3 describes the main algorithm, *merging*, and its naive alternative, *partitioning*. Section 4 discusses the problem of volatility prediction in general and the datasets we use in particular. Section 5 describes an application of the merging algorithm to our datasets and reports the results. Finally, in Section 6 the results as well as some properties of the algorithm and the domain are discussed.

As an empirical study of an application of the theory of prediction with expert advice, this paper can be compared to (Vovk and Zhdanov, 2009), which also deals with the square loss function. Such applications are relatively few and may serve as testing grounds for method of prediction with expert advice in the future.

2. Preliminaries

2.1. Prediction Framework

This paper is concerned with prediction in the following framework. Let outcomes $\omega_1, \omega_2, \ldots$ from an *outcome set* Ω occur successively in discrete time. A *learner* tries to predict each outcome and outputs a prediction γ_t from a *prediction set* Γ on the basis of a *signal* x_t from a *domain* X each time before it sees the outcome ω_t. The quality of predictions is assessed by means of a *loss function* $\lambda : \Gamma \times \Omega \to [0, +\infty]$.

The framework can be summarised in the following protocol:

Protocol 1: Basic prediction

for $t = 1, 2, \ldots$ **do**
  nature announces $x_t \in X$
  learner outputs $\gamma_t \in \Gamma$
  nature announces $\omega_t \in \Omega$
  learner suffers loss $\lambda(\gamma_t, \omega_t)$
end

Over T trials the learner suffers the cumulative loss

$$\mathrm{Loss}_T = \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) \ .$$

We will denote the loss of the learner by Loss_T or by $\mathrm{Loss}_T(\mathrm{Algorithm})$ with Algorithm in brackets being the name of the algorithm the learner uses.

In this paper we are mostly interested in the case where the prediction and outcome spaces are an interval $\Omega = \Gamma = [A, B]$ and the square loss function $\lambda(\gamma, \omega) = (\gamma - \omega)^2$ is used.

2.2. Prediction with Expert Advice

This section discusses prediction with expert advice; see (Cesa-Bianchi and Lugosi, 2006) for a complete overview. The problem of prediction with expert advice can be summarised as follows.

Suppose that there is a pool Θ of *(static) experts*; throughout this paper we assume that Θ is finite. The experts try to predict the outcomes from the same sequence and their predictions $\gamma_t(\theta)$ are made available to the learner before it outputs its own.

The goal of the learner is to suffer total loss comparable to the best expert in the pool in some sense.

Protocol 2: Prediction with expert advice

for $t = 1, 2, \ldots$ **do**
  experts $\theta \in \Theta$ announce predictions $\gamma_t(\theta) \in \Gamma$
  learner outputs $\gamma_t \in \Gamma$
  nature announces $\omega_t \in \Omega$
  each expert $\theta \in \Theta$ suffers loss $\lambda(\gamma_t(\theta), \omega_t)$
  learner suffers loss $\lambda(\gamma_t, \omega_t)$
end

Over T trials each expert θ suffers the cumulative loss

$$\mathrm{Loss}_T(\theta) = \sum_{t=1}^{T} \lambda(\gamma_t(\theta), \omega_t)$$

and the learner suffers the cumulative loss

$$\mathrm{Loss}_T = \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) \ ;$$

one wants the inequality $\mathrm{Loss}_T \lesssim \mathrm{Loss}_T(\theta)$ to hold for all $T = 1, 2, \ldots$ and $\theta \in \Theta$.

It is in the spirit of prediction with expert advice not to impose any restrictions on the law generating outcomes ω_t or on the internal working of experts. As a matter of fact, 'nature' and 'experts' are just names for the slots in the protocol.

2.3. Aggregating Algorithm

In this section we overview the standard aggregating algorithm (AA) for prediction with expert advice after (Vovk, 1998, 2001). It takes the following parameters: a learning rate $\eta \in (0, +\infty)$ and an initial distribution over the set of static experts θ; a distribution can be represented by an array of initial weights $p_0(\theta), \theta \in \Theta$.

The algorithm maintains an array of weights $w_t(\theta), \theta \in \Theta$. Their initial values are $w_0(\theta) = p_0(\theta), \theta \in \Theta$, and they are updated according to the rule

$$w_t(\theta) = w_{t-1}(\theta) e^{-\eta \lambda(\gamma_t(\theta), \omega_t)} = p_0(\theta) e^{-\eta \operatorname{Loss}_t(\theta)} \ .$$

On step t upon observing the experts' predictions $\gamma_t(\theta)$ the learner outputs a prediction γ_t that for every possible $\omega \in \Omega$ satisfies the condition

$$\lambda(\gamma_t, \omega) \le c(\eta) g(\omega) \ , \tag{1}$$

where

$$g(\omega) = -\frac{1}{\eta} \ln \frac{1}{\sum_{\theta \in \Theta} w(\theta)} \sum_{\theta \in \Theta} w(\theta) e^{-\eta \lambda(\gamma_t(\theta), \omega)} \tag{2}$$

and $c(\eta)$ is a constant specified by the loss function λ. The constant is defined in such a way that γ_t can always be found. One can show by induction (see Lemma 1 from (Vovk, 2001)) that

$$\sum_{t=1}^{T} g(\omega_t) = -\frac{1}{\eta} \ln \sum_{\theta \in \Theta} p_0(\theta) e^{-\eta \operatorname{Loss}_T(\theta)}$$

and therefore

$$\operatorname{Loss}_T(AA) = \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) \le c(\eta) \operatorname{Loss}_T(\theta) + \frac{c(\eta)}{\eta} \ln 1/p_0(\theta) \ . \tag{3}$$

The aggregating algorithm thus performs nearly as well as the best expert losswise.

If the prediction and outcome spaces are an interval $\Omega = \Gamma = [A, B]$ and the square loss function is $\lambda(\gamma, \omega) = (\gamma - \omega)^2$, then for η satisfying $0 < \eta \le \frac{2}{(B-A)^2}$ we have $c(\eta) = 1$ (see (Vovk, 2001; Chernov and Zhdanov, 2010a)) and therefore the optimal value is $\eta = \frac{2}{(B-A)^2}$. For this values of η we can use a simple *substitution function*

$$\gamma = \frac{A+B}{2} - \frac{g(B) - g(A)}{2(B-A)}$$

mapping g to a γ satisfying (1).

The aggregating algorithm generalises Bayesian mixtures of probabilistic hypothesis (see, e.g., Section 2 of (Bishop and Nasrabadi, 2006)); it is identical to the Bayesian mixture for the so called logarithmic loss. However it is more general in that it admits arbitrary loss functions such as the square loss function.

2.4. Specialist Experts

Suppose that an expert in the prediction with expert advice framework can abstain from making a prediction on step t; if it does so, we say that it sleeps on step t. One may want to obtain an equivalent of bound (3) ensuring that the learner competes well with every expert θ on the steps where θ is awake.

The concept of a specialist expert was proposed in (Freund et al., 1997). In this paper we will be discussing specialist experts as a special case of the theory of expert evaluators

after (Chernov et al., 2010; Chernov and Vovk, 2009). A simple extension of the AA may be used for specialist experts.

If an expert θ sleeps on step t, let us assume that it suffers notional loss $\lambda(\gamma_t(\theta), \omega_t)/c(\eta)$ (if $c(\eta) = 1$ we can simply say that it 'goes with the crowd' and subscribes to yet unknown γ_t whatever it is going to be) and apply the AA. The weight of a sleeping expert is updated according to this notional loss. If Θ_{sleep} is the set of experts sleeping on step t and Θ_{awake} is the set of experts that are awake (not sleeping) on step t, then (1) becomes

$$\lambda(\gamma_t, \omega) \le -\frac{c(\eta)}{\eta} \ln \frac{1}{\sum_{\theta\in\Theta} w_{t-1}(\theta)} \left(\sum_{\theta\in\Theta_{\text{awake}}} w_{t-1}(\theta) e^{-\eta\lambda(\gamma_t(\theta),\omega)} + \sum_{\theta\in\Theta_{\text{sleep}}} w_{t-1}(\theta) e^{-\eta\lambda(\gamma_t,\omega)/c(\eta)} \right)$$

or

$$e^{-\eta\lambda(\gamma_t,\omega)/c(\eta)} \sum_{\theta\in\Theta} w_{t-1}(\theta) \ge \sum_{\theta\in\Theta_{\text{awake}}} w_{t-1}(\theta) e^{-\eta\lambda(\gamma_t(\theta),\omega)} + \sum_{\theta\in\Theta_{\text{sleep}}} w_{t-1}(\theta) e^{-\eta\lambda(\gamma_t,\omega)/c(\eta)} \ .$$

Clearly, all terms corresponding to sleeping experts cancel out and we get

$$\lambda(\gamma_t, \omega) \le \frac{c(\eta)}{\eta} \ln \frac{1}{\sum_{\theta\in\Theta_{\text{awake}}} w_{t-1}(\theta)} \sum_{\theta\in\Theta_{\text{awake}}} w_{t-1}(\theta) e^{-\eta\lambda(\gamma_t(\theta),\omega)} \ ,$$

i.e., the formula is identical to that from the aggregating algorithm except that the sum is taken over the experts that are awake. Note that we still need to update the weights of sleeping experts so that they get the right weights when they wake up. We will call this algorithm aggregating algorithm with sleeping experts (AAS).

Arguing as in the case of the AA, we get a bound similar to (3); by dropping equal terms in the losses on the left- and right-hand side we obtain

$$\text{Loss}_T^{(\theta)}(AAS) \le c(\eta)\, \text{Loss}_T^{(\theta)}(\theta) + \frac{c(\eta)}{\eta} \ln 1/p_0(\theta) \ , \tag{4}$$

where the sum in $\text{Loss}^{(\theta)}$ is taken only over steps when expert θ was not sleeping.

3. Algorithms

In this section we describe an algorithm for a predictor working in the environment of Protocol 1.

Suppose that we want to apply a time series prediction method. The simplest way of doing this is to treat the outcomes $\omega_1, \omega_2, \ldots$ as a time series ignoring the signals x_t altogether. Obviously this can lead to a loss of potentially useful information.

Let us take a more refined approach and consider a partition of the domain $X = \cup_{k=1}^{K} X_k$, where $X_i \cap X_j = \varnothing$ if $i \neq j$. This partition allows one to split the time series into k different series. The predictor then can work as follows. On step t it finds k such that $x_t \in X_k$, then picks the subsequence $t_1 < t_2 < \ldots < t_s < t$ such that $x_{t_i} \in X_k$ and uses the time series $\omega_{t_1}, \omega_{t_2}, \ldots, \omega_{t_s}$ to make a prediction for step t. We will call this the *partitioning method*.

A disadvantage of this method is that it ignores all dependencies among ω_t for different k. The moment t_s when the signal belonged to X_k for the last time may have occurred long ago compared to t and the outcomes may have evolved significantly since then.

This motivates the following algorithm. Consider a finite subset $\{U_1, U_2, \ldots, U_K\} \subseteq 2^X$ such that $\cup_{k=1}^{K} U_k = X$. We will call sets U_k vicinities because it is natural to choose them in such a way that elements of U_i are in some respect akin to each other. Each vicinity U_i generates a specialist expert that predicts as follows. The expert is awake only on steps t where $x_t \in U_i$. It maintains the series $\omega_{t_1}, \omega_{t_2}, \ldots$ of outcomes for such steps and uses the series to make predictions for steps t where $x_t \in U_i$. For t such that $x_t \notin U_i$ the expert makes no predictions. The experts are then merged using the aggregating algorithm for specialist experts. We will call this the *merging method*.

The computational complexity of the merging method depends on the underlying time series prediction algorithm. The extra complexity brought about by merging depends on the employed loss function; however, for most natural loss functions including the square loss merging takes time linear in the number of vicinities.

4. RTSSE Implied Volatility Prediction

In this section we describe a model problem in detail.

4.1. Implied Volatility

An option is a derivative financial instrument linked to an underlying asset, which is usually a share, but can also be a portfolio of shares, a futures on a share etc. There are two popular types of options, puts and calls. Definitions and more details on puts and calls are available from standard textbooks such as (Hull, 2006; Wilmott, 2007); for the purposes of this paper it suffices to note that puts and calls have two important parameters, strike price X and time to maturity T. Throughout the life of a particular option X stays fixed while T decreases and when it reaches 0, the option ceases to exist.

The most popular approach to options pricing is based on the Black-Scholes(-Merton) theory. This theory assumes that the underlying asset price S follows an exponential Wiener process with constant volatility σ, which cannot be directly observed but can be estimated from historical data. Given σ, the prices of so called *European* call and put options, can be calculated using the Black-Scholes formulas

$$c = SN(d_1) - Xe^{-rT}N(d_2) \ , \tag{5}$$

$$p = Xe^{-rT}N(-d_2) - SN(-d_1) \tag{6}$$

with

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \ ,$$
$$d_2 = \frac{\ln(S/X) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} \ ,$$

where N is the cumulative distribution function of the Gaussian distribution with the mean of 0 and standard deviation of 1, S is the price of the underlying asset, σ is its volatility, r is the risk-free interest rate (assumed to be 0 in this paper) and X and T are the strike price and time left until the maturity of the option.

In practice this model is often violated. The prices c and p can be observed directly in transactions as well as S. Given the current prices of options and the underlying asset we can find σ that satisfies the Black-Scholes equations. This σ is known as the implied volatility. Contrary to Black-Scholes theory, it is often not constant and exhibits a dependency on the strike price and time. The graph of a dependency of σ on the strike price X (other parameters being equal) is known as the volatility smile due to its characteristic shape; the graph of the dependency of σ on X and T is known as the volatility surface. There is no unique generally accepted theory explaining the phenomenon of implied volatility; however, volatility remains a meaningful parameter and traders often use it to quote option prices. The reader may refer to Chapter 16 of (Hull, 2006) as a financial introduction to volatility smiles.

4.2. RTSSE Datasets

In this paper we approach the problem of finding the implied volatility from a purely machine learning perspective. The datasets we use were provided by the Russian Trading System Stock Exchange (RTSSE) and record data from mid-2000s, when the Russian stock market was experiencing steady unperturbed growth (perhaps unhealthy from the economics point of view).

The following three datasets were used: eeru1206 describes put and call options maturing in December 2006 on futures on shares of Unified Energy System of Russia (the company has since been split and its shares are no longer trading); gaz307 describes put and call options maturing in March 2007 on futures on shares of Gazprom; and rts307 describes put and call options maturing in March 2007 on futures on the RTSSE index.

The options studied were *American* rather than European, which implies a difference in the execution arrangements. Under very general assumptions, American call options should not be executed early and therefore they cost the same as European call options and should satisfy (5). The same cannot be said of American put options, which can cost more than European put options. Formula (6) is technically speaking not applicable to American put options. However one can still calculate implied volatility using the Black-Scholes formula; it has no meaning within the standard Black-Scholes model, but is was used by RTSSE as a helpful descriptive parameter.

Each dataset contains records of consecutive transactions with put and call options on a particular underlying asset with the same maturity date. The numbers of transactions are given in Table 1. The following attributes of a transaction are available: date and

Table 1: Datasets Summary

Dataset	Underlying asset	Maturity	Number of transactions
eeru1206	futures on share	December 2006	13152
gaz307	futures on share	March 2007	10985
rts307	futures on index	March 2007	8410

time, strike price X, time left to maturity of the option T, the price of the underlying asset S at the time of transaction, and a bit differentiating puts from calls. We attempt to predict the implied volatility in the transaction. The quality of the prediction is measured by the squared deviation of the predicted implied volatility from the true implied volatility calculated from the option price using the Black-Scholes formula (note that the option price is not one of our attributes; otherwise the problem would amount to learning the Black-Scholes formula itself).

We compare the losses incurred by our methods against a benchmark technique employed by the stock exchange for quoting implied volatilities. This is a proprietary method based on fitting coefficients in a formula describing the volatility smile; the method involves occasional manual adjustments. As the outputs of the proprietary technique were used for publicly available quotes, one cannot rule out the influence of the technique on the behaviour of the implied volatility, which makes competing with this method particularly difficult.

4.3. Preliminary Analysis

The behaviour of volatility is illustrated in Figures 1–4, which show dependencies of volatility on strike and transaction number (which is essentially effective time) near the beginning and near the end of eeru1206. Initially volatility is relatively flat. Towards the end of the dataset it becomes much more variable and the dependency of volatility on strike takes on a smile-like shape. The range of the plots was capped at 1, so occasional outliers exceeding 1 are not shown.

The pictures show that on the one hand, the sequence of volatilities looks very much like a time series and on the other hand volatility exhibits a dependency on strike price, especially towards the end of the dataset (i.e., close to maturity of the option).

The number of possible strikes is limited. While theoretically the strike can have any real value, stock exchanges usually restrict strikes to some round numbers in order to improve liquidity. Thus one can consider splitting the time series into separate time series, one for each strike. However it is easy to see from the pictures that some strikes are much more common than others. While for popular strikes the most recent transaction may be quite near, for rare strikes it is far away.

Note that we cannot straightforwardly consider this problem within the framework of multivariate time series as, e.g., in (Lutkepöhl, 2005): we cannot speak of an evolution of a vector of volatilities for different strikes because of an irregular nature of the sequence of strikes. As a matter of fact, most standard models for volatility smiles are based on vectors of closing prices as in, e.g., (Konstantinidi et al., 2008).

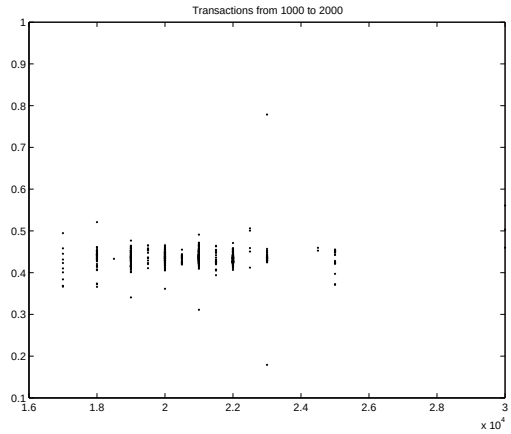

Figure 1: Volatility vs strike, transactions 1000-2000

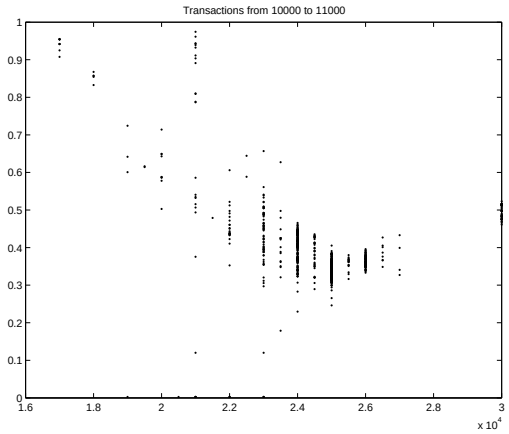

Figure 2: Volatility vs strike, transactions 10000-11000

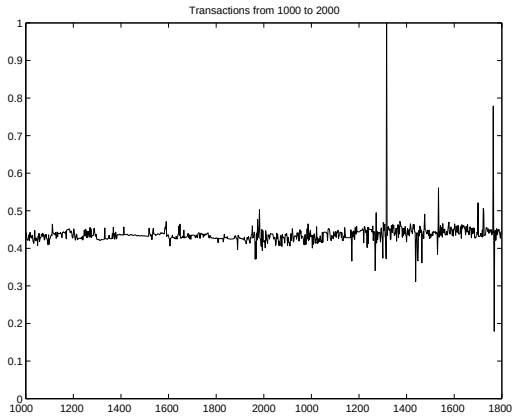

Figure 3: Volatility vs number, transactions 1000-2000

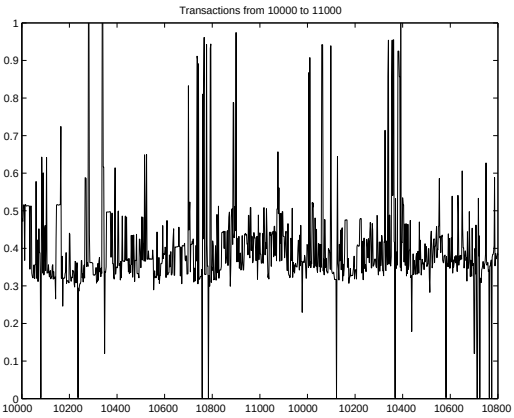

Figure 4: Volatility vs number, transactions 10000-11000

One thus is naturally drawn to the idea of grouping strikes together and forming combined time series for groups of strikes. However the pictures do not seem to suggest a natural grouping. It is known that most transactions happen for strikes near the current underlying price and the strikes on the sides are less frequently used. While the pictures seem to be in agreement with this generally, there is no simple pattern in the distribution of strikes.

Here the ideas of the paper come naturally. Let us create multiple vicinities of strikes, introduce associated specialist experts, and let the AAS converge on the relevant ones.

5. Applying the Algorithm

5.1. Setup

In this section we describe an application of the algorithm of the paper to the problem.

Let S consisting of $s_1 < s_2 < \ldots < s_L$ be the strikes for a dataset. A simple vicinity of diameter d is a set of d consecutive strikes; there are $L - d + 1$ vicinities of diameter d.

A compound vicinity is a subset of $S \times \{0, 1\}$, where the 0/1 bit denotes whether the option in transaction is a put or call. A compound vicinity of diameter d is a product of a vicinity of diameter d by either 0 or 1; there are $2(L - d + 1)$ compound vicinities of diameter d. Note that some of them may give rise to empty time series if, say, there were no transactions on put options with particular strikes. However every transaction belongs to at least one compound vicinity.

In the experiments below we took all simple and compound vicinities of diameters from 1 to d with $d = 5$.

We are interested in the performance w.r.t. the squared loss so we apply the AAS for square loss. In order to apply AAS, predictions of time series methods were capped at 1 and thus for the purpose of merging we assumed that the prediction and outcome space were $[0, 1]$.[1] Equal initial weights were given to all experts.

Two very simple methods for predicting time series turned out to be very successful, predict the last element and exponentially weighted moving average (EWMA). In the former given a series $y_1, y_2, \ldots, y_T$, the value $\gamma_{T+1} = y_T$ is predicted. In the later the value $\gamma_{T+1} = \lambda y_T + (1 - \lambda)\gamma_T$ is predicted, where γ_T is the prediction output on the previous step and λ is a parameter (taken to be 0.95). At the beginning where no previous element is available, we used the default value of 0.3.

5.2. Experimental Results

The results of the prediction methods were evaluated against the loss of the proprietary RTSSE technique discussed in Section 4.2 called the competitor loss for brevity. In the pictures below we plot the *adjusted loss* of algorithms, which is the cumulative loss minus the cumulative loss of the competitor. Note that for the purpose of calculating the loss, the outcomes were not capped at 1.

Figures 5–10 show the adjusted loss of predict the last element and EWMA in two modes, partitioning and merging. The solid lines represent the results of merging experts based on all simple and compound vicinities of diameters from 1 to 5. The dotted lines represent the naive partitioning mode with compound vicinities of diameter 1. No merging happens there: a signal x_t uniquely determines a vicinity.

Table 2 shows the figures of cumulative adjusted loss at the end of the datasets.

The figures and the table lead to the following conclusions.

1. Simple time series methods applied strike-wise perform comparably to the RTSSE proprietary technique. At the end of the datasets (i.e., when the options approach maturity) they outperform the proprietary technique.

1. Tighter limits would allow one to choose a better learning rate η; however we did not investigate this further.

Table 2: Cumulative loss at the end of the dataset

Dataset	Competitor loss	Improvement achieved by			
		predict last element		EWMA	
		naive partitioning	merging	naive partitioning	merging
eeru1206	185.05	13.91	25.25	15.13	25.73
gaz307	48.56	2.71	5.3	3.75	6.19
rts307	156.33	0.65	5.31	0.76	5.56

The figures show that the competitor outperforms our methods at the beginning. A plausible explanation is that the competitor incorporates some prior knowledge about the behaviour of volatility, while our methods need to learn from scratch. It is tempting to attribute our good performance at the end to learning the data well. However there is a more realistic explanation. The period when we outperform the competitor is rather short; it is even shorter in terms of 'physical' time (transactions happen much more frequently near the end). Presumably the competitor was optimised to perform well *most of the time*.

2. EWMA with $\lambda = 0.95$ performs slightly but consistently better than predict the last element.

3. Merging method with compound vicinities of diameters from 1 to 5 performs consistently better than naive partitioning. It is better most of the time with the gap widening with time.

It should be noted here that the methods perform very fast. Neither the underlying methods nor AAS take much time. On a typical pc the running time goes from seconds to single minutes as d is increased from 1 to 15.

6. Discussion of Experimental Results

6.1. Comparison with Ridge Regression

In this section we report results of kernel ridge regression for comparison. Ridge regression is a popular method of machine learning and it can be seen as an extension of model building used in econometrics.

As input, ridge regression was given all parameters of the transaction enumerated in Section 4.2: strike price, put/call bit, the price of the underlying asset, and the time left to maturity; each parameter was normalised to fit the interval $[-1, 1]$. Vapnik's polynomial kernel of degree d, i.e., $\mathcal{K}(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$ was used. Regression was applied in a sliding window fashion with the window size of 250. On step t regression was trained on the training set formed by 250 previous pairs $(x_{t-251}, y_{t-251}), (x_{t-250}, y_{t-250}), \ldots, (x_{t-1}, y_{t-1})$; regression was thus retrained on every step. For the first 250 transactions prediction was done using the volatility from the previous transaction (recall from Figure 3 that at the beginning of the dataset volatility is very stable).

Table 3 shows the figures of adjusted loss at the end of the dataset for the retrospectively optimal parameters, degree d and ridge a. Note that retrospective choice of optimal parameters skews the comparison; the variability of the cumulative loss for different values of the parameters was substantial.

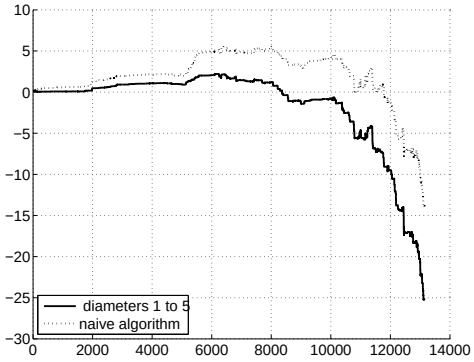

Figure 5: Predict last element on eeru1206

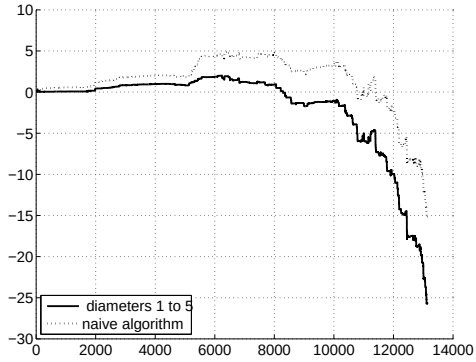

Figure 6: EWMA on eeru1206

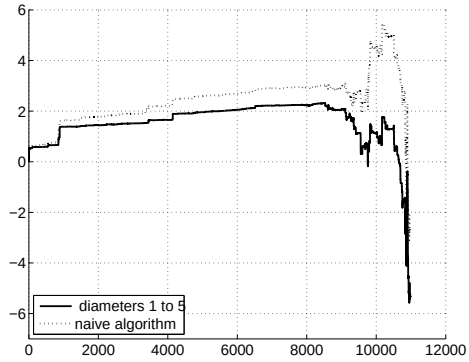

Figure 7: Predict last element on gaz307

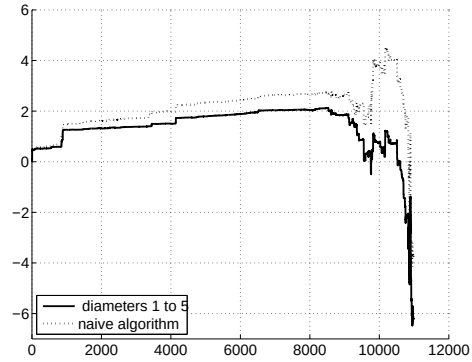

Figure 8: EWMA on gaz307

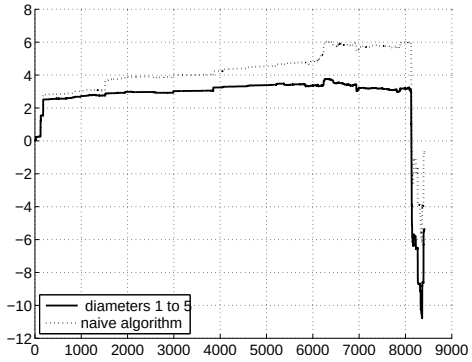

Figure 9: Predict last element on rts307

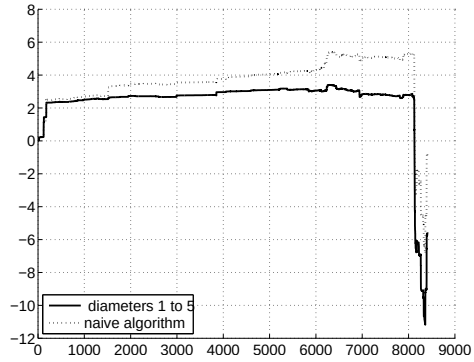

Figure 10: EWMA on rts307

Figures 11–13 show the cumulative adjusted loss of regression for comparison with our methods. The pictures were plotted to the same scales as those in Figures 5–10; for Figure 13 this required cutting off parts of the graph.

Table 3: Cumulative ridge regression loss at the end of the dataset

Dataset	Improvement achieved by RR	Parameters	
		d	a
eeru1206	26.82	4	1
gaz307	4.09	4	0.4
rts307	94.80	5	0.2

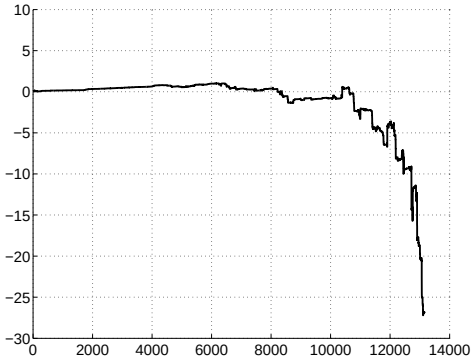

Figure 11: Ridge regression on eeru1206

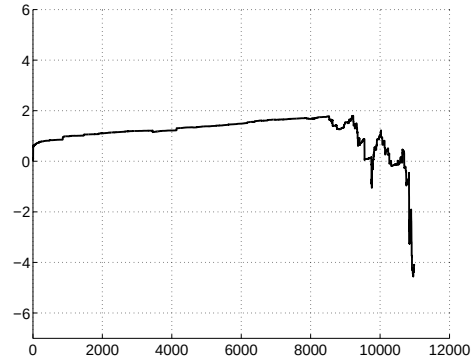

Figure 12: Ridge regression on gaz307

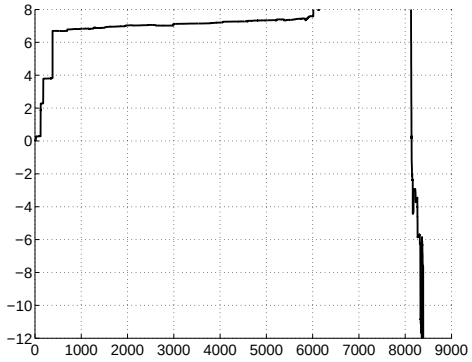

Figure 13: Ridge regression on rts307

For the first two datasets the performance of regression is comparable to our method; our methods outperforms ridge regression on gaz307 in terms of the cumulative loss at the end of the dataset. For the third dataset ridge regression greatly outperforms our method, but it is due to a very short interval at the end.

It is interesting that ridge regression uses much more information that our method. The merging algorithm in our setup makes predictions only on the basis of strike and put/call bit. This hints at an interesting property of the domain. The property requires further investigation.

Table 4: Improvements of EWMA for different ranges of vicinities

Maximum size	eeru1206	gaz307	rts307
1	19.9	4.42	1.73
2	23.04	6.23	3.98
3	24.64	6.3	4.93
4	25.3	6.23	5.37
5	25.73	6.19	5.56
6	25.92	6.09	5.61
7	26.03	6.03	5.61
8	26.11	5.96	5.58
9	26.15	5.86	5.56
10	26.17	5.79	5.54
11	26.19	5.71	5.51
12	26.19	5.66	5.49
13	26.18	5.62	5.46
14	26.17	5.6	5.44
15	26.15	5.58	5.43

6.2. Selecting the Range of Sizes

In our experiments above we took all compound vicinities of sizes from 1 to 5. Let us discuss the choice of the maximum size. Table 4 gives improvements over the competitor at the end of datasets for EWMA. Each line shows the result for the merging algorithm with simple and compound vicinities of diameters from 1 to the maximum size in the first column.

Number 5 is not necessarily the best maximum size, but the improvement brought about by larges sizes is very small. This comes at a cost of increased running time. For very large sizes the performance goes gown, but very slowly.

This behaviour is easy to explain theoretically. The regret term in (4) is proportional to $\ln(1/p_0(\theta))$. Since we took a uniform initial distribution on experts, the regret is proportional to the logarithm of the number of experts.

The number of simple vicinities of diameter d is $L - d + 1$, where L is the number of strikes, so the total number of vicinities grows approximately linearly with d. The regret term thus grows approximately logarithmically. This growth is slow.

The conclusion is that loss-wise one should not be afraid to take large maximum diameter. In practice the main constrain restricting the growth of d is running time.

It is remarkable that vicinities of size *exactly* d perform poorly. For example, the merging algorithm with EWMA running on simple and compound vicinities of diameter exactly 5 brings improvement 18.91 for eeru1206, 3.82 for gaz 307, and 2.69 for rts307. These results are not very good and worse than line 4 in Table 4; however adding vicinities of diameter 5 to vicinities of diameters 1 to 4 helps.

The conclusion is that in the mixture there should be vicinities of different sizes. AAS automatically picks up the right ones.

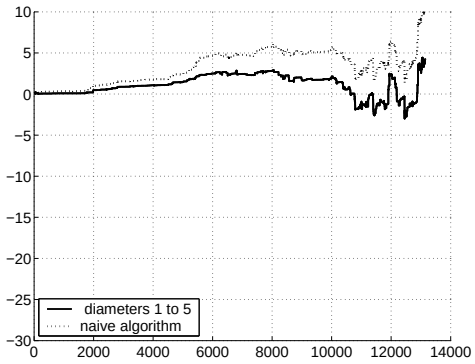

Figure 14: EWMA on eeru1206 with simple vicinities

6.3. Selecting Types of Vicinities

In our experiments above we took both simple and compound vicinities. What happens if we drop compound vicinities?

Figure 14 shows the adjusted loss of EWMA in two modes, partitioning and merging, on eeru1206. The naive partitioning algorithm works on simple vicinities of diameter 1 and the merging algorithm on vicinities of diameters 1 to 5.

The performance is dramatically inferior to that on compound vicinities. Note however that the merging algorithm still outperforms the naive.

It appears that the put/call bit makes a lot of difference to prediction. This indicates an interesting domain property. Theoretically for European options implied volatility should be the same for puts and calls with equal parameters; this follows from the put-call parity, which is a very general statement. The difference may be due to the fact that the options are actually American. Further investigation of this is beyond the scope of the paper.

7. Conclusions

We have proposed a computationally simple algorithm for on-line prediction based on the use of specialist experts technique. Theempirical results show that the algorithm consistently outperforms a naive analog, outperforms a proprietary industrial algorithm, and performs comparably to ridge regression, sometimes outperforming it.

References

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43:211–246, 2001.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

S. Busuttil and Y. Kalnishkan. Online regression competitive with changing predictors. In *Algorithmic Learning Theory, 18th International Conference, Proceedings*, pages 181–195, 2007a.

Steven Busuttil and Yuri Kalnishkan. Weighted kernel regression for predicting changing dependencies. In *Machine Learning: ECML 2007*, pages 535–542. Springer, 2007b.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

A. Chernov and V. Vovk. Prediction with expert evaluators' advice. In *Algorithmic Learning Theory, ALT 2009, Proceedings*, volume 5809 of *LNCS*, pages 8–22. Springer, 2009.

A. Chernov and F. Zhdanov. Prediction with expert advice under discounted loss. In *Proceedings of ALT2010*, volume 6331 of *LNAI*, pages 255–269, 2010a.

A. Chernov, Y. Kalnishkan, F. Zhdanov, and V. Vovk. Supermartingales in prediction with expert advice. *Theoretical Computer Science*, 411(29-30):2647–2669, 2010.

A. V. Chernov and F. Zhdanov. Prediction with expert advice under discounted loss. In *Proccedings of ALT 2010*, volume LNAI 6331, pages 255–269. Springer, 2010b.

Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Proceedings of STOC'97*, pages 334–343. ACM, 1997.

M. Herbster and M. K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, 6th edition, 2006.

E. Konstantinidi, G. Skiadopoulos, and E. Tzagkaraki. Can the evolution of implied volatility be forecasted? Evidence from European and US implied volatility indices. *Journal of Banking & Finance*, 32(11):2401–2411, 2008.

H. Lutkepöhl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56:153–173, 1998.

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Vladimir Vovk and Fedor Zhdanov. Prediction with expert advice for the brier game. *Journal of Machine Learning Research*, 10:2445–2471, 2009. ISSN 1533-7928.

Paul Wilmott. *Paul Wilmott introduces quantitative finance*. Wiley, 2nd edition, 2007.

F. Zhdanov and Y. Kalnishkan. An identity for kernel ridge regression. *Theoretical Computer Science*, 473:157–178, 2013.