Time Series Forecasting Using Distribution Enhanced Linear Regression

Goce Ristanoski[1,2], Wei Liu[2], and James Bailey[1,2]

[1] NICTA Victoria Laboratory
[2] Department of Computing and Information Systems, The University of Melbourne, Australia
g.ristanoski@student.unimelb.edu.au,
{wei.liu,baileyj}@unimelb.edu.au

Abstract. Amongst the wealth of available machine learning algorithms for forecasting time series, linear regression has remained one of the most important and widely used methods, due to its simplicity and interpretability. A disadvantage, however, is that a linear regression model may often have higher error than models that are produced by more sophisticated techniques. In this paper, we investigate the use of a grouping based quadratic mean loss function for improving the performance of linear regression. In particular, we propose segmenting the input time series into groups and simultaneously optimizing both the average loss of each group and the variance of the loss between groups, over the entire series. This aims to produce a linear model that has low overall error, is less sensitive to distribution changes in the time series and is more robust to outliers. We experimentally investigate the performance of our method and find that it can build models which are different from those produced by standard linear regression, whilst achieving significant reductions in prediction errors.

Keywords: Time series prediction, samples grouping, quadratic mean.

1 Introduction

The forecasting of time series is a well known and significant prediction task. Many methods have been proposed in this area, ranging from the simple to the very sophisticated. An important class of techniques includes methods which learn a model based on optimizing a regularized risk function, such as linear regression, Gaussian processes, neural networks and support vector machines. Common drawbacks in the development of time series forecasting methods are that many of them are applicable to only a specific type of time series, such as medical data or stock market series; they may require certain conditions to be fulfilled; or the improvement in performance is associated with a high increase in complexity.

Linear regression is one of the most popular and widely used techniques for time series prediction, since it is simple, intuitive and produces models with a

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 484–495, 2013.

high degree of interpretability. Its performance is also often surprisingly good compared to more complex methods. Nevertheless, reduction of prediction error is a key concern and it remains attractive to consider techniques for improving the behavior of standard linear regression, particularly for challenging circumstances. Such circumstances include non stationary and noisy time series, where changes in the distribution of the series often occur over time.

Given these issues, this paper investigates a modification of linear regression that takes a different perspective. Given a single univariate time series, instead of optimizing the arithmetic mean of the loss over all training samples, we propose to optimize the quadratic mean of the loss over groups of training samples. In particular, the univariate time series is segmented into a number of groups, where each group contains one or more samples. A linear model is then learned which simultaneously optimizes both the average loss of each group, as well as the variance of the loss across groups. In other words, there are two concurrent optimization objectives. First, the model which is produced should have low overall error rate - this is achieved by ensuring the average loss within each group is small. Second, the groups should not vary too much with respect to the error rate of each individual group - this ensures that there is no single group which can significantly bias the characteristics of the output model. A primary question is how should the univariate time series be segmented into groups ? Our proposal is to segment the samples according to their distribution characteristics. The intuition here is that we would like to learn a model which is not biased towards any single distribution, since we do not know which distribution the future behavior of the time series will most resemble.

We experimentally evaluate the performance of our technique using 20 real stock market datasets, 5 non-financial time series datasets and 5 synthetic datasets. We find that our new method (which we call QMReg) can produce linear models which are quite different from those of standard linear regression, and often have significantly less error, with empirical reductions typically in the range of 10%-30%. Our proposed method is an intuitive technique that ensures more evenly distributed, less volatile error and sensitivity to changes in the distribution of the series.

2 Related Work

Data mining researchers have a range of different types of classification and regression algorithms at their disposal, and many of them have been applied to the time series forecasting problem. Artificial Neural Networks(ANNs) [17], Support Vector Machines [13] and Hidden Markov Models [20] [9] are other methods that have been used with some success for financial time series forecasting, and clustering has been used as an aid in the forecasting process as well.

The practical use of linear regression along with statistical knowledge is embodied in the Autoregressive Integrated Moving Average (ARIMA) models, which are descriptive, intuitive and often perform as well as advanced models. Weighted linear regression and AutoRegressive Conditional Heteroskedasticity

(ARCH) models are more complex modifications of linear regression, which require some additional statistical expertise.

Robust regression is a more complex type of linear regression that assesses the statistical properties of the data samples when learning a model [15], by handling the outliers in the data, and assigning appropriate weights (or losses) to reduce their influence [18]. Incorrect labelling of a sample as an outlier can be a concern when using robust regression. Choosing the most appropriate approach often requires statistical expertise by the user.

Since our method is based on grouping of instances as input to a loss minimization function, a potentially related area of research is regularized multitask learning [16], where the objective is to learn multiple classification tasks simultaneously, rather than independently. However, to our knowledge, research in multitask learning has not used a quadratic mean loss function for the simultaneous optimization of loss across groups, as is done in this paper.

3 Distribution Based Quadratic Mean Convex Optimization

We propose an algorithm that has two phases: (1) detection of distribution change points to segment the time series into groups, and (2) training the regression model using a quadratic mean to minimize both the individual loss of each group and the variance of the loss across groups.

3.1 Time Series Segmentation - Distribution Change Points Detection

Distribution changes in time series variables are from a continuous process, subject to a set of external factors [1] [5]. The task of breaking up the samples of a dataset into segments or groups based on distribution or similarities is not an unfamiliar challenge - simple clustering be effective in many cases. When it comes to time series, as we may prefer to preserve the time element, distribution based methods can also be used. Potential candidate tests for non-parametric change point detection methods include the Wilcoxon rank sum method (WXN) and the kernel change method (KCD) [14]. They can be applied for this task as they are understandable and easy to introduce in the learning process. The Wilcoxon rank sum method (WXN) assess whether two sets of data samples follow the same distribution according to a statistical measure. It is an easy to implement statistical test, and no a-priori knowledge is required (other than specification of an appropriate p-value, commonly set to 0.05).

The WXN paradigm is as follows: for a fixed window of m points, [1, m], we appoint after it another sliding window of the same length, [$m+1$, $2m$]. We move the second window and compare if the samples in both windows follow the same distribution: if that is the case, we continue moving the second window, until the distribution changes. The change point will be at the last sample of the second window (point $2m+p$), p>0, where we detect the group window v for

that group of samples; we move the first window just after that point $[2m+p+1, 3m+p]$, the second window comes after the first one $[3m+p+1, 4m+p]$ and we repeat the process for the rest of the dataset(Figure 1). The choice of the window size m can vary, and we choose it to be the same size as the testing set.

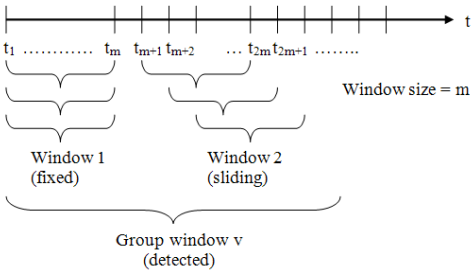

Fig. 1. The Wilcoxon method with a fixed reference window and a sliding window

3.2 Quadratic Mean Based Empirical Loss Function

The quadratic mean is defined as the square root of the average of the squares of each element in a set. In the case of just two errors, ϵ_1 and ϵ_2, the values of the quadratic mean QM and the arithmetic mean AM can be written as:

$$AM = \frac{\epsilon_1 + \epsilon_2}{2}, QM = \sqrt{\frac{\epsilon_1^2 + \epsilon_2^2}{2}} = \sqrt{AM^2 + (\frac{\epsilon_1 - \epsilon_2}{2})^2} \quad (1)$$

This shows that the quadratic mean is lower bounded by the arithmetic mean, and this bound is reached when ϵ_1=ϵ_2. This form of optimization was successfully tested for the scenario of imbalanced relational datasets [7] where there are only two groups (positive and negative classes). The more advanced form we use in our methods is specialised for the case of time series and permits any number of groups.

Many machine learning methods address the learning process as finding the minimum of the regularized risk function. For n training samples ($\mathbf{x}_i$, y_i) (i=1,..,n), where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the i–th training sample, d is the number of features, and $y_i \in \{-1, 1\}$ is the true label for the i–th training sample (in the case of classification) and $y_i \in \mathbb{R}$ in the case of regression, the regularized risk function is:

$$\mathbf{w}^* = argmin_{\mathbf{w}} \ \lambda \mathbf{w}^T \mathbf{w} + R_{emp}(\mathbf{w}) \quad (2)$$

$\mathbf{w}$ is the weight vector which also includes the bias term b (which makes $\mathbf{x}$ having and additional bias feature $x_{d+1} \equiv 1$), and λ is a positive parameter that balances the two items in Equation 2, and $R_{emp}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w})$. The loss function $l(\mathbf{x}_i, y_i, \mathbf{w})$ in $R_{emp}(\mathbf{w})$ measures the distance between a true label y_i and the predicted label from the forecasting done using $\mathbf{w}$, and in the case of Linear Regression it has the form of $\frac{1}{2}(\mathbf{w}^T \mathbf{x}\text{-}y)^2$.

We investigate the use of the quadratic mean as a risk function which balances k values, instead of just two values. Each value represents the average loss for a group of instances. The grouping of instances can be conducted in a range of ways. A simple way is to segment the time series data into k groups by their distribution, and we use the Wilcoxon method for that purpose. Each group consists of consecutive data samples, and may vary in size depending on the distribution of the underlying data. The effect of using the quadratic mean is to produce a model which optimizes the average loss for each group, as well as the variance of the average loss across the k groups.

The details behind the groups error optimization are as follows: with n samples, we denote the sizes of k groups as n_1, n_2,..,n_k, where n_1+n_2+..+n_k=n. The empirical loss function of k group has the form of:

$$R^{QM}_{emp,k}(\mathbf{w}) = \sqrt{\frac{\sum_{j=1}^{k} f_j(\mathbf{w})^2}{k}} \tag{3}$$

where f_j is the average error for group j consisting of n_j consecutive samples following the same distribution

$$f_j(\mathbf{w}) = \frac{\sum_{i=1}^{n_j} l(x_{ji}, y_{ji}, \mathbf{w})}{n_j} \tag{4}$$

where x_{ji} is the i-th sample of group j, and y_{ji} is the i-th output of group j. After some manipulation, it can be rewritten as

$$R^{QM}_{emp,k}(\mathbf{w}) = \sqrt{\mu^2 + \sigma^2} \tag{5}$$

where μ is the mean error of the k groups $\frac{1}{k}\sum_{j=1}^{j=k} f_j(\mathbf{w})$ and σ is the standard deviation of the error across groups $1, \ldots, k$. This form clearly shows the overall loss is the sum of two components, the average loss per group and the variance across groups.

An ideal $\mathbf{w}$ minimizes the square root of the average of squares of errors per group, while keeping the structural risk minimization as well, therefore resulting in the final optimization function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \ \lambda \mathbf{w}^T \mathbf{w} + R^{QM}_{emp}(\mathbf{w}) \tag{6}$$

This form of calculation of the loss function is the form we use for the proposed QMReg model, and this empirical loss function introduces robustness in the algorithm, making it capable of minimizing the effect of outliers and the error per distribution group. By using linear optimization methods, such as the bundle method [6], we can calculate the subradients of the empirical loss and use them to iteratively update the w vector in a direction that minimizes the quadratic mean loss presented at Formula 3. The loss for a given sample is $l(\mathbf{x}_i,y_i,\mathbf{w})=\frac{1}{2}(\mathbf{w}^T\mathbf{x}$-$y)^2$, and it's gradient will have the from of $l'(\mathbf{x}_i,y_i,\mathbf{w})=(\mathbf{w}^T\mathbf{x}-y)^2$. Using this in the calculation of the loss for a group in Equation 4, for the k groups of

Algorithm 1. Bundle methods for solving the k-group quadratic mean minimization problem

Input: convergence threshold ϵ; initial weight vector $\mathbf{w}_0$;

// Step 1: Change point detection
Initialize distribution windows $Window1$ (starting form the first sample) and $Window2$ (which follows $Window1$)
Initialize iteration index $k \leftarrow 1$, initialize group window v_k as empty
repeat
  if samples in $Window1$ and $Window2$ have different distribution (according to Wilcoxon rank sum test) - change point detected **then**
    $v_k \leftarrow$ samples starting $Window1$ till end of $Window2$
    Set $Window1$ after v_k, $Window2$ follows $Window1$
    $k \leftarrow k+1$
  else
    Move $Window2$ one sample further
  end if
until all samples passed
// Step 2: Weight vector training:
Initialize iteration index $t \leftarrow 0$;
repeat
  $t \leftarrow t + 1$;
  Compute subgradient $a_t \leftarrow \partial_{\mathbf{w}} \mathrm{R}^Q_{emp,k}(\mathbf{w}_{t-1})$;
  Compute bias $b_t \leftarrow \mathrm{R}^Q_{emp,k}(\mathbf{w}_{t-1}) - \mathbf{w}^T_{t-1} a_t$;
  Update the lower bound $\mathrm{R}^{lb}_t(\mathbf{w}) = \max_{1 \le i \le t} \mathbf{w}^T a_i + \mathrm{b}_i$;
  Update $\mathbf{w}_t \leftarrow \arg\min_w J_t(\mathbf{w}) = \lambda \mathbf{w}^T \mathbf{w} + \mathrm{R}^{lb}_t(\mathbf{w})$;
  Compute current gap $\epsilon_t \leftarrow \min_{0 \le i \le t} J(\mathbf{w}_i) - J_t(\mathbf{w}_t)$
until $\epsilon_t \le \epsilon$ or $\epsilon_t - \epsilon_{t-1} \approx 0$
Return $\mathbf{w}_t$

samples, the subgradient function will have the form of Equation 7, and the detailed pseudo-code of the entire learning process is presented as Algorithm 1.

$$\partial_{\mathbf{w}} R^Q_{emp,k}(\mathbf{w}) = \partial_{\mathbf{w}} \sqrt{\frac{\sum_{j=1}^{k} f_j(\mathbf{w})^2}{k}} = \frac{1}{2} \left(\frac{\sum_{j=1}^{k} f_j(\mathbf{w})^2}{k}\right)^{-\frac{1}{2}} \left(\sum_{j=1}^{k} \frac{2 f_j(\mathbf{w}) f'_j(\mathbf{w})}{k}\right) \quad (7)$$

3.3 "Every Sample as a Group" Strategy

We compare our method of k groups with 2 extreme cases - when there is only one group, and when every single instance is a group. It can be easily shown that in the case of 1 group, the quadratic mean is equivalent to the standard linear regression model. As a single instance can be represented as a group, we investigate this research direction as well. The resulting model is QMSampleGroup. In this case the quadratic mean will try to minimize the variance of error across the entire set of samples.

4 Experiments and Results

Evaluation of the performance of the QMReg method was the main target of the experimental work we conducted. To achieve this goal in the experiments both real datasets and synthetic datasets were used. We tested 20 real stock market time series datasets obtained from [11], in the time frame of 2000-2012. We obtained daily stock market closing prices, one of the most often analysed types of data [1]. The sizes of the datasets are between 200 and 600 samples, divided on training set and test set.

A synthetic dataset was created, and 4 different versions of it were also tested (Table 1). The initial set represented a visible deterministic trend, after which 2 types of changes were introduced: increasing or decreasing the last samples, and adding different amounts of noise, in order to test the newly proposed algorithm the ability to work with noisy data. We also performed testing on 5 non-financial time series, revealing opportunities for application of quadratic mean based approach in the non-financial time series domain [12].

Table 1. Synthetic datasets description

Dataset	Description
Simulated sample 1	Visible deterministic trend
Simulated sample 2	Deterministic trend, structural break(in same direction), caused by increasing the last samples, noise(10% of original value)
Simulated sample 3	Deterministic trend, structural break(in opposite direction), caused by decreasing the last samples, noise(10% of orig. value)
Simulated sample 4	Deterministic trend, structural break(in same direction), last samples increased further, noise(25% of original value)
Simulated sample 5	Deterministic trend, structural break(in opposite direction), last samples decreased further, noise(25% of original value)

4.1 Testing and Results

Comparison between 6 methods was conducted in order to evaluate the effect of the QMReg methodology: Standard Least Squares Linear Regression (LS, regressing to past 4 values), Distribution based Quadratic Mean Linear Regression (QMReg, regressing to past 4 values), Quadratic Mean Linear Regression with every sample as a group (QMSampleGroup, regressing to past 4 values), ARIMA(3,0,1), Robust Regression (Huber M-estimator) and SVM Regression(SVM, $\alpha = 1$, C=1, ϵ=0.001, ξ=ξ^*=0.001),). The Root Mean Square Error(RMSE) was chosen as a performance metric, and we also calculate the error reduction (ER) compared to the Least Squares models:

$$ER = \frac{\text{RMSE of LS} - \text{RMSE of QS methods}}{\text{RMSE of LS}} * 100.$$

From the results presented in Table 2, we can clearly see that the QMReg method performed significantly better than the standard Least Squares linear regression,

Table 2. Root Mean Square Error (RMSE) and Error reduction (ER) of QMReg and QMGroupSample methods compared to LS(in %)

Datasets	RMSE						ER
	LS	QMReg	QM Sample Group	ARIMA	Robust Regress.	SVM Regress.	QMReg
Amazon.com	0.725	0.625	0.628	1.85	0.6269	0.629	**13.8**
Apple	3.75	3.513	3.4	16.1	3.566	3.44	**6.3**
American Express	0.913	0.645	0.64	1.61	0.635	0.642	**29.4**
British Airways	0.129	0.1163	0.118	0.228	0.121	0.1182	**9.6**
Boeing	1.36	1.22	1.23	8.28	1.188	1.15	**10.3**
Coke	0.558	0.378	0.428	1.04	0.375	0.382	**32.3**
Colgate Plamolive	0.996	0.69	0.714	1.6	0.693	0.707	**30.7**
Ebay	0.66	0.615	0.622	2.17	0.616	0.614	**6.8**
Fedex	1.99	1.13	1.132	1.67	1.098	0.87	**43.2**
Ford	0.276	0.273	0.275	0.85	0.273	0.267	**1.1**
Hewlett-Packard	0.889	0.554	0.581	1.94	0.566	0.589	**37.7**
IBM	7.54	7.17	7.826	25.6	7.725	7.443	**4.9**
Intel	0.932	0.819	0.83	2.36	0.832	0.838	**12.1**
Island Pacific	1.18	1.12	1.085	1.11	1.113	1.1	**5.1**
Johnson & Johnson	0.439	0.382	0.413	1.1	0.392	0.399	**13.0**
McDonalds	0.756	0.71	0.716	2.5	0.719	0.704	**6.1**
Microsoft	0.439	0.256	0.383	0.91	0.258	0.27	**41.7**
Starbucks	0.619	0.618	0.614	1.86	0.621	0.624	**0.2**
Siemens	2.391	2.1	2.12	3.4	2.097	2.1	**12.2**
Walt Disney	0.75	0.565	0.54	3.2	0.542	0.552	**24.7**
Simulated sample 1	1.82	1.79	1.668	11.87	1.81	1.96	**1.6**
Simulated sample 2	5.6	1.838	1.949	12	1.781	1.843	**67.2**
Simulated sample 3	2.65	1.88	1.898	11.9	1.982	2.06	**29.1**
Simulated sample 4	2.52	2.14	2.81	12.8	2.31	2.17	**15.1**
Simulated sample 5	8.6	6.9	7.39	8.26	6.77	2.67	**19.8**
Chemical process	0.241	0.225	0.274	0.36	0.234	0.228	**6.6**
Temperature anomalies	14.38	13.52	13.57	16.3	13.53	13.57	**6.0**
Radioactivity	14.63	12.315	14.44	10.1	11.197	10.72	**15.8**
Airline passengers	49.78	45.81	46.6	118.8	48.398	56.3	**8.0**
Chocolate	1764	1382	1635	1970	1680	1686	**21.7**
Wilcoxon signed rank test p-value	base	1.83E-06	1.3E-04	3.56E-05	9.78E-06	9.31E-05	
		base	0.00286	1.30E-05	0.2563	0.2845	
			base	1.43E-05	0.1682	0.2845	

much better than the ARIMA model, and very similar to the Robust Regression and SVM, and was the method with the most stable convergence towards the optimal weight vector as well, which was not always the case with Least Squares. The performance of the QMReg method was mostly greater for datasets where higher number of groups was detected. The SVM output arguable less easy to interpret - support vector samples in the dataset are not as much of use as simple coefficients for the features. With similar performance as SVM regression,

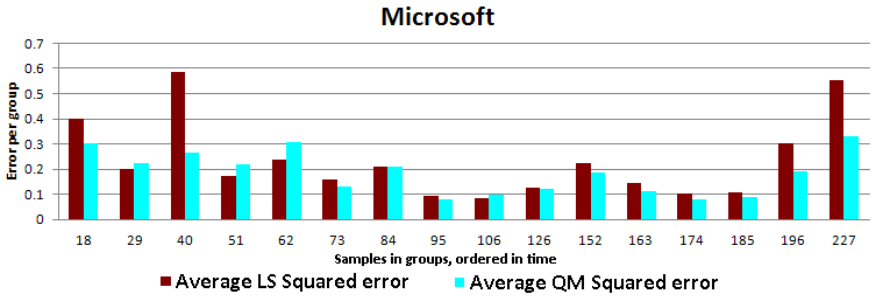

Fig. 2. Error between the groups of the loss, for Microsoft dataset

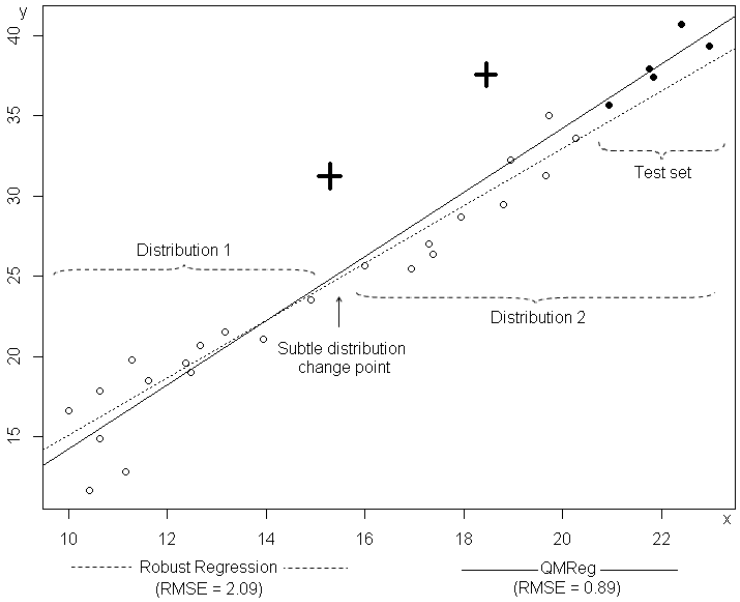

Fig. 3. Robust regression and QMReg comparison - the pluses are the outliers, and the black dots are future samples to be predicted. Robust regression considers the points at the very end as possible outliers, while QMReg detects the subtle change of direction and adjusts the line accordingly.

QMReg does show potential for use in cases when one is not familiar with more complex forecasting methods, but can interpret the output of a linear regression model. The objective of the QMReg to reduce the loss between groups can be seen from Figure 2 which graphically shows how the errors per group are minimized and made more even when QM grouping in performed.

The Robust regression is easy to interpret too, but a more statistical analysis of the data is required - the non-parametric QMReg can be used as a black-box method, and still deliver similar results. The difference with Robust regression can be seen from Figure 3: the crossed points are outliers, and red points are

the test samples. We can notice that both methods are dealing successfully with the outliers, but the subtle change in the distribution is more correctly detected by the QMReg method, while robust regression is considering the end points as possible outliers and still keeps the line towards the overall mean. It is this subtle change detection that further leads to lower error for QMReg (RMSE=0.89) than the Robust regression error (RMSE=2.09). We can see the QMReg is highly competitive when compared to Robust regression, and based on Figure 3 we believe it may have the potential to deal with non-stationary behaviour better.

4.2 Loss Function Analysis

By performing grouping among the training set, the QM methods tend to minimize not only the overall training error, but the error per group. This results in a lower variance in the loss, as it can be seen from Table 3: the standard deviation in QMReg method was at its best up to 58% less than the one of the LS method. The grouping performed in the case of QMReg was also conducted for LS. The loss per group was calculated, and standard deviation among the loss per group showed that the QM method indeed minimizes the loss amongst the groups: Table 3 shows the standard deviation of the error per group is statistically lower in the case of QMReg when compared to LS.

5 Conclusion and Future Work

Time series forecasting is a classic prediction problem, for which linear regression is one of the best known and most widely used methods. In this paper, we have proposed a technique that enhances standard linear regression, by employing an optimization objective which explicitly recognises different groups of samples. Each group corresponds to a segment of the time series whose samples have similar distribution characteristics. Our objective simultaneously minimizes the expected loss of each group, as well as the variance of the loss across the groups. By doing so, we ensure a model that produces more stable, less volatile predictions, and capable of additionally minimizing the effect of outliers or noisy data.

The experimental study highlighted the promise of our approach, showing that it could produce linear models different to that of standard linear regression and which also achieved consistent reductions in error in the range 10% to 30% on average, up to 40% error reduction in some cases. As the performance of our proposed method is comparable to more advanced forecasting methods(SVM regression and Robust Regression), our model has an advantage that it improves the performance of linear regression while avoiding unnecessary complexity and unwanted parameters, so it can be used by more general practitioners on diverse types of time series.

For future work, we would like to investigate the use of weighted groupings, for cases where the quality of a group with respect to the prediction task can be estimated, and introduce the time element in the learning process even further.

Table 3. Loss function mean and standard deviation for entire training set, and standard deviation of loss across groups per training set

Dataset	Loss function				Standard deviation for loss across groups	
	LS		QMReg			
	mean	st. dev.	mean	st. dev.	LS	QMReg
Amazon.com	0.7074	1.1087	0.5081	0.8229	0.4888	0.3139
Apple	15.1928	27.2601	13.9676	26.0380	8.8954	8.3127
American Express	1.1356	2.2984	0.8372	1.5941	0.8179	0.5714
British Airways	0.0303	0.0483	0.0203	0.0398	0.0199	0.0156
Boeing	2.2065	3.5453	0.9353	1.5959	1.1735	0.4683
Coke	0.4660	1.1622	0.2504	0.8432	0.5104	0.2234
Colgate Plamolive	1.3929	2.8065	0.7759	2.1025	1.2410	0.5965
Ebay	0.2531	0.4295	0.2224	0.4208	0.1550	0.1060
Fedex	6.4903	9.3358	2.5329	3.9214	3.9518	1.3231
Ford	0.0680	0.1184	0.0681	0.1191	0.0485	0.0481
Hewlett-Packard	0.9649	2.2101	0.5336	1.2165	0.9802	0.4411
IBM	85.553	133.895	60.431	123.244	52.7541	56.8021
Intel	1.2986	2.4139	1.1098	2.0457	0.6658	0.4592
Island Pacific	0.9762	1.3706	0.5165	0.8315	0.4787	0.1660
Johnson & Johnson	0.2526	0.4612	0.2372	0.4137	0.1572	0.1343
McDonalds	0.3491	0.7640	0.2742	0.5731	0.2595	0.1650
Microsoft	0.2600	0.6304	0.1980	0.5074	0.1563	0.0867
Starbucks	0.1654	0.4354	0.1672	0.4239	0.1295	0.1232
Siemens	4.6880	6.1569	4.0694	5.4571	1.7561	1.2753
Walt Disney	0.4286	0.7747	0.2374	0.4255	0.3840	0.1434
Simulated sample 1	3.5835	3.2914	2.9072	2.7637	0.8706	0.6042
Simulated sample 2	14.5109	26.8793	4.6307	15.8900	13.8375	5.0620
Simulated sample 3	9.3680	18.8459	3.7615	8.3814	5.8793	1.6188
Simulated sample 4	40.6497	86.3151	31.5504	80.3505	38.3393	29.8716
Simulated sample 5	93.7317	133.0539	53.3164	104.7921	51.9541	29.3767
Chemical process	0.0680	0.1442	0.0756	0.1653	0.0339	0.0277
Temperature anomalies	232.47	368.44	206.57	318.60	16.7000	52.8821
Radioactivity	175.04	230.35	180.35	244.17	46.7919	31.9748
Airline passengers	809.82	1418.44	796.04	1130.33	490.8909	426.3732
Chocolate	1499216	2158796	952022	1486350	622544.00	467145.00
Wilcoxon signed rank test p-value	base		2.72E-05	3.90E-05	base	0.0001816

References

1. Tsay, R.S.: Analysis of Financial Time Series. Wiley-Interscience (2005)
2. Hulten, G., Spencer, L., et al.: Mining time-changing data streams. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, pp. 97–106 (2001)
3. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)

4. Dong, G., Han, J., et al.: Online mining of changes from data streams: Research problems and preliminary results. In: Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams (2003)
5. Liu, X., Zhang, R., et al.: Incremental Detection of Distribution Change in Stock Order Streams. In: 26th International Conference on Data Engineering Conference, ICDE (2010)
6. Teo, C.H., Vishwanthan, S.V.N., Smola, A.J., Le, Q.V.: Bundle methods for regularized risk minimization. Journal of Machine Learning Research 11, 311–365 (2010)
7. Liu, W., Chawla, S.: A Quadratic Mean based Supervised Learning Model for Managing Data Skewness. In: Proceedings of the Eleventh SIAM International Conference on Data Mining, pp. 188–198 (2011)
8. Vellaisamy, K., Li, J.: Multidimensional decision support indicator (mDSI) for time series stock trend prediction. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 841–848. Springer, Heidelberg (2007)
9. Cheng, H., Tan, P.-N., Gao, J., Scripps, J.: Multistep-ahead time series prediction. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 765–774. Springer, Heidelberg (2006)
10. Liu, Z., Yu, J.X., Lin, X., Lu, H., Wang, W.: Locating motifs in time-series data. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 343–353. Springer, Heidelberg (2005)
11. Web enabled scientific services and applications, `http://www.wessa.net/stocksdata.wasp`
12. Hyndman, R.J.: S&P quarterly index online database, `http://robjhyndman.com/tsdldata/data/9-17b.dat`
13. Muller, K.-R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Using Support Vector Machines for Time Series Prediction (2000)
14. Liu, X., Wu, X., Wang, H., Zhang, R., Bailey, J., Kotagiri, R.: Mining distribution change in stock order streams. In: IEEE 26th International Conference on Data Engineering, ICDE (2010)
15. Wilcox, R.R.: Introduction to Robust Estimation and Hypothesis Testing. Elsevier Academic Press, New York (2005)
16. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
17. Adhikari, R., Agrawal, R.K.: A novel weighted ensemble technique for time series forecasting. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012, Part I. LNCS, vol. 7301, pp. 38–49. Springer, Heidelberg (2012)
18. Khoa, N.L.D., Chawla, S.: Robust outlier detection using commute time and eigenspace embedding. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part II. LNCS, vol. 6119, pp. 422–434. Springer, Heidelberg (2010)
19. Widiputra, H., Pears, R., Kasabov, N.: Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 161–172. Springer, Heidelberg (2011)
20. Cheng, H., Tan, P.-N.: Semi-supervised learning with data calibration for long-term time series forecasting. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2008)
21. Meesrikamolkul, W., Niennattrakul, V., Ratanamahatana, C.A.: Shape-based clustering for time series data. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012, Part I. LNCS, vol. 7301, pp. 530–541. Springer, Heidelberg (2012)