Time-Series-Based Classification of Financial Forecasting Discrepancies

Ben Peachey Higdon(✉), Karim El Mokhtari, and Ayşe Başar

Data Science Laboratory, Ryerson University, Toronto, Canada
bpeachey@ryerson.ca

Abstract. We aim to classify financial discrepancies between actual and forecasted performance into categories of commentaries that an analyst would write when describing the variation. We propose analyzing time series in order to perform the classification. Two time series classification algorithms – 1-nearest neighbour with dynamic time warping (1-NN DTW) and time series forest – and long short-term memory (LSTM) networks are compared to common machine learning algorithms. We investigate including supporting datasets such as customer sales data and inventory. We apply data augmentation with noise as an alternative to random oversampling. We find that LSTM and 1-NN DTW provide the best results. Including sales data has no effect but inventory data improves the predictive power of all models examined. Data augmentation has a slight improvement for some models over random oversampling.

Keywords: Time series classification · Data augmentation · LSTM · Machine learning

1 Introduction

An important aspect of budgeting for large companies is having reliable forecasts of future performance. Finance managers aim to understand the variance between a financial forecast and actual performance. They generate commentaries to explain the reasons for such variance by collecting information from different aspects of the organisation such as sales, inventory and logistics.

In our previous work [4], we proposed generating these financial commentaries based on variance data from a consumer goods company using recurrent neural networks. The commentaries we used were inconsistently worded which made this a challenge when working with real world data. In this paper we simplify the problem and aim to predict the category of commentary rather than generating the text itself. Additionally, we now consider training on a one year long monthly time series of variance to account for past patterns. We test two vector-based algorithms and compare them to two time series classification algorithms and

Supported by SOSCIP and Mitacs.

M. Bramer and M. Petridis (Eds.): SGAI-AI 2019, LNAI 11927, pp. 474–479, 2019.
https://doi.org/10.1007/978-3-030-34885-4_39

a deep learning approach. We also investigate including auxiliary datasets to improve predictions. We apply data augmentation techniques to the time series and compare this approach against random oversampling.

2 Background

Time series classification algorithms can be grouped into families depending on how the classification is performed. One family of algorithms relies on distance measures to determine the similarity between time series instances. Commonly used is a nearest neighbour classifier with dynamic time warping (DTW) as the distance measure [1]. DTW is an elastic distance measure that allows for some warping on the time axis to find a better match between time series.

One nearest neighbour (1-NN) classification with DTW is found to be a strong baseline when compared to many time series classification algorithms [1].

Other time series classification algorithms first define features of the series such as the mean or standard deviation [10]. A classifier is then trained on the features and not the values of the time series. This approach can be extended to calculate features for specific intervals of the time series. One such algorithm known as time series forest (TSF) trains decision trees on features from randomly selected intervals and the majority vote from all trees is the final output [3]. The calculated features used in TSF are the mean, standard deviation and the slope of the chosen intervals. Interval feature methods such as TSF can provide insight into the location of the most important regions and features of a phase-dependant time series. Bagnall et al. [1] conducted experiments using several algorithms on a large time series repository and found TSF to be one of the stronger algorithms albeit its simplicity.

Recurrent neural networks (RNNs) are a family of deep learning models that excel at processing sequential data. Long short-term memory (LSTM) networks [6] are RNNs that have become popular as they can retain information across long sequences. Lipton et al. [9] used LSTMs to predict diagnoses from multivariate time series of medical observations such as blood pressure. Their LSTMs were able to outperform a strong baseline with hand-engineered features.

Inspired by data augmentation in image classification [8,11] we explore augmenting time series. In image classification tasks, data is augmented by transforming the images in such a way that the label is preserved e.g. rotation, translation. The advantage of augmentation is two-fold – the risk of overfitting is reduced and more training examples are generated [8].

3 Dataset Description

The main dataset that an analyst uses to write a commentary consists of the difference of the forecasted and actual shipments in dollars for every brand and every customer. For us, the targets to predict are the commentary classes. We split the commentaries into five classes (*promotion*, *point of sale*, *phasing*, *other* and *no comment*) based on a list of keywords common to each class. We remove some labels from very few multi-labeled instances according to a class prediction

priority. For example, if an instance is labeled as both *promotion* and *phasing*, and *phasing* is higher on the priority list, the label is changed to be only *phasing*.

For every brand, month and customer, we construct a time series of the past 13 months of variance data $V = (v_1, v_2, \ldots, v_{13})$ where v_{13} is the variance corresponding to the month the commentary is generated. We remove any brands that fall within a threshold variance as well as all time series with zero data, as these will be automatically classified as *no comment*.

Additionally, we have been provided with point of sale (POS) and inventory datasets from one of the customers of the consumer goods company. The POS dataset lists total sales in dollars and the number of items sold for each brand. The inventory dataset lists the various inventory measures including the amount of units in stock. With these auxiliary datasets we construct 13 month time series of average unit price and average amount of in-store stock. Including these extra datasets requires us to only use data from the one customer which leaves us with a smaller subset of data. The full data and this one customer subset are imbalanced, with most of the time series having no comment.

4 Methods

We employ two widely used machine learning algorithms as a baseline: support vector machine (SVM) and a random forest (RF). Both SVM [7] and RF [1] have been shown to be well suited towards time series classification and both are thoroughly described in literature. We implement two algorithms, 1-NN DTW and TSF which are well suited to time series classification to compare against the baselines. We choose these two algorithms for their simplicity as a first step towards investigating time series classification algorithms for this application. Interval based time series such as TSF are especially interesting as we foresee some regions of the time series will carry more meaningful information.

We also employ two LSTM-based models, the first of which is a single LSTM followed by a softmax layer to output the classification. The second architecture, a multi-LSTM, depicted in Fig. 1, has an LSTM layer per dataset. The encoded results of each LSTM are concatenated and passed to a softmax layer. This is a more complex model and takes more time to train than a single LSTM but we believe it better represents the different patterns in each dataset.

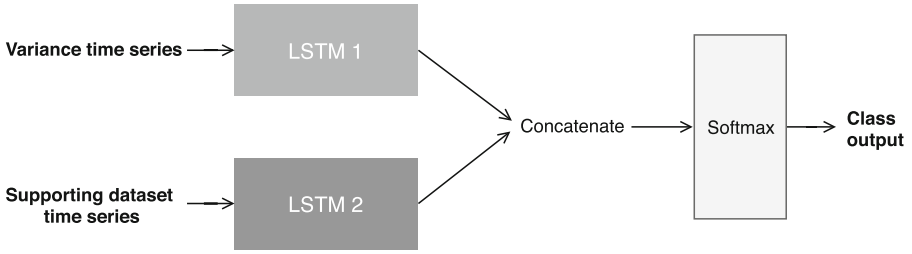

Fig. 1. The multi-LSTM architecture used when including supporting data

When including the supporting datasets, the vector-based classifiers train on the concatenated values of the time series involved. For the LSTM, TSF and 1-NN DTW models the time series can be treated as multidimensional. The multi-LSTM receives the time series separately.

To overcome the class imbalance problem we randomly oversample, copying instances from the smaller classes until the class distribution is balanced [5]. As an alternative to random oversampling we investigate adding Gaussian noise to augment the data [2,11]. Instead of duplicating the smaller classes we augment the time series to achieve a balanced class distribution. We predict that it is not the values of the time series that are most important but the trends of the data. We do not expect that adding a small amount of noise to this data will impact the true class but it will provide us with a more diverse training set than randomly oversampling.

We train the models on seeded resampled train-test sets and average the results across all runs. For every run the time series are z-normalized then oversampling or augmenting are applied to the training set. We use macro-averaged F1 score as the main performance metric. The macro-average of a metric is the average of the metric calculated for each class. If there are no positive predictions for a class the F1 score is treated as zero for that class. We choose this metric because it can reveal the performance on minority classes [5,7].

5 Experiments

In the first experiment we use a one customer subset of the variance data and compare results when including the POS or the inventory dataset. Model parameters are optimized for each combination of supporting data to account for the new information. Training data is randomly oversampled across all runs in this experiment. As seen in Fig. 2 the inclusion of POS data has no benefit whereas the inventory dataset does improve the mean F1 score in all models.

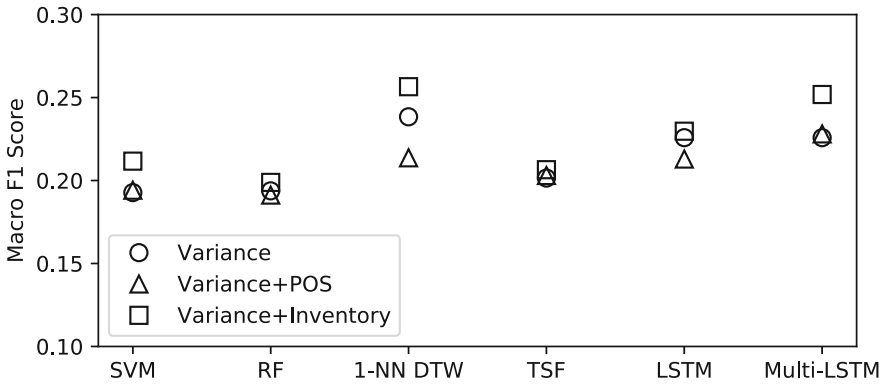

Fig. 2. Mean macro-averaged F1 scores on single customer subset when including supporting POS and inventory data

We use the full variance data in the second experiment to test whether we can improve over randomly oversampling by augmenting the time series with added Gaussian noise with mean of 0 and standard deviation of 0.01. Model parameters are kept constant across all tests. The results are shown in Fig. 3. Augmentation slightly improves over random oversampling with the RF and the LSTM models. The time series classification algorithms are not affected by augmenting likely because they are already capturing the trend of the data.

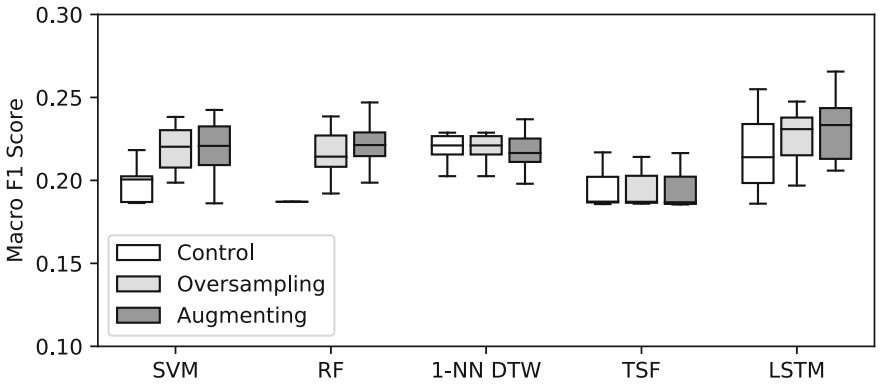

Fig. 3. Macro-averaged F1 scores on full variance data when oversampling and augmenting compared to the control

In both experiments 1-NN DTW and the LSTM outperform the baseline algorithms. Although we expect TSF to do well it does not surpass the baseline. This may have to do with how the random intervals are chosen. Since all intervals are equally likely, the points in the centre of a time series are more likely to be included. However, the edges of our time series have essential information, i.e. current month and previous year. We expect that selecting intervals with the goal of a uniform point distribution would perform better on our data.

6 Conclusion

In this paper we investigate techniques for classifying financial time series into categories of commentaries used in forecasting reports. We examine the effect of including supporting sales and inventory time series. When adding a supporting dataset we see that an LSTM per dataset architecture outperforms a single LSTM. Our results show a positive effect in all models when including inventory data. Although these supporting datasets are difficult to acquire for the consumer goods company we show that inventory data can be valuable to the classification. Finally, we compare data augmentation via added noise and random oversampling as methods to reduce overfitting and show that augmenting has a minor improvement over random oversampling.

Across both the full data and the one customer subset, the 1-NN DTW and LSTM models are seen to be the strongest. Because these two successful models are specialized at capturing patterns from sequences we draw the conclusion that there are temporal features in the data that aid in the classification of the forecasting discrepancies. This validates our approach of analyzing time series for this application.

In future work we will continue to investigate the effect of including supporting datasets. With POS and inventory data from more customers we will have a larger subset of the data to work with and thus more meaningful results. Since 1-NN DTW has proven to be effective we will evaluate further time series classification algorithms.

Acknowledgments. This work is supported by grants from Mitacs and Smart Computing for Innovation (SOSCIP) consortium.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov. **31**(3), 606–660 (2017)
2. Cortes-Ciriano, I., Bender, A.: Improved chemical structure-activity modeling through data augmentation. J. Chem. Inf. Model. **55**(12), 2682–2692 (2015)
3. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Inf. Sci. **239**, 142–153 (2013)
4. El Mokhtari, K., Maidens, J., Bener, A.: Predicting commentaries on a financial report with recurrent neural networks. In: Meurs, M.-J., Rudzicz, F. (eds.) Canadian AI 2019. LNCS (LNAI), vol. 11489, pp. 531–542. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18305-9_56
5. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Köknar-Tezel, S., Latecki, L.J.: Improving SVM classification on imbalanced time series data sets with ghost points. Knowl. Inf. Syst. **28**(1), 1–23 (2011)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
9. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with LSTM recurrent neural networks. arXiv preprint: arXiv:1511.03677 (2015)
10. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time-series data. Int. J. Comput. Res. **10**(3), 49–61 (2001)
11. Zur, R.M., Jiang, Y., Pesce, L.L., Drukker, K.: Noise injection for training artificial neural networks: a comparison with weight decay and early stopping. Med. Phys. **36**(10), 4810–4818 (2009)