Classifying financial forecasting discrepancies*

No Author Given

Data Science Laboratory, Ryerson University, Toronto, Canada

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 financial time series leading up to the discrepancy in order to perform the classification. We investigate what models are best suited towards this problem. Two simple 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 examine the effect of including supporting datasets such as customer sales data and inventory. We also consider the result of data augmentation with noise as an alternative to random oversampling. The findings are that the LSTM and a 1-NN DTW models provide the best results. Including customer inventory data improves the predictive power of all models examined while sales data has no effect. Data augmentation grants a slight improvement by 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. When there is a variance between a financial forecast and actual performance it is crucial to understand its origin. Informed by financial data including sales, inventory and logistics, analysts write commentaries to explain why the variances appear.

In a previous work [1], it was proposed to generate these financial commentaries based on variance data from a consumer goods company using recurrent neural networks. The commentaries 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 text. 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 to two time series classification algorithms and a deep learning approach. We investigate including auxiliary datasets to improve predictions. We also apply data augmentation techniques to the time series and contrast against random over sampling.

* Supported by organization x.

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 either Euclidean distance or dynamic time warping (DTW) as the distance measures [2]. DTW is an elastic distance measure that allows for some warping on the time axis to find a better match between time series. If we have two time series S and T with elements $(s_1, s_2, \ldots, s_n)$ and $(t_1, t_2, \ldots, t_n)$ we can define a grid of the distances between the series $D(i, j) = (s_i - t_j)^2$. Then a warping path $W = w_1, w_2, \ldots, w_f$ where each component of W is a distance in the grid $D(i, j)$. The warping path has a few constraints, namely: $w_1 = (1, 1)$, $w_f = (n, n)$, $0 \leq i_{k+1} - i_{k+2} \leq 1$ and $0 \leq j_{k+1} - j_{k+2} \leq 1$. The DTW distance is the minimal sum from the sums of all warping paths $W \in \mathcal{W}$

$$P_W(S, T) = \sum_{k=1}^{f} w_k$$

$$DTW(S, T) = \min_{W \in \mathcal{W}} P_W(S, T)$$

One nearest neighbour (1-NN) classification with DTW is found to be a very strong baseline when compared to many time series classification algorithms [2].

Other time series classification algorithms first define features of the series such as the mean or standard deviation [4]. 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, the majority vote from all trees is the final output [5]. The calculated features used in TSF are the mean, standard deviation and the slope of the chosen intervals. In order to reduce complexity TSF selects $\sqrt{M}$ intervals where M is the length of the time series, resulting in $3\sqrt{M}$ total features. 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. Time series classification algorithms are commonly tested on the University of California, Riverside time series repository[3]. Bagnall et al. [2] conducted experiments using several algorithms on the 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 very popular as they can retain information across long sequences. Lipton et al. [7] used LSTMs to predict diagnoses from multivariate timeseries of medical observations such as blood pressure and body temperature. Their LSTMs were able to outperform a strong baseline with hand-engineered features. LSTMs have also seen success in speech recognition; using bidirectional LSTMs Graves et al. [8] recorded the lowest error at the time when classifying a standard set of phonemes.

Inspired by data augmentation in image classification [9, 10] 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.

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 three classes (*promotion*, *point of sale*, *phasing*) based on a list of keywords common to each class. There are two additional classes– commentaries not containing keywords are labeled as *other* and all instances without a commentary are given a *no comment* label. 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 leading up to the commentary. 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*. The resulting dataset has 2643 time series belonging to the 5 classes.

Additionally, we have been provided with point of sale (POS) and inventory datasets from one of the consumer goods company's customers. 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 monthly time series of average unit price and average amount of stock to support the variance time series. Including these extra datasets requires us to use data only from the one customer which leaves us with 286 instances of data. The full data and this one customer subset are heavily 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 random forest. These vector-based algorithms do not capture the temporal features of time series. 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 which we will call a multi-LSTM illustrated in Fig 1 has an additional LSTM layer for each included 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 has a stronger ability to represent the very different patterns of the two datasets.

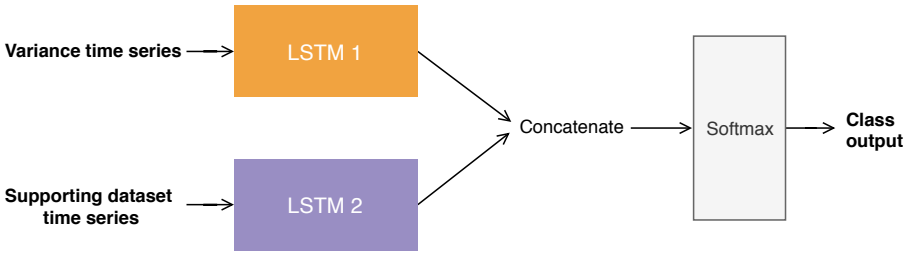

Fig. 1: The multi-LSTM architecture used when including supporting data

When including the supporting POS and inventory 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.

In order to combat the class imbalance, we randomly oversample, copying instances from the smaller classes until the class distribution is balanced. As an alternative to random oversampling we investigate adding Gaussian noise to augment our time series. Instead of duplicating the smaller classes we augment the time series to achieve a balanced class distribution. For this application we predict that it is not the values of the time series that are most important but the trend of the data. Adding a small amount of noise to the time series should not impact the true class but 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. We also monitor the macro-averaged precision and recall. 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. This type of averaging gives insight into the results on the smaller classes which is what we seek in the case of our imbalanced dataset.

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 Figure 2 the inclusion of POS data has no benefit whereas the inventory dataset does improve the mean F1 score in all models.

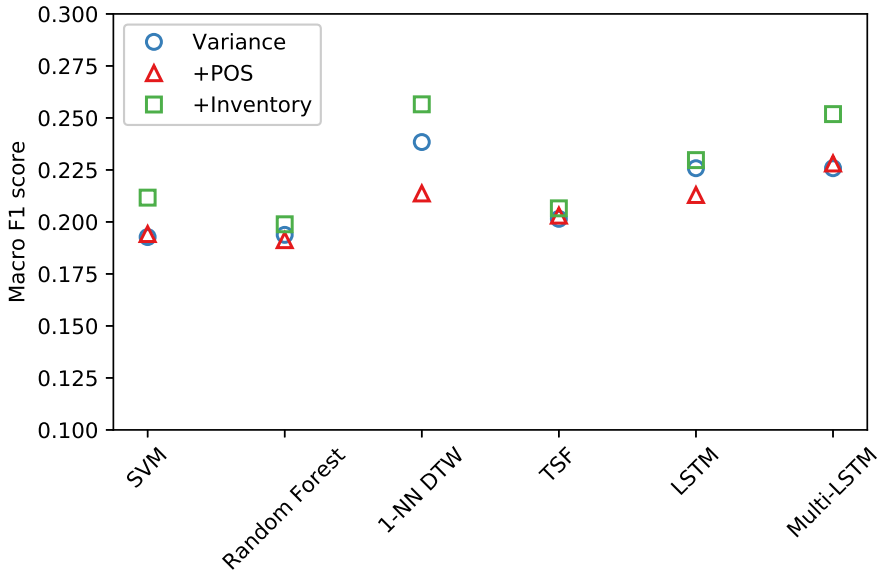

Fig. 2: Mean macro-averaged F1 scores on single customer subset when including supporting datasets

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. The standard deviation was chosen out of a range from 0.001 to 0.2 through cross validation. Model parameters are kept constant across all tests. The results are shown in Figure 3. Augmentation slightly improves over random oversampling with the random forest 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.

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

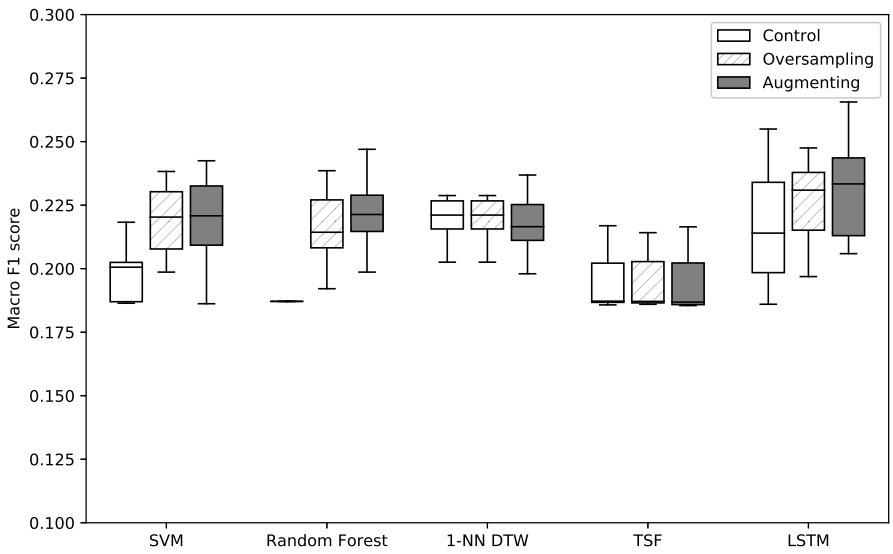

Fig. 3: Macro-averaged F1 scores on full data when oversampling and augmenting

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 Threats to validity

In this work, we deal with a small and imbalanced dataset. To achieve significant results we make the following efforts to overcome these threats to the validity. In regards to the size, we run our experiments 25 times on resampled train-test splits to achieve meaningful results. The choice of an appropriate metric will avoid misleading results arising from the class imbalance. We choose the macro-averaged F1 score as our main evaluation metric as it will capture how the models are performing on the smaller classes. In addition we use random oversampling or data augmentation to mitigate the effects of the class imbalance.

7 Conclusion

In this paper we investigate techniques for classifying financial time series into categories of commentaries used in forecasting reports. The main dataset is the variance between forecasted and actual shipments from a consumer goods company. We compare models well suited to time series classification (1-NN DTW, TSF and LSTM) to two baseline algorithms (SVM and random forest) using

macro-averaged F1 score. We examine the effect of including POS and inventory datasets for a small one customer subset. When adding a supporting dataset we test a single LSTM as well as a multi-LSTM architecture. We observe that the multi-LSTM is outperforming the single LSTM in this case. Including the inventory dataset has a positive effect on all models, while POS showed no improvements over only using variance data. We then consider data augmentation via added noise as a way to artificially increase the size of the dataset and reduce overfitting. Small improvements can be seen over randomly oversampling in the LSTM and random forest models. Across both the full data and the one customer subset, 1-NN DTW and LSTM models are shown to be the strongest.

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.

References

1. El Mokhtari K., Maidens J., Bener A. (2019) Predicting Commentaries on a Financial Report with Recurrent Neural Networks. In: Meurs MJ., Rudzicz F. (eds) Advances in Artificial Intelligence. Canadian AI 2019. Lecture Notes in Computer Science, vol 11489.
2. 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 Mining and Knowledge Discovery. 31, 606-660 (2016).
3. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G. (2015). The UCR time series classification archive.
4. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based Classication of Time-series Data. Information Processing and Technology. 49-61 (2001).
5. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences. 239, 142-153 (2013).
6. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation. 9, 1735-1780 (1997).
7. Lipton, Z. C., Kale, D. C., Elkan, C., Wetzel, R. Learning to diagnose with LSTM recurrent neural networks. International Conference on Learning Representations (2016).
8. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing (2013).
9. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Communications of the ACM. 60, 84-90 (2017).
10. Yaeger, L. S., Lyon, R. F., Webb, B. J.: Effective training of a neural network character classifier for word recognition. Advances in neural information processing systems. 807-816 (1997).