A Case-Based Reasoning Approach to Company Sector Classification Using a Novel Time-Series Case Representation

Rian Dolphin[1], Barry Smyth[1,2], and Ruihai Dong[1,2]

[1] School of Computer Science, University College Dublin, Dublin, Ireland
rian.dolphin@ucdconnect.ie
[2] Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland
{barry.smyth, ruihai.dong}@ucd.ie

Abstract. The financial domain has proven to be a fertile source of challenging machine learning problems across a variety of tasks including prediction, clustering, and classification. Researchers can access an abundance of time-series data and even modest performance improvements can be translated into significant additional value. In this work, we consider the use of case-based reasoning for an important task in this domain, by using historical stock returns time-series data for industry sector classification. We discuss why time-series data can present some significant representational challenges for conventional case-based reasoning approaches, and in response, we propose a novel representation based on the factorization of a similarity count matrix, which can be readily calculated from raw stock returns data. We argue that this representation is well suited to case-based reasoning and evaluate our approach using a large-scale public dataset for the industry sector classification task, demonstrating substantial performance improvements over several baselines using more conventional representations.

Keywords: Case-Based Reasoning · Time-Series · Finance · Representation Learning

1 Introduction

Case-based reasoning (CBR) approaches have been applied in financial domains, and for a variety of tasks, from the early days of the field; see for example the work of [23] on the use of CBR for financial decision-making. In the years since, there have been many efforts made to apply CBR ideas to a diverse range of financial decision-making and prediction tasks [13,25,1,6]. Nevertheless, the use of CBR in such domains is not without its challenges, not the least of which concerns the very nature of many financial datasets and their relationship to the similarity assumption that underpins CBR. The central dogma of CBR is that similar problems have similar solutions, yet financial regulators are always at pains to point out that "past performance is not a guarantee of future results" suggesting that this principle may not be so reliable in financial domains, at least

when it comes to predicting the future. As society changes and economies ebb and flow, companies that were once the stock market darlings fall out of favour, while new winners seem to emerge, with some regularity, albeit unpredictably and often with little or no warning. Two companies that were similar in the past may no longer be considered similar in the present, while the trajectories of companies with divergent histories might suddenly converge if future circumstances conspire in their favour. All of this greatly complicates the central role of similarity in case retrieval.

In addition, the feature-based (attribute-value) representations that are commonplace in CBR systems may not provide such a good fit with the type of sequential time-series data (e.g. daily, weekly, and monthly stock prices/returns) that are the norm in financial domains. This is not to say that case-based methods cannot be used with time-series data. Certainly, there is a wealth of literature on representing time-series data for use with case-based methods in a range of application domains from agricultural science [3] to sports science [7]. Usually, the approach taken is to use various feature extraction techniques to identify landmark features from the raw time-series data; to effectively transform a raw time-series into a more conventional feature-based representation that can be used with standard similarity metrics. Similar approaches have been applied in the financial domain [12] but, as mentioned above, the stochastic nature of financial markets makes it difficult to effectively isolate useful case representations from market noise, which further complicates the similarity assessment even given a suitable fixed representation.

Thus, in this work, our main technical contribution is to propose and evaluate a novel approach to learning case representations for financial assets (companies/stocks) using raw time-series data made up of historical daily returns. We describe how to transform raw, time-series data into an embedding-style representation of each stock/company; see for example [16,17] for examples of embedding representations. We argue that this facilitates the capture of more meaningful patterns and sub-patterns over extended periods of time, while facilitating the type of temporal alignment that is necessary during case comparison and similarity assessment. We argue that this approach is well-suited to the use of case-based and nearest-neighbour approaches in financial domains, because it can be used with a variety of standard similarity metrics, as well as more domain/task specific metrics. We demonstrate its performance benefits in a comparative evaluation of the industry sector classification task, an important and practical benchmark problem in many financial applications [18].

The remainder of this paper is organised as follows. In the next section, we review the use of CBR in the financial domain and with time-series data more broadly, highlighting several common tasks and the approaches taken thus far, as well as the important challenges that remain with respect to representation and similarity assessment. Then, in Section 3 we discuss the details of our proposed approach, by describing how raw time-series data, such as financial returns, can be transformed into an embedding-based representation that is well suited to case-based approaches. In Section 4 we evaluate our proposed approach by using

it to classify companies into their market sectors based on their historical returns data. We present the results of a comprehensive quantitative evaluation, which compares our proposed representation to a variety of baseline and naive approaches. We demonstrate how our embeddings-based representations can offer significant classification improvements, relative to more conventional representations of the raw time-series data, as well as state-of-the-art neural models [20,4]. In addition, before concluding with a summary of our findings and a discussion of limitations and opportunities for further work, we present further qualitative evidence in support of the proposed approach, by using our representations to visualise the industry sectors that emerge from the clustering of our cases and some examples of nearest-neighbours in the resulting representation space.

2 Background

CBR continues to offer many benefits even in a world of big data and deep learning. Its so-called *lazy* approach to problem-solving, which retains the raw cases, offers several benefits when it comes to transparency, interpretability, and explainability [14]. And, the central role similarity plays – using similar cases to solve future problems – helps to lift the computational veil that all too often obscures the processes that drive more recent machine learning approaches [26]. That being said, the success of CBR is contingent upon the quality of the available cases and the suitability of the case representations and metrics used to evaluate case similarity and drive inference. CBR approaches have been particularly effective in domains where cases are plentiful and where feature-based representations are readily available. For example, in loan/credit approval tasks [24], past decisions provide a plentiful supply of relevant cases, and each case can be represented by salient features such as the value of the requested loan, the debt-load of the applicant, the current earnings of the applicant, the purpose of the loan etc.

However, in other financial domains the situation is more complex, especially when the available data is sequential/temporal in nature, as is often the case. When it comes to representation, several approaches have been proposed to capture the salient features of financial time-series data, such as asset prices of stock returns. They can be broadly categorised into three groups: (i) traditional feature-based summaries, (ii) raw time-series, and (iii) machine learning-based representations.

Feature-based representations of financial time-series tend to derive summary features from statistical moments and technical indicators [8]. For example, time-series data can be represented by extracting key statistical features (e.g. max/min values, mean and standard deviations etc.) over fixed time periods. In addition, the financial domain has the advantage of the availability of several widely-accepted technical indicators, which correspond to common patterns observed in historical trading data such as *on-balance volume*, the *accumulation/distribution line*, the *average directional index*, or the *Aroon indicator* [8]. These exotic-sounding indicators are among the tools of the trade for technical

analysts and day traders and can be readily extracted from financial time series data to provide a valuable source of domain-specific features.

In contrast to feature-based representations, some researchers have explored the use of raw time-series data in CBR applications. Here, instead of computing domain-specific features, the choice of similarity metric accounts for the temporal nature of the data. One popular time series similarity technique used in CBR is Dynamic Time Warping (DTW) [19], which measures the similarity between two time-series by allowing for temporal shifts in alignment in order to optimise the correspondence between the two time-series. While DTW has been successfully applied in CBR systems across various domains [3], it is not directly applicable to financial returns data, at least according to the type of tasks that are of interest in this work, because allowing significant temporal shifts in alignment can distort the relationships that exist between two stocks/assets; two stocks having similar returns only constitutes a meaningful relationship if those returns are aligned over the same period of time (in phase). More specifically, in the financial domain, authors in [1] propose a geometrically inspired similarity metric for financial time series, while [6] proposes a metric combining cumulative returns with an adjusted correlation. However, as we show in Section 4, applying a similarity metric to raw time-series data may not capture all of the relational information needed leading to poorer performance in some tasks.

More recently, so-called *distributed representations* [16] and the use of *embeddings* have become important in the machine learning literature, especially in natural language domains. Embeddings provide a way to translate a high-dimensional representation (such as text) into a low-dimensional representation, which can make it more straightforward to use machine learning techniques when compared with high-dimensional, sparse vectors such as a one-hot encoded vocabulary. Embeddings have been shown to do a good job of capturing some of the latent semantics of the input by locating semantically similar examples close to each other in the embedding space [16]. Indeed they have recently helped to transform many approaches to natural language processing. Similar ideas have been recently explored with financial time series data [4,5,20] and serve as state-of-the-art baselines. In what follows, we show how to learn case representations, by using the financial returns data of individual companies, and by mapping this high-dimensional raw time-series data into a low-dimensional embedding space. We do this by constructing a similarity-based representation of companies across several time periods and using matrix factorization techniques to compute a low-dimensional representation of these similarity patterns, which can then be used as the basis for our case representation.

3 An Embeddings-Based Case Representation

In this section, we describe the technical details of our approach to transforming raw time-series data into an embeddings-based representation. We will do this using a dataset of stock market *returns* data (see below) for $N = 611$ stocks spanning 2000-2018 [5]. Equivalently, we could use data for other types of finan-

cial assets, or more generally a variety of alternative time-series data from other domains. In our evaluation, we use daily, weekly, and monthly returns but the approach described is, in principle, agnostic to granularity.

3.1 From Raw Cases to Sub-Cases

We can consider each complete time-series as a *raw case* so that, for example, $c(a_i)$ corresponds to the full time-series for company/asset a_i as in Equation 1. Note that in this work the time-series provides so-called *returns* data rather than actual *pricing* data. The former refers to the relative movement in stock price over a given time period; for example, a return of 0.02 indicates that a price increased by 2% over a given time period whereas a return of -0.005 indicates that a stock price fell by 0.5% over a given time period. From this daily returns dataset, we can also aggregate to weekly or monthly returns in a straightforward manner by accumulating returns across longer periods.

$$c(a_i) = \{r_1^{a_i}, r_2^{a_i}, ..., r_T^{a_i}\} \tag{1}$$

The first step in our approach transforms these raw cases into a set of *sub-cases* such that $c(a_i, t, n)$ denotes the sub-sequence of n (the *look-back*) returns ending at time t, as shown in Equation 2. For example, later we consider a representation that is based on daily returns with a look-back of five (trading) days (one trading week), which is based on sub-cases with five returns ($n = 5$).

$$c(a_i, t, n) = \{r_{t-n+1}^{a_i}, r_{t-n+2}^{a_i}, ..., r_t^{a_i}\} \tag{2}$$

These sub-cases serve as useful and manageable sub-sequences of returns data for the purpose of similarity comparison during the next step.

3.2 Generating the Count Matrix

Thus, each company/asset can be transformed into a set of sub-cases and for each asset, look-back duration, and point in time there is a unique sub-case. Next, given a suitable similarity metric, we can produce a $N \times N$ matrix, $\mathcal{S}^{[t,n]}$ of pairwise similarities for any time t and look-back n, such that each element is given by $\mathcal{S}_{i,j}^{[t,n]} = sim\big(c(a_i, t, n), c(a_j, t, n)\big)$. Taking stock a_i as an example, we can then use $\mathcal{S}^{[t,n]}$ to identify the stock a_j which is most similar to a_i at time t by finding the column with the maximum value in row i of $\mathcal{S}^{[t,n]}$.

By repeating this procedure for every a_i and t we can count the number of times that every stock a_j appears as the most similar stock to a given a_i, across all time points. The result is a so-called *count matrix* $\mathcal{C}$ such that $\mathcal{C}_{i,j}$ denotes the number of time periods where stock a_j appeared as the most similar stock to a_i; see Equation 3.

$$\mathcal{C}_{i,j} = \sum_{\forall t} \delta \left(j \ , \ \arg\max_{\hat{j} \neq i} sim \left(c_{a_i,t}, c_{a_j,t} \right) \right) \tag{3}$$

where $\delta(i, j)$ is the Kronecker delta function as defined in equation 4.

$$\delta(i,j) = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases} \tag{4}$$

This approach to computing $\mathcal{C}$ can be viewed as a special case of a more general approach to computing $\mathcal{C}^{[k]}$. Since $\mathcal{C}$ is based on counts that come from the *single* most similar stocks, we can view it as $\mathcal{C}^{[k]}$ where $k = 1$. More generally then, $\mathcal{C}^{[k]}_{i,j}$ denotes the number of times where stock a_j appeared among the k most similar stocks to stock a_i. In other words, rather than limiting the count matrix to the single most similar stocks we can include a hyper-parameter k to accommodate a more *generous* counting function in order to encode information about a greater number of pairwise similarities. In fact, this is an important practical distinction as our preliminary studies found that representations based on higher values of k performed better during our evaluation. As such for the remainder of this work we will implicitly assume $k = 50$, which is the setting used during the evaluation in the next section; we will continue to refer to $\mathcal{C}^{[k]}$ as $\mathcal{C}$, without loss of generality.

In this way, $\mathcal{C}$ tells us about the most similar comparison stocks for a given stock over time. As the time-series data fluctuates to reflect complex, noisy, and unpredictable market changes, different stocks will appear among the *top-k* most similar stocks at different points in time and for different periods of time. We note that every value in $\mathcal{C}$ must be less than or equal to T, the number of time points in our raw data, and that cases are not compared to themselves, so the diagonal entries in $\mathcal{C}$ are fixed as 0.

As an aside, the count matrix may initially seem superfluous, with the temptation to aggregate similarity scores directly over time instead of using this more discrete approach. However, we found that direct aggregation yielded inferior representations evidenced by poorer downstream performance on financial classification tasks. We hypothesise that the benefit of the proposed approach stems from filtering out some of the noise inherent in market data that make for spurious raw similarities.

3.3 Generating Embedding Representations

We can use the count matrix to generate our final case representation by generating an embedding matrix $\mathcal{E} \in \mathbb{R}^{N \times d}$ (randomly initialised) where d is a hyperparameter to determine the dimensionality of the embedding, and N is the number of companies as before. If $\mathcal{E}_i \in \mathbb{R}^d$ denotes the i^{th} row of $\mathcal{E}$, which represents the embedding for stock a_i, then we can learn the $N \times d$ embedding matrix, $\mathcal{E}$, using matrix factorization techniques by minimising the loss function shown in Equation 5 with respect to $\mathcal{E}$. This is related to the problem of learning user and item embedding matrices (U and V respectively) from a rating matrix R, in recommender systems, by optimising for $R = UV^T$ [11], except that here we are producing a case embedding matrix ($\mathcal{E}$) based on $\mathcal{C} = \mathcal{E}\mathcal{E}^T$. However,

since we are only optimising a single matrix we must adjust the approach to exclude the diagonal entries of $\mathcal{C}$ from the optimisation.

$$\mathcal{L} = \sum_{i \in \{1,\ldots,N\}} \sum_{j \neq i} \left(\mathcal{C}_{i,j} - \mathcal{E}_i^T \mathcal{E}_j\right)^2 \tag{5}$$

To complete the process of learning case embeddings there are a number of routine adjustments that need to be made in order to deal with the type of overfitting and scaling problems that may occur due to the presence of outliers and skew within the distribution of values in the count matrix. First, to prevent the learned embeddings from overfitting to outliers, we define an upper bound for values in $\mathcal{C}$ as the 99.9^{th} percentile of off-diagonal elements in $\mathcal{C}$, clipping any values in $\mathcal{C}$ above this boundary to the boundary; this produces a clipped matrix which we refer to as $\tilde{\mathcal{C}}$. Second, to reduce skew in $\tilde{\mathcal{C}}$ we apply a standard log transformation in Equation 6. Finally, we apply min-max scaling to the resulting clipped and transformed count matrix, and regularization to the embedding vectors, which gives the final loss function as shown in Equation 7. In this final loss function. $\mu(\cdot)$ represents the min-max scaling of a matrix over all elements, $f(\cdot)$ represents the log transformation in Equation 6 and λ is the regularization rate, which takes a value of 0.1 in our later experiments.

$$f(\mathbf{x}) = \left(\frac{1}{2}\log(1+\mathbf{x})\right)^2 \tag{6}$$

$$\mathcal{L} = \sum_{i \in \{1,\ldots,N\}} \sum_{j \neq i} \left[\mu\left(f(\tilde{\mathcal{C}}_{i,j})\right) - \mathcal{E}_i^T \mathcal{E}_j\right]^2 + \lambda \cdot \left(||\mathcal{E}_i||^2 + ||\mathcal{E}_j||^2\right) \tag{7}$$

3.4 Discussion

In summary then, the above procedure transforms a raw, $(N \times T)$ times series dataset into a more compact ($N \times d$, where $d << T$) matrix of embeddings vectors. Each company case corresponds to a single d-dimensional embedding vector; that is, a row of $\mathcal{E}$ with its d feature values. This has the advantage of greatly reducing the dimensionality of our cases ($d << T$) but, in addition, we also hypothesise that the manner in which these embeddings have been produced means that they will capture more useful information than the raw returns data alone, or than more traditional summary features, by surfacing important temporal similarity information about the relationship between stocks.

It is worth noting too that this approach serves as a framework for generating case representations with different levels of granularity, context windows/look-back durations, and different similarity metrics. For example, it may be useful to focus on daily returns over a 5-day look-back period (sub-cases that correspond to single trading weeks) for one task and weekly returns over a 12-week look-back period for a different task. Or it may be useful to consider ways in which the resulting embedding representations can be combined to provide even richer representations. For example, the Orthogonal Procrustes Problem [22] offers a

robust solution for aligning embeddings produced by different models. Given two embedding spaces A and B, the objective is to find an orthogonal transformation matrix Ω, most closely mapping A to B. Mathematically, this can be expressed as the minimization problem $\arg\min_{\Omega} \|\Omega A - B\|_F$ subject to Ω^T. In principle, such an approach may facilitate combining case representations produced from different sub-cases but we leave this as a matter for future work.

4 Evaluation

So far, this paper has presented a novel approach for learning embedding-based case representations from financial time-series data, specifically the daily, weekly, and monthly returns data from stocks. We argue that this approach allows us to encode important temporal relationships between financial assets, which are otherwise difficult to capture in more traditional case representations (such as summary, raw feature-based or fixed, attribute-value style representations). In this section, we demonstrate the value of this new approach by evaluating the efficacy of these representations using several qualitative and quantitative techniques. In particular, we provide the results of a comparative evaluation of our embeddings-based representations versus more conventional CBR approaches, as well as recently proposed domain-specific methods [4,20], in a common financial domain classification task.

4.1 Dataset and Methodology

In this evaluation, we evaluate the performance of several approaches to industry sector classification using a real-world, publicly available dataset. This is a challenging classification task in its own right, which is instrumental to a multitude of downstream tasks in the financial domain [18].

Evaluation Dataset. As mentioned previously, the dataset used in this work is a publicly available dataset of *returns* data for 611 individual company stocks, spanning the years 2000-2018 [4]. Each stock is associated with a time-series of stock returns (relative changes in price) over daily, weekly, or monthly time periods. The dataset also contains additional (meta) data about each company stock, including industry sector classification data, which will be used in this evaluation. It is important to note that the industry sector labels in the dataset are not orthogonal (companies can operate across a number of sectors), and that they are assigned subjectively by analysts at the Global Industry Classification Scheme (GICS). As a result, in the classification task to come, perfect agreement with ground-truth labels is not a realistic goal, but high agreement serves as a strong indication that the representations are capturing useful information.

Industry Sector Classification Task. For this evaluation we will perform *industry sector classification*, which involves predicting a company's primary industry sector, based on their returns time-series data. This is a vital task for

many types of financial and economic analyses — identifying peers and competitors, quantifying market share and benchmarking company performance — none of which would be possible without sector classification schemes [18]; notably approximately 30% of publications in the top-three finance journals make use of industry classification schemes [27]. In this work, our primary focus is to use a CBR approach to classify stocks, using different representations (see below) to produce different case-base configurations. In each configuration, the problem description of a case corresponds to its returns data (whether using a raw, summary, or embeddings representation) and the solution part of a case corresponds to the stock's sector classification. Then, for a target/query stock a_q we identify its 5 nearest-neighbours, using a straightforward Euclidean or correlation metric (as given in Table 1), with simple *majority voting* to identify the predicted industry sector for a_q.

Algorithmic Configurations. In this evaluation we will test a number of different approaches, each distinguished according to the representation used for cases and the granularity of the returns data (daily, weekly, monthly) used. Arguably the simplest approach is to generate a feature-based representation based on summary features extracted from the raw returns data. These summary features include standard statistical features such as *mean, min, max, volatility, 25th percentile, median, 75th percentile* calculated over the daily, weekly and monthly returns data. These ($\times$3) configurations are referred to as *Summary* in what follows; see the first 3 rows in Table 1. We also implement two versions using the raw returns data as case representations (*Raw*) one set ($\times$3) uses a Euclidean distance metric (referred to as E in Table 1) when computing the k nearest-neighbours, and another ($\times$3) uses Pearson's correlation to identify the k nearest-neighbours (referred to as P in Table 1); the latter being a more common similarity metric to use in financial domains. Finally, we test several ($\times$18) varieties of our newly proposed embeddings-based representation (*Embedding*), with varying look-back durations for the daily, weekly, and monthly returns; the final three sections of Table 1. In particular, we vary the similarity metric used when *computing the count matrix* to look at the effect of using Euclidean distance (E) versus Pearson's correlation (P) versus the more recent hybrid metric (H), which combines Euclidean distance with a modified correlation component [6]. We note that for all of these embedding representations, the dimensionality is chosen as $d = 15$ and the standard Euclidean distance metric is used during the subsequent kNN classification (with the exception of raw with correlation) to enable a like-for-like comparison with the other baselines. In addition, we evaluate the proposed approach against non-CBR baselines including more general time series classification approaches [9,15,21], as well as domain-specific neural methods [4,20].

Evaluation Metrics. For each of these different variations, we use a standard 5-fold cross-validation to generate and test the classifications produced. For each

variation, we produce a standard *classification report*[3] which provides *precision* (the ratio of true positives to the sum of true and false positives), *recall* (ratio of true positives to the sum of true positives and false negatives), and F1 (the harmonic mean of precision and recall). The reported values are the weighted average for each class weighted by the number of samples in each class. There are 11 sector classes in the dataset: Basic Industries, Capital Goods, Consumer Durables, Consumer Non-Durables, Consumer Services, Energy, Finance, Health Care, Public Utilities, Technology, Transportation.

4.2 Results

The results are presented in Table 1. Each row corresponds to a specific algorithmic configuration and shows the representation used (*Summary*, *Raw*, and *Embeddings*), the granularity of the returns data (*Daily, Weekly, Monthly*) and the similarity metric used for the final kNN classification task (*Euclidean or Correlation*). In addition, for the *Embedding* representations, we also include settings for the relevant look-back periods. Finally, each configuration is associated with an overall weighted precision, recall, and F1 score as mentioned previously. Additionally, non-CBR baselines are reported in the lower section of the table with the same evaluation metrics.

A number of performance patterns are evident in these results. The poorest performances are associated with the *Summary* representations (F1$\leq$ 0.15). This is not surprising given that these representations abstract away a lot of the detail that exists in the returns data. While it may be possible to improve upon these representations, for example by including more domain-specific technical features, they provide a useful naive baseline against which to evaluate the improvements of more sophisticated approaches. The more reasonable *Raw* representations perform considerably better, with F1 values as high as 0.43 found among the variations that use a correlation-based similarity metric, arguably the most popular metric in the financial literature. In general, these *Raw* variations using correlation (*Raw+P*) out-perform the corresponding representations using Euclidean distance (*Raw+E*); the former report with $0.33 \leq F1 \leq 0.36$ compared to $0.41 \leq F1 \leq 0.43$ for the latter. Thus, the *Raw+P* variations serve as a useful baseline against which to evaluate the efficacy of the new embeddings-based representations.

Most of the embeddings-based representations outperform these *Raw+P* baselines, regardless of granularity or look-back duration. And, we note too that shorter look-back periods are associated with better performance than longer look-back periods. As further evidence that correlation-based similarity is more appropriate for financial returns data than Euclidean metrics, we note that the embeddings-based representations that are learned using correlation-based similarity (*Embedding + P* with $0.41 \leq F1 \leq 0.64$) out-perform the corresponding

[3] In this work all code is written in Python and uses the standard SciKit Learn implementation of kNN, cross-validation, and classification reporting. Non-CBR baselines were implemented in sktime where available, and PyTorch in other cases.

Table 1: Results for the case-based industry sector classification task for each of the 27 variations under consideration.

Representation	**kNN Metric**	**Granularity**	**Lookback**	**Precision**	**Recall**	**F1**
Summary	E	Daily	—	0.11	0.15	0.12
Summary	E	Weekly	—	0.13	0.15	0.13
Summary	E	Monthly	—	0.15	0.18	0.15
Raw	E	Daily	—	0.46	0.39	0.33
Raw	E	Weekly	—	0.45	0.43	0.36
Raw	E	Monthly	—	0.39	0.40	0.33
Raw	P	Daily	—	0.56	0.48	0.41
Raw	P	Weekly	—	0.50	0.49	0.42
Raw	P	Monthly	—	0.54	0.49	0.43
Embedding + E	E	Daily	5	0.67	0.62	0.63
Embedding + E	E	Daily	22	0.59	0.55	0.55
Embedding + E	E	Weekly	4	0.60	0.56	0.57
Embedding + E	E	Weekly	52	0.44	0.36	0.38
Embedding + E	E	Monthly	12	0.38	0.31	0.32
Embedding + E	E	Monthly	24	0.35	0.29	0.31
Embedding + P	E	Daily	5	0.68	0.62	0.64
Embedding + P	E	Daily	22	0.67	0.64	0.64
Embedding + P	E	Weekly	4	0.66	0.60	0.61
Embedding + P	E	Weekly	52	0.46	0.39	0.41
Embedding + P	E	Monthly	12	0.51	0.46	0.47
Embedding + P	E	Monthly	24	0.50	0.42	0.44
Embedding + H	E	Daily	5	**0.69**	**0.65**	**0.66**
Embedding + H	E	Daily	22	0.65	0.62	0.62
Embedding + H	E	Weekly	4	0.66	0.60	0.61
Embedding + H	E	Weekly	52	0.50	0.44	0.46
Embedding + H	E	Monthly	12	0.56	0.50	0.51
Embedding + H	E	Monthly	24	0.49	0.43	0.44

Non-CBR Methods	**Granularity**	**Precision**	**Recall**	**F1**
Shapelet Transform [9]	Daily	0.39	0.46	0.40
WEASEL [21]	Daily	0.50	0.47	0.47
Canonical Interval Forest [15]	Daily	0.57	0.56	0.52
Financial Time Series Embeddings [4]	Daily	0.62	0.60	0.60
Financial Correlation Graph Embeddings [20]	Daily	0.64	0.60	0.61

representations that were based on Euclidean distance (*Embedding + E* with $0.31 \leq F1 \leq 0.63$). Furthermore, the hybrid metric introduced by [6], which combines elements of Euclidean distance and correlation, tends to perform as well as, and usually better than, the embeddings-based representations using correlation alone (*Embeddings + H* with $0.44 \leq F1 \leq 0.66$). Indeed, the embed-

Table 2: Examples of top-3 nearest neighbours for given query stocks

Query Stock Sector - Industry	**3 Nearest Neighbours - Sector - Industry**	**Similarity**
JP Morgan Chase	Bank of America Corp - Finance - Major Bank	0.98
Finance	State Street Corp - Finance - Major Bank	0.98
Major Bank	Wells Fargo & Company - Finance - Major Bank	0.97
Microsoft	IBM - Technology - Computer Manufacturing	0.95
Technology	HP - Technology - Computer Manufacturing	0.93
Software	Adobe - Technology - Software	0.92
Walmart	Costco - Consumer Services - Dept Store	0.89
Consumer Services	Kroger - Consumer Services - Food Chains	0.82
Department Store	McDonalds - Consumer Servies - Food Chains	0.78

dings produced with this hybrid metric always outperform the $Raw + P$ baseline regardless of granularity and look-back.

The domain agnostic non-CBR methods proposed in [21,15] outperform the raw and summary baselines, while the recent task-specific approaches [20,4] are stronger again ($0.60 \leq F1 \leq 0.61$). However, the proposed approach remains the strongest performer.

4.3 Discussion

These results support the hypothesis that the proposed embeddings-based representations are capable of capturing more useful information from the time-series returns data than more conventional representations. The best proposed representation is associated with an F1 score of 0.66 compared to just 0.43 for the best CBR baseline representation and 0.60/0.61 for the task specific baselines. Moreover, the proposed representation is well-suited for use in a CBR setting which, due to the retrieval of existing labelled cases, offers further advantages when it comes to interpretability.

By way of further explanation, Table 2 shows some examples of the nearest neighbours that are identified for 3 different query companies (JP Morgan Chase, Microsoft and Walmart) using an embeddings-based representation. For each query company, we summarise the top-3 nearest neighbours, the sector class (e.g. Finance), a finer-grained industry label (e.g. Major Bank), and their corresponding similarities to the query stock. The results align with our intuitions and in each case, the nearest neighbours match the query's industry sector (Finance, Technology, and Consumer Services, respectively).

As another example, Figure 1 shows a 2D visualisation of the clusters of companies that emerge when using the embeddings-based representations. Each node corresponds to an individual stock and an edge is created between two stocks if their similarity exceeds some minimum threshold (0.75 in this example). Then, a force-directed graph drawing algorithm [10] is used to position the nodes in such a way as to optimise their placement in the resulting similarity

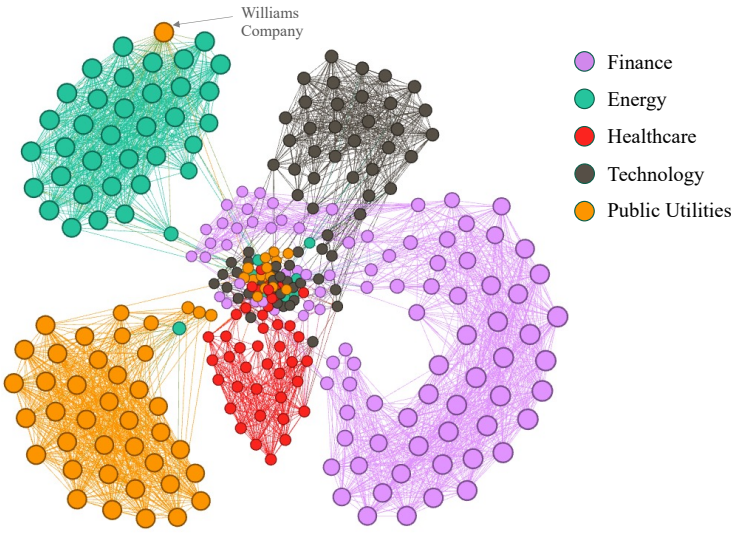

Fig. 1: Visualization of embedding clustering. A subset of sectors is used for visual clarity.

space. The nodes have been colour-coded based on their ground-truth industry sectors and we can see clearly how nodes from the same industry sector tend to be clustered together, indicating that the embeddings-based representations are doing a good job of capturing this relationship; as an aside it is worth noting that the embedding representations also exhibited clear clustering using visualisation approaches such as PCA and t-SNE.

We also observe some interesting patterns in the graph that are not immediately obvious from the sector labels. For example, a node from the Public Utilities sector (indicated in orange) appears as an outlier in the Energy cluster (green). This node, highlighted in Figure 1, is an energy supply company called Williams Company, whose primary business is natural gas processing and transportation. Thus, although it has Public Utilities classification in our dataset, the case representation facilitates recognising its similarity with the Energy business and it is positioned accordingly.

Visualisations such as this are powerful tools for technical analysts to better understand the evolving structure of modern markets, but to be useful they must rely on representations that are capable of reasonably accurately capturing meaningful relationships between different stocks and companies. The case representation proposed here should help to improve the utility of such tools because it does a better job at recognising the relationships that exist between companies but that may be obscured by the raw time-series data and not captured by traditional subjective industry classification schemes.

5 Conclusion

Using CBR with time-series data presents a number of challenges, not the least of which is how to generate case representations that are capable of capturing the complex temporal behaviour of the underlying data. Time-series data are becoming more and more common in the modern world with the increasing ability to capture and store large amounts of real-time, real-world data. This is especially true in the financial domain and this work, we have described the development of a novel representation of financial time-series data that is well suited to CBR. We have demonstrated the effectiveness of this representational approach on the important task of industry sector classification, in comparison to several CBR baselines as well as recent task-specific neural methods. The results indicate that the proposed approach offers performance benefits compared to these alternatives.

There are several opportunities for additional work arising from this initial study. For example, no comprehensive hyper-parameter tuning has been carried out for the proposed representations and it is likely that by varying key parameters, such as the embedding dimensionality (d), k, and λ, further improvements could be found; the fact that significant improvements were obtained for the "default" settings used here speaks to this. And, although the focus of this work has been on the use of the proposed representation in a CBR context, the representation should be equally applicable as a training data representation for other machine learning models. In fact, preliminary results, not provided here for reasons of space and clarity, suggest further performance gains if the embeddings are applied within a non-CBR classifier.

Within the financial domain, there are many other tasks that can be explored as targets for this type of representation. For example, risk management and portfolio optimisation [4] are obvious candidates in this regard. Moreover, given the proliferation of time-series data across many domains (clinical health [2], exercise and fitness [7] etc.) it will be interesting to assess whether this type of representation can add value across different task types.

Acknowledgements: This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183.

References

1. Chun, S.H., Ko, Y.W.: Geometric case based reasoning for stock market prediction. Sustainability (Switzerland) **12** (9 2020). https://doi.org/10.3390/su12177124
2. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. In: ICCBR 2021, Salamanca, Spain, September 13–16, 2021, Proceedings 29. pp. 32–47. Springer (2021)
3. Delaney, E., Greene, D., Shalloo, L., Lynch, M., Keane, M.T.: Forecasting for sustainable dairy produce: Enhanced long-term, milk-supply forecasting using k-nn for data augmentation, with prefactual explanations for xai. In: ICCBR 2022, Nancy, France, September 12–15, 2022, Proceedings. pp. 365–379. Springer (2022)

4. Dolphin, R., Smyth, B., Dong, R.: Stock embeddings: Learning distributed representations for financial assets. arXiv preprint arXiv:2202.08968 (2022)
5. Dolphin, R., Smyth, B., Dong, R.: A machine learning approach to industry classification in financial markets. In: Artificial Intelligence and Cognitive Science: 30th Irish Conference, AICS 2022, Munster, Ireland, December 8–9, 2022, Revised Selected Papers. pp. 81–94. Springer (2023)
6. Dolphin, R., Smyth, B., Xu, Y., Dong, R.: Measuring financial time series similarity with a view to identifying profitable stock market opportunities. In: ICCBR 2021, Salamanca, Spain, September 13–16, 2021, Proceedings 29. pp. 64–78. Springer (2021)
7. Feely, C., Caulfield, B., Lawlor, A., Smyth, B.: Using case-based reasoning to predict marathon performance and recommend tailored training plans. In: ICCBR 2020, Salamanca, Spain, Proceedings 28. pp. 67–81. Springer (2020)
8. Gold, S.: The viability of six popular technical analysis trading rules in determining effective buy and sell signals: Macd, aroon, rsi, so, obv, and adl. Journal of Applied Financial Research **2** (2015)
9. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data mining and knowledge discovery **28**, 851–881 (2014)
10. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. PloS one **9**(6), e98679 (2014)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
12. Kumar, G., Jain, S., Singh, U.P.: Stock market forecasting using computational intelligence: A survey. Archives of Computational Methods in Engineering pp. 1–33 (2020)
13. Li, S.T., Ho, H.F.: Predicting financial activity with evolutionary fuzzy case-based reasoning. Expert Systems with Applications **36**(1), 411–422 (2009)
14. McSherry, D.: A lazy learning approach to explaining case-based reasoning solutions. In: ICCBR. pp. 241–254. Springer (2012)
15. Middlehurst, M., Large, J., Bagnall, A.: The canonical interval forest (cif) classifier for time series classification. In: 2020 IEEE international conference on big data (big data). pp. 188–195. IEEE (2020)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Nalmpantis, C., Vrakas, D.: Signal2vec: Time series embedding representation. In: Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019. Springer (2019)
18. Phillips, R.L., Ormsby, R.: Industry classification schemes: An analysis and review. Journal of Business & Finance Librarianship **21**(1), 1–25 (2016)
19. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing **26**(1), 43–49 (1978)
20. Sarmah, B., Nair, N., Mehta, D., Pasquali, S.: Learning embedded representation of the stock correlation matrix using graph machine learning. arXiv preprint arXiv:2207.07183 (2022)
21. Schäfer, P., Leser, U.: Fast and accurate time series classification with weasel. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 637–646 (2017)

22. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966)
23. Slade, S.: Case-based reasoning for financial decision making. In: Proceedings of the First International Conference on Artificial Intelligence Applications on Wall Street, New York, NY. IEEE Computer Society (1991)
24. Smyth, B., Cunningham, P.: A comparison of incremental case-based reasoning and inductive learning. In: Advances in Case-Based Reasoning: Second European Workshop, EWCBR-94 Chantilly, France, November 7–10, 1994 Selected Papers 2. pp. 151–164. Springer (1995)
25. Wang, Y., Wang, Y.: A case-based reasoning-decision tree hybrid system for stock selection. International Journal of Computer and Information Engineering **10**(6), 1223–1229 (2016)
26. Warren, G., Smyth, B., Keane, M.T.: "better" counterfactuals, ones people can understand: Psychologically-plausible case-based counterfactuals using categorical features for explainable ai (xai). In: ICCBR 2022, Nancy, France, September 12–15, 2022, Proceedings. pp. 63–78. Springer (2022)
27. Weiner, C.: The impact of industry classification schemes on financial research. Available at SSRN 871173 (2005)