Designing an investment portfolio for public companies in the life sciences

Rosa X Ma
Stanford University
Department of Genetics
rosaxma@stanford.edu

Vandon T Duong
Stanford University
Department of Bioengineering
vandon@stanford.edu

1 Abstract

Stocks of public companies in the biotech sector tend to be extremely volatile and largely dissociated from traditional fundamentals (e.g. EPS, P/E ratio). There are numerous, obscure factors that drive the valuation of companies in the life sciences. We propose a deep learning-based investment strategy, which includes diversification through clustering and portfolio construction by selecting stocks in each cluster based on its Sharpe ratio. Specifically, our clustering model has two phases: (1) parameter initialization with a deep convolutional autoencoder and (2) parameter optimization and clustering. In the second phase, we minimize the Kullback-Leibler (KL) divergence between the auxiliary target distribution and the Student's t-distribution. Experimental results show the diversification accomplished by our clustering method suggested portfolios that outperform multiple indexes during a downturn in the biotech sector.

2 Introduction

Compared with any other industry, the price movements of drug developers are uniquely influenced by clinical catalysts (e.g. clinical trial results, regulatory approval/denial, reimbursement, etc.) [1]. This makes investing in biotech very risky. Yet, the biotech sector often outperforms other industries and the overall market. It may be possible to devise an efficient strategy guided by machine learning for trading stocks in the life sciences. One important way to mitigate risk in investing is to diversify the holdings in a portfolio [2]. Clustering techniques can be employed to help pick stocks in a way that ensures diversification. We propose a two-part approach: (1) generate clusters of public companies to identify expected/unexpected relationships, and (2) building and testing investment portfolios on the basis of cluster analysis. Our algorithm clustered biotech companies based on trading data only, as opposed to information required by due diligence like therapeutic pipeline, stage of clinical trials, the competitive landscape, etc. This unsupervised method enables a simple investment approach: the Sharpe ratio of each stock was calculated and the top 1, 3, and 5 holdings per cluster (HpCs) were selected for the long-equity, equal-allocation portfolios. Ideally, selecting stocks across the clusters would diversify the holdings and lower investment risk. We trained our algorithm during a period of significant gains for the biotech sector and tested the clustering results during a subsequent period of market retraction. We compared the performance of our portfolios to the general market ($SPY) and several biotech-specific indexes ($XBI, $IBB, $UBIO, $BBC, $CNCR). Using this approach, we also compared how clustering would be altered by feature selection: (1) close price, (2) close price, and volume, and (3) close, open, low, high prices, and volume. From the literature, most investment algorithms focus on close price only.

3 Related work

Developing robust methods to cluster stocks is an active research area in quantitative finance. Popular clustering techniques, including K-means, are not suitable for analyzing time-series data (e.g. stock price). The distance metrics used in these clustering techniques are limited to the original data space and tend to be ineffective in cases of high dimensionality. Variants like kernel K-means yield improved cluster quality [3-4], but do not address the “curse of high dimensionality” and fixed kernel functions limit nonlinear transformations. Dimensionality reduction methods such as principal components analysis (PCA) are not applicable to our input data (i.e. daily changes in price or volume) because its dimensionality is greater than the number of samples (i.e. public biotech companies). Artificial neural networks (ANNs) are capable of generalizing underlying trends where most conventional methods fail [5]. Artificial recurrent neural networks (RNNs) such as long short-term memory (LSTM) have been applied to portfolio optimization. LSTM networks outperform approaches like random forest and logistic regression in predicting stock price movement [6-7], indicating that deep learning models can adequately capture market patterns. However, simply using ANNs for feature extraction requires labeling data based on predefined criteria. Additionally, RNNs are time-consuming to train and make the strong assumption that every time point in the time series depends on all the previous time points. Historical trajectory in stock prices is not necessarily indicative of future performance, especially in the biotech sector where valuation is often driven by clinical catalysts and not fundamentals. Therefore, we are interested in exploring an approach which does make strong assumptions about the mutual dependence of stock price, yet does not require prior knowledge and data labeling.

Convolutional neural networks (CNN) can outperform multilayer perceptron (MLP) and LSTM in identifying patterns in stock movements [8]. To address the problems of high dimensionality and nonlinearity posed by time-series data, 4-channel (low, high, open, and close prices) market data can be converted to candlestick charts to present price history as images [9]. This study employed convolutional auto-encoder for feature extraction and network modularity method for clustering the learnt features. However, in this approach the feature extraction is independent of and cannot learn from the clustering. Deep Embedded clustering (DEC) is a recently developed method to simultaneously optimize feature representation and clustering [10]. DEC is less sensitive to the choice of hyperparameters compared to state-of-the-art methods, which is useful where cross-validation is not possible (e.g. our unsupervised learning algorithm). To achieve optimal clustering of biotech stocks, we built a convolutional auto-encoder to extract market patterns and applied DEC to perform clustering and further fine-tuning of feature representation.

4 Dataset and features

A list of 267 stock tickers from popular biotechnology exchange-traded funds (ETFs) and news reportings were assembled. The IEX Finance python library was used to scrape close, open, high, low prices and volume from 05/25/2016 to 06/07/2019 corresponding to this list. We used data from 05/25/2016 to 05/20/2018 (i.e. 500 trading days) for clustering and 05/21/2018 to 06/07/2019 for portfolio evaluation. To normalize the dataset, we calculated the log differences between each day. This is effectively the daily percentage return in price or the daily percentage change in volume.

$$\% = \log_{10}(x_i) - \log_{10}(x_{i-1}) \ ,$$

An evaluation of the dataset showed a steady increase in the number of public biotech companies (i.e. more biotech IPOs), and significant fluctuations in price and volume (Fig. 1). Specifically, we observed numerous events in which the daily return (or loss) was over 50% (i.e. high volatility in the biotech sector). For our study, we evaluated different clustering techniques using data from 500 consecutive trading days and 226 companies. We removed 41 stocks that were missing more than half of the data in this range of dates.

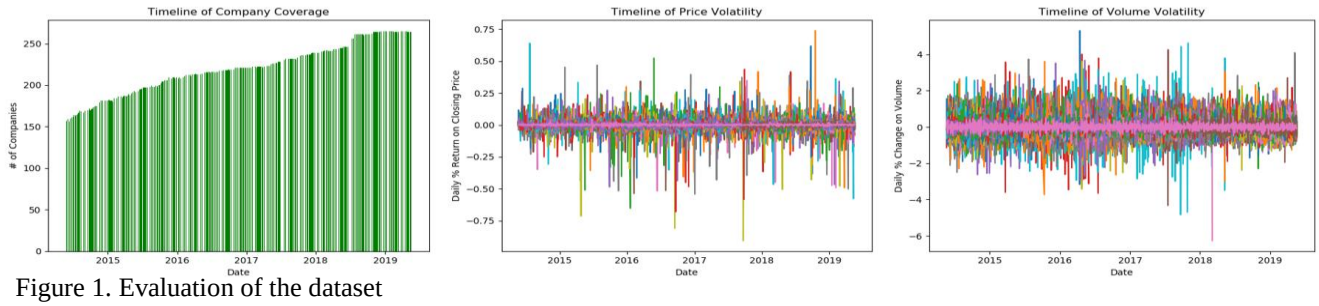

Figure 1. Evaluation of the dataset

5 Methods

Our goal is to cluster a set of n points (stocks) $\{x_i \in X\}_{i=1}^{n}$ into k clusters, each represented by a centroids μ_j, j = 1,..,k. As a baseline for comparison, we first performed K-means clustering initialized by K-means++ [11]. To decide the number of clusters, we used the elbow method to calculate the sum squared error (SSE) for a range of different clusters [12]. However, the curve was not informative as there was no observable elbow (Fig.2). As an alternative way to decide the number of clusters and initialize the centroids, we performed hierarchical clustering (Fig.2) using the Ward's method to calculate the linkage [13].

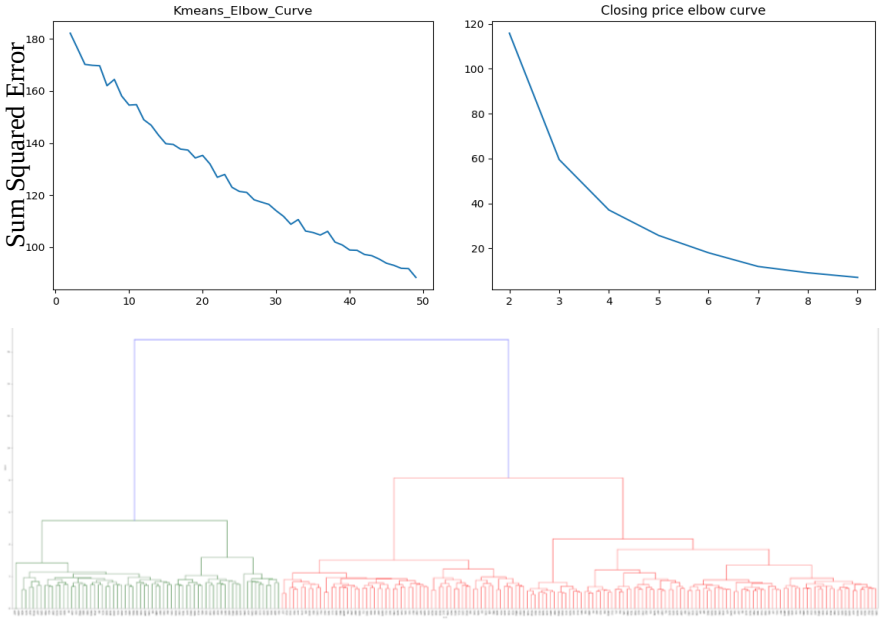

Figure 2. Clustering analyses. Top left: Elbow curve of naive K-means clustering. Top right: Elbow curve of Deep Embedded Clustering (DEC). Bottom: Dendrogram for hierarchical clustering.

In addition to clustering directly in the data space X, we mapped the input from X to a latent feature space using non-linear mapping f_θ, where θ are learnable parameters. Therefore, our algorithm has two phases: (1) parameter initialization with a deep convolutional autoencoder, and (2) parameter optimization and clustering. In the second phase, we minimize the Kullback-Leibler (KL) divergence between the auxiliary target distribution and the Student's t-distribution.

To identify patterns in the time-series input, we pre-trained a symmetric convolutional deep autoencoder (Fig 3.). The weights were initialized with Glorot uniform initializer. The minibatch size was set to 128 and the model trained at a learning rate of 0.0005 using the Adam optimizer. The model would stop training when the change in difference was less than $5e^{-7}$. The encoder learned f_θ that maps $x_i \in X$ to $z_i \in Z$. After the pre-training, the decoder layers were discarded and 10 features were extracted from the last layer (i.e., bottleneck layer) as a representation of our high dimensional input data (phase 1). To perform clustering, a custom clustering layer that performs soft assignment using the Student's t-distribution was built on top of the bottleneck layer:

$$q_{ij} = \frac{(1+\frac{\|z_i-\mu_j\|^2}{\alpha})^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1+\frac{\|z_i-\mu_{j'}\|^2}{\alpha})^{-\frac{\alpha+1}{2}}} \ ,$$

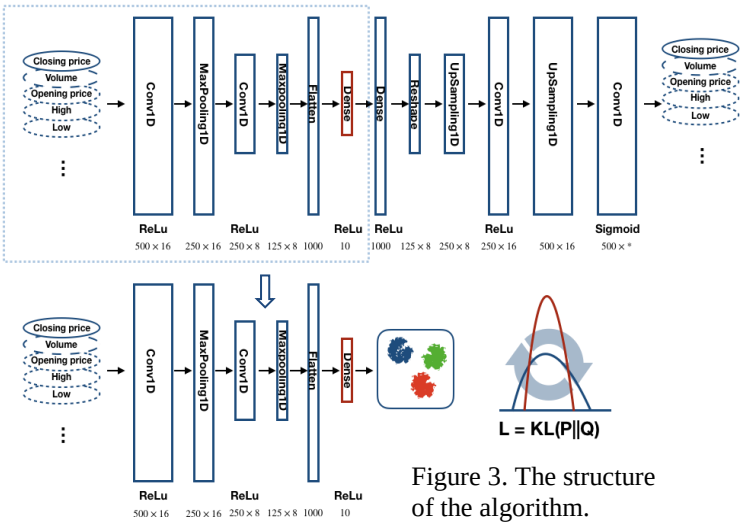

Figure 3. The structure of the algorithm.

where $z_i = f_\theta(x_i) \in Z$ corresponds to $x_i \in X$ after embedding, f_θ is the initial estimate of the non-linear mapping from phase 1, α are the degrees of freedom of the Student's t-distribution, μ_j represents the centroids, and q_{ij} is the probability of assigning sample i to cluster j. We set α = 1, since we cannot cross-validate α in the unsupervised setting. The k centroids

$\{\mu_i\}_{j=1}^{k}$were initialized by performing K-means clustering on the features from the bottleneck layer with 20 restarts and selecting the best solution. The number of clusters was determined by the elbow method.

To improve the clustering assignment and feature representation simultaneously. [1] defined the auxiliary target distribution (hereinafter referred to as target distribution):

$$p_{ij} = \frac{\frac{q_{ij}^2}{f_j}}{\sum_{j'} \frac{q_{ij}^2}{f_{j'}}} ,$$

where $f_j = \sum_i q_{ij}$ are soft cluster frequencies. The auxiliary target distribution was designed to put more emphasis on data points assigned with high confidence, and normalize loss contribution of each centroid to prevent large clusters from distorting the hidden feature space. The objective is defined as a KL divergence loss between the soft assignments q_i and the auxiliary distribution p_i as follows:

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \, log \, \frac{p_{ij}}{q_{ij}} \quad ,$$

With this target distribution we were able to iteratively refine the clusters by learning from the high confidence assignments, as determined by the target distribution. In our implementation, the batch size is 128, the target distribution is updated every 250 iterations, and the entire model was trained to minimize the KL divergence loss between the target distribution and the clustering output. The convergence threshold was set at 0.1%.

To construct the long-equity, equal-allocation portfolios, we calculated the Sharpe ratio of each individual stock during the training period [14]:

$$SR = \frac{E[R_a - R_f]}{\sigma_e} = \frac{\bar{R}_e}{\sigma_e} \quad ,$$

where R_a is the asset return, $R_f$is the risk free rate, $\underline{R}_e$is the mean excess return, and $\sigma_e$is the standard deviation of excess return. The $XBI, a popular ETF which tracks a modified equal weighted index of the biotechnology industry, was used as the risk free rate. The Sharpe ratios of stocks in each cluster were ranked. The top (1, 3, 5) performing stocks from each cluster were placed into portfolios at equal allocations. These portfolios were termed 1, 3, and 5 holdings per cluster (HpC). Clusters with fewer than 5 stocks were not included in the allocation.

We then tracked the total return (TR) of each portfolio during both training and testing periods:

$$TR = \frac{(C-P)}{P} \quad ,$$

where C is the fund after d days of investment and P is the principal. We also tracked the internal rate of return (IRR) of each portfolio:

$$IRR = (TR + 1)^{365/d} - 1 \quad ,$$

For comparison, we calculated the total return and IRR of the general market and biotech-specific indexes ($SPY, $XBI, $IBB, $UBIO, $BBC, $CNCR).

6 Experiments, Results, and Discussion

We first applied our two-phase algorithm to only the close prices of the 226 companies from the 500-day period. For the training of the convolutional autoencoder (CAE), the weight was initialized using Glorot uniform initializer, and the model was trained using Adam optimizer, inspired by [1]. Batch size, learning rate, and early stopping criterion were empirically determined. After the training (Fig 4.), the features from the last layer of the autocoder (bottleneck layer) were extracted to cluster by K-means. 5 clusters were chosen based on the elbow method (Fig. 4). Initially, the target distribution was set to update every 50 iterations. However, the clustering results tend to be unstable (data not shown). To stabilize the clustering, we gradually increased the iterations between each update because less frequent updates may prevent the target distribution from being affected by fluctuations in parameter fine-tuning. Eventually, we found that updating every 250 iterations gives relatively stable clusters. The simulations take at least 1000 iterations to converge, indicating that the target distributions are updated throughout the training. In figure 5, we visualize the embedded representation of the input before and after parameter optimization by applying PCA to the embedded points z_i. It is clear that the clusters became well-separated after DEC.

Having obtained satisfactory results from only using the close prices as input, to take advantage of the versatility of deep learning, we modified the CAE to enable it to take 2-channel time-series (close price and volume), and 5-channel time-series (low, high, open, close prices and volume) as input. All other hyper-parameters remained the same as in the case of close price only, which was supported by the training loss plot and elbow function (Fig. 4). To better understand the contribution of each component of our model

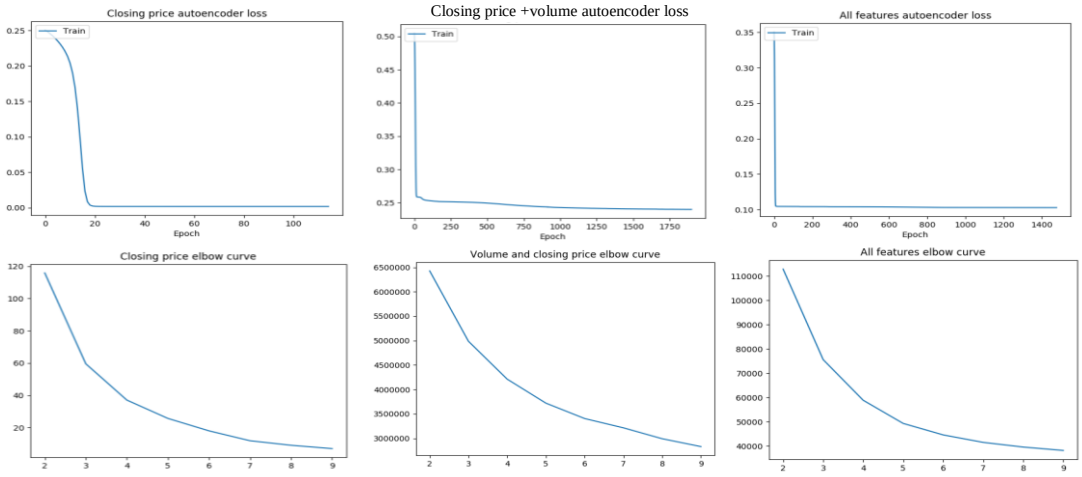

Figure 4. Top panel: Training losses of autoencoders using three different inputs. (left: close price only. Middle: Close price and volume. Left: All five features); Bottom panel: Elbow curves of Kmeans clusterings on features extracted from the bottleneck layer. (left: close price only. Middle: Close price and volume. Left: All five features)

(i.e. the encoder and the DEC layer), we also included the results of K-means clustering on features extracted from the bottleneck layer before parameter optimization in our downstream analyses.

To compare the clustering from the autoencoder + DEC model against the clustering from the baseline (e.g. K-means and hierarchical clustering), we backtracked to set the minimal distance at 4 and obtained 5 clusters from hierarchical clustering. In turn, the cluster centroids were used to initialize K-means. Only the close price was used as input in the baseline experiments. The clustering from both hierarchical and K-means were included in the downstream analyses.

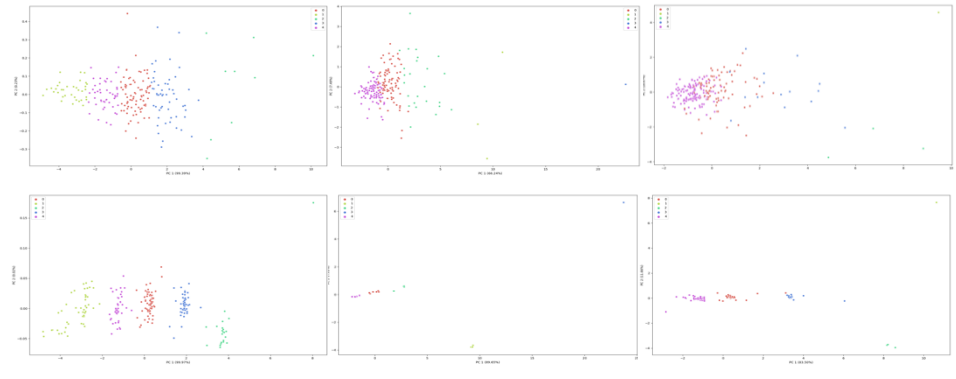

Figure 5. PCA plots of features extracted from the bottleneck layer of the autoencoder. Top: features before parameter optimization. Bottom: features after parameter optimization (left: close price only. Middle: Close price and volume. Left: All five features)

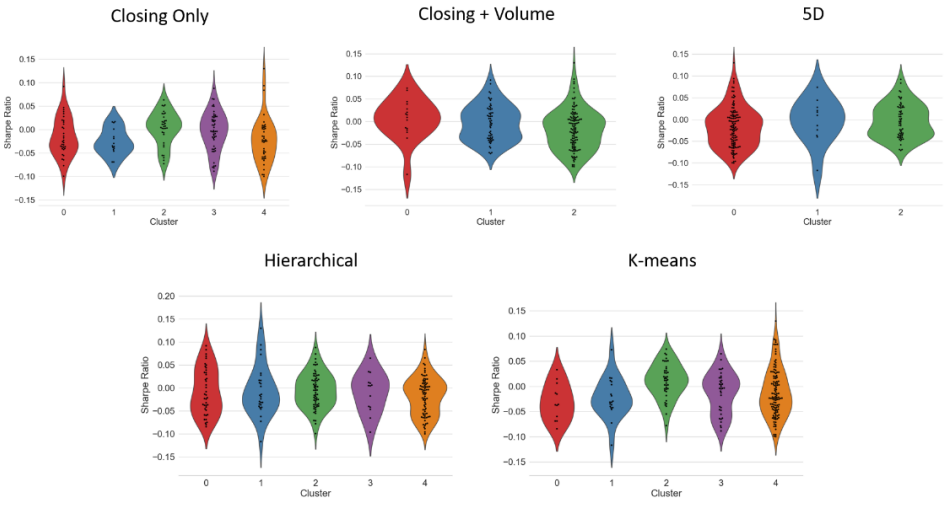

Figure 6. Risk-adjusted performance by cluster

Violin plots of the risk-adjusted performances (i.e. Sharpe ratios) of the clusters were generated from each model during the training period (Figure 6). The y-axis represents Sharpe ratio and each dot represents a public company. The clustering of close price and volume was similar to the clustering of close, open, low, high prices and volume, also seen by the visualization below. These models also resulted in two clusters with fewer than 5 companies, so were not included for the portfolio construction. The companies in the three large clusters (shown) and two small clusters (not shown) were fairly consistent, suggesting redundancy in the feature selection. This makes sense as the inflections in open, close, low, and high prices are likely mutually dependent. The stocks in the model for close price only were more uniformly distributed among clusters. Furthermore, the clustering of the close only model was significantly different from that of close price and volume and that of close, open, low, and high prices and volume.

We observed that stocks for gene editing companies (e.g. $CRSP, $EDIT, $NTLA) tend to cluster together. This was expected since each of these companies rely on the same underlying technology, CRISPR-Cas9 [15]. Until recently, these CRISPR-Cas9 gene editing companies were preclinical and highly susceptible to stock price fluctuations by scientific papers and preprints. Notably, another gene editing company ($SGMO) based on the zinc finger technology did not cluster with the CRISPR-cas9 companies. Not only are the

underlying technologies different, but $SGMO is further in clinical trials than $CRSP, $EDIT, and $NTLA and targeting other indications.

Interestingly, $AXON and $AXGT always clustered together. These are indeed the same company, which underwent a corporate rebranding (Axovant Sciences to Axovant Gene Therapy) after failing multiple clinical trials. The company was unsuccessful with repurposing small molecule drugs for treatment of dementia, and now is pursuing gene therapy for other neurological and neuromuscular diseases. The clustering of $AXON and $AXGT confirms that the algorithm is able to pair stocks that are essentially the same despite no overlap in time-series data. This is likely because the influence of $AXON and $AXGT on other stocks in the clustering is appropriately captured. We also noticed that companies of large market cap and with clinically-advanced or approved drugs were clustered together. For example, $ABBV, $AMGN, $CELG, $GILD, $RGEN, $PFE, and others. These large pharmaceutical companies tend to be affected by macrotrends and political rhetoric than small biotech startups. They also actually have revenue and fundamentals are more reliable compared to small, early-stage companies. Interestingly, $ILMN, a sequencing service monopoly, is also grouped with this cluster.

The total return and IRR of indexes and investment portfolios for the training and testing periods are shown in Figure 7. ETFs tracking general market and biotech-specific indexes were used as benchmarks for portfolio evaluation. Compared to the general market ($SPY), the biotech sector ($XBI, $IBB, $UBIO, $BBC, and $CNCR) had a strong performance during the training period but significant retraction in the testing period. Baseline clustering (hierarchical and K-means) tend to overfit to the training period. Using the autoencoder with DEC has a robust portfolio performance across the different clustering models. Clustering based on close price only is the most effective in mitigating losses. The model based on both close price and volume, and the model based on all features tend to overfit to the training period compared with the model based only on close price.

Training: 2016.05.25 - 2018.05.20

Indexes	Return	IRR
SPY	14%	7%
XBI	24%	11%
IBB	7%	4%
UBIO	1%	1%
BBC	25%	12%
CNCR	8%	4%

Close Only	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	116%	48%	118%	48%
3 HpC	96%	40%	95%	40%
5 HpC	78%	34%	68%	30%

Close Volume	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	130%	52%	103%	43%
3 HpC	135%	54%	104%	43%
5 HpC	118%	48%	95%	40%

Baseline	Hierarchical Return	Hierarchical IRR	K-means Return	K-means IRR
1 HpC	120%	49%	125%	50%
3 HpC	108%	44%	92%	39%
5 HpC	100%	42%	76%	33%

5D	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	130%	52%	103%	43%
3 HpC	116%	47%	104%	43%
5 HpC	106%	44%	88%	37%

Testing: 2018.05.21 - 2019.06.07

Indexes	Return	IRR
SPY	3%	3%
XBI	-6%	-6%
IBB	-2%	-2%
UBIO	-18%	-17%
BBC	-10%	-9%
CNCR	-14%	-14%

Close Only	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	-9%	-9%	-12%	-11%
3 HpC	-12%	-12%	-5%	-5%
5 HpC	-5%	-5%	-14%	-14%

Close Volume	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	-15%	-14%	-24%	-23%
3 HpC	-10%	-10%	-8%	-8%
5 HpC	-8%	-8%	-10%	-10%

Baseline	Hierarchical Return	Hierarchical IRR	K-means Return	K-means IRR
1 HpC	-13%	-12%	-18%	-17%
3 HpC	-10%	-10%	-12%	-11%
5 HpC	-9%	-8%	-11%	-10%

5D	Encoder + DEC Return	Encoder + DEC IRR	Encoder + K-means Return	Encoder + K-means IRR
1 HpC	-14%	-14%	-24%	-23%
3 HpC	-10%	-10%	-8%	-8%
5 HpC	-6%	-6%	-10%	-10%

Figure 7. Compilation of the returns and IRRs for indexes and designed portfolios.

7 Conclusion and Future Work

We proposed a deep learning-based investment strategy. The clustering algorithm first maps the input stock data from the high-dimensional data space to a low-dimensional feature space using an autoencoder. It then uses Deep Embedded Clustering (DEC) to cluster the stocks in the feature space. The portfolios constructed according to the diversification accomplished by our clustering method outperform multiple indexes during a downturn in the biotech sector.

However, one important limitation of our method is that it requires pre-specification of the number of clusters, and initial node selection. Methods that do not require a specified number of clusters such as modularity optimization [16] could be incorporated into the clustering layer of the model.

In the future, we will compare our stock selection method (i.e. uniformly picking high performing stocks across clusters) with random selection. Ideally, this should show that our approach is better than random selection. We will also investigate the performance of portfolios created with non-uniform selection among the clusters.

We are also interested in interpreting our models. Specifically, we want to understand what did our models learn from the data; especially, what do the 10 features from the bottleneck layer represent? The knowledge gained from the interpretation can inform our understanding of the biotech stock market. Furthermore, we will evaluate the distribution of stocks among clusters in terms of therapeutic area (e.g. oncology, neurodegeneration, allergy, etc.), stage of clinical development (e.g. preclinical, IND-enabling, phase of clinical study, commercialization), size of market cap (e.g. micro, small, large, etc.), and partnerships with big pharma. This will offer insight into how the clustering is distributed and what market and other atypical, biotech-specific attributes could be related to stock price trajectories. The model can be applied to other sectors such as automobile, energy, and real estates or even the general market. We expect the model to have similar performance in these markets.

Our scripts are available at: https://github.com/vandontduong/cs229-biotech-stocks-machine-learning.

8 Contributions

Rosa researched and decided on the clustering algorithms to use, as well as conducted the cluster analysis. Vandon compiled the list of biotech stocks, scraped and preprocessed the stock data, identified behaviors in the generated clusters, and conducted the construction and evaluation of investment portfolios. We worked together in developing the overall strategy, assessing the algorithm performance and results, and writing the paper.

9 References

1. R. T. Thakor, N. Anaya, Y. Zhang, C. Vilanilam, K. W. Siah, C. H. Wong, and A. W. Lo, “Just how good an investment is the biopharmaceutical sector?,” Nature Biotechnology, vol. 35, no. 12, pp. 1149–1157, 2017.
2. W. N. Goetzmann and A. Kumar, “Equity Portfolio Diversification*,” Review of Finance, vol. 12, no. 3, pp. 433–463, 2008.
3. T. Datta and I. Ghosh, “Using Clustering Method to Understand Indian Stock Market Volatility,” Communications on Applied Electronics, vol. 2, no. 6, pp. 35–44, 2015.
4. R. Chitta, R. Jin, and A. K. Jain, “Stream Clustering: Efficient Kernel-Based Approximation Using Importance Sampling,” 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015.
5. G. Zhang, B. E. Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” 1998.
6. T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, 2018.
7. T. Gao and Y. Chai, “Improving Stock Closing Price Prediction Using Recurrent Neural Network and Technical Indicators,” Neural Comput., vol. 30, no. 10, pp. 2833–2854, Oct. 2018.
8. R. Chitta, R. Jin, and A. K. Jain, “Stream Clustering: Efficient Kernel-Based Approximation Using Importance Sampling,” 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015.
9. G. Hu, Y. Hu, K. Yang, Z. Yu, F. Sung, Z. Zhang, F. Xie, J. Liu, N. Robertson, T. Hospedales, and Q. Miemie, “Deep Stock Representation Learning: From Candlestick Charts to Investment Decisions,” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.
10. J. Xie, R. Girshick, and A. Farhadi. "Unsupervised deep embedding for clustering analysis," International conference on machine learning, 2016.
11. D. Arthur, and S. Vassilvitskii. "k-means++: The advantages of careful seeding," Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.
12. T. M. Kodinariya, and P. R. Makwana. "Review on determining number of Cluster in K-Means Clustering," International Journal 1.6, pp. 90-95, 2013.
13. J. H. Ward, “Hierarchical Grouping to Optimize an Objective Function,” Journal of the American Statistical Association, vol. 58, no. 301, p. 236, 1963.
14. W. F. Sharpe, “Mutual Fund Performance,” The Journal of Business, vol. 39, no. S1, p. 119, 1966.
15. J. L. Contreras and J. S. Sherkow, “CRISPR, surrogate licensing, and scientific discovery,” Science, vol. 355, no. 6326, pp. 698–700, 2017.
16. L M. Girvan and M. E. J. Newman, “Community structure in social and biological networks.,” Proc. Natl. Acad. Sci. U. S. A., vol. 99, no. 12, pp. 7821–6, Jun. 2002