K-means Clustering on S&P 500 Data

Jessica Kwok, Melody Zhao, Kewei Zhou

Recap of Midterm Project Purpose

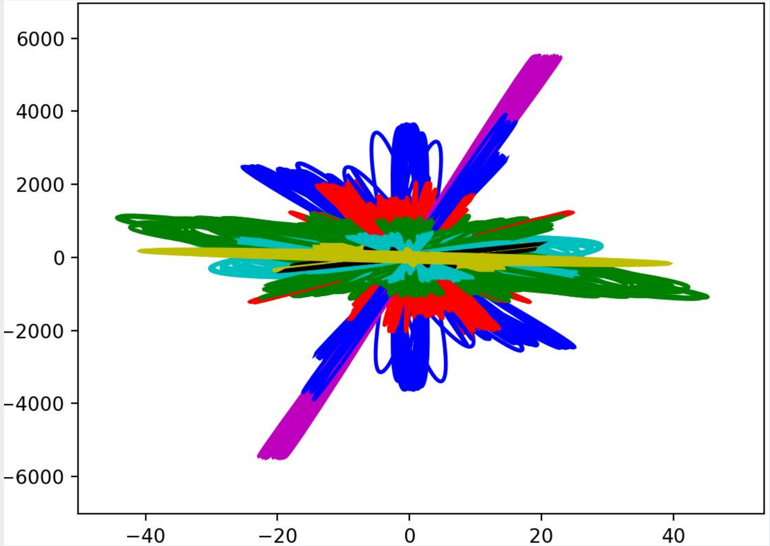
- Connect significant events to stock performance trend during that period of time
    - E.g. COVID-19, SARS, 911
- Predict future trend based on current trends
    - Pattern finding
    - Help people be more prepared

Midterm Project Methods

- Data Pre-processing
- Principal Component Analysis (PCA)
- K-means Clustering
- Markov Model

Midterm Project Results

Result from K-Means Clustering



Result from Hidden Markov Model

Four day’s label = [0 1 2 3]

Fifth day’s label

Accuracy: **92%**

Outline

- Optimizing parameters for K-Means Model
  - Compare to Previous Results
- Applying to a new variables
  - Original: Price Change vs. Volume Change
  - New: Price Change vs. Volatility
- RNN for time series prediction

Testing Measure

- Question: How to quantify how “good” is the clustering?
- Silhouette Coefficient
  - **Cohesion**: how similar an object is to its own cluster
  - **Separation**: how different an object is to other clusters
  - Range: 1 to -1 (1: well-matched, 0:poorly matched)

$$s = \frac{b - a}{\max(a, b)}$$

a = the mean intra-cluster distance
b = the mean nearest-cluster distance of all the samples.

Testing Measure (cont.)

- For reference regarding what clustering is considered good

RANGE OF SC	INTERPRETATION
0.71-1.0	A strong structure has been found
0.51-0.70	A reasonable structure has been found
0.26-0.50	The structure is weak and could be artificial. Try additional methods of data analysis.
Below 0.25	No substantial structure has been found

Changing Distance Function

$$D = W_{center} * \Delta_{center} + W_{\lambda 1} * \Delta_{\lambda 1} + W_{\lambda 2} * \Delta_{\lambda 2} + W_{\theta} * \Delta_{\theta}$$

$$W_{center} + W_{\lambda 1} + W_{\lambda 2} + W_{\theta} = 1$$

Silhouette score: 0.27

$$D = W_{center} * (\Delta_{center})^2 + W_{\lambda 1} * (\Delta_{\lambda 1})^2 + W_{\lambda 2} * (\Delta_{\lambda 2})^2 + W_{\theta} * (\Delta_{\theta})^2$$

$$W_{center} + W_{\lambda 1} + W_{\lambda 2} + W_{\theta} = 1$$

+ 0.24

Silhouette score: 0.51

Optimizing Parameters for K-Means Model

- Since we have a quantitative measure, we can optimize the parameters
- Library used: Scipy optimize
    - minimize
- Objective function: - Silhouette Score
- Input/Variable: the weights

$$[W_{center}, W_{\lambda 1}, W_{\lambda 2}, W_{angle}]$$

Optimization (cont.)

- Nelder-Mead Method:
    - No constraint, $W_{center} + W_{\lambda 1} + W_{\lambda 2} + W_{\theta} > 1$
    - Direct search method, which does not require gradient of the function
- Methods with non-linear constraint and bounds
    - $W_{center} + W_{\lambda 1} + W_{\lambda 2} + W_{\theta} = 1$
    - $W_{center}, W_{\lambda 1}, W_{\lambda 2}, W_{\theta} \in (0, 1)$
    - **SLSQP**, trust-constr, COBYLA

Optimization (cont.)

- Difficulties:
  - Constraints are too harsh → Get the input as output
  - A lot of local minimums → cannot find global minimum
    - Manually selected ~15 points, looped through to calculate score
    - Use the maximum as the initial guess for the optimize function
  - Silhouette function is not smooth and continuous
    - Other models and methods may be needed
  - Inconsistent Silhouette score, due to random sampling and rounding
    - Might confuse the minimize function

Compare Previous Result to Current Result

Before Optimizing

Weight = [0.2, 0.78, 0.015, 0.005]

Silhouette score: 0.51

+ 0.22

After Optimizing

Weight = [0.114, 0.786, 0.0365, 0.0635]

Silhouette score: 0.73

New Variable

- Instead of passing in volume change vs. price change, we passed in frac-high vs. price change

$$frac_{high} = \frac{(high - open)}{open}$$

- Used as input for many published Hidden Markov Model
- Measure of volatility

New Variable (cont.)

Step 1

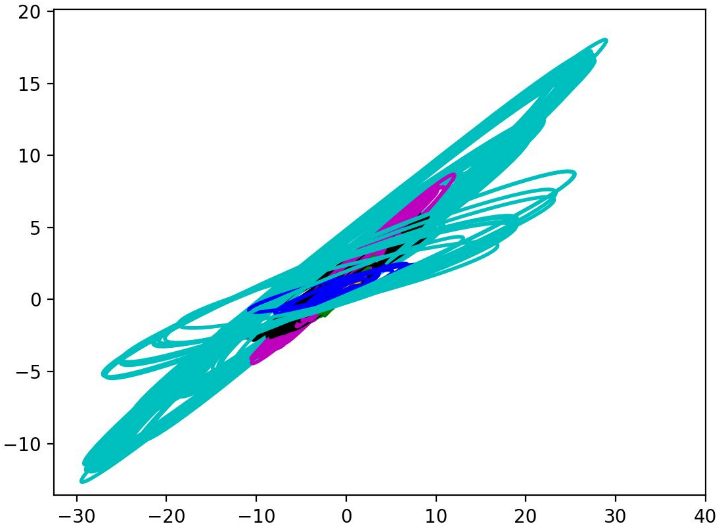
Manually pick possible weights

Step 2

- Loop through the weights
- highest silhouette score: 0.50 with parameter [0.1, 0.1, 0.1, 0.7]

Step 3

- Starting from highest score parameter, use SLSQP to minimize -silhouette score
- Highest score: 0.55 with parameter [0.09, 0.1, 0.1, 0.71]

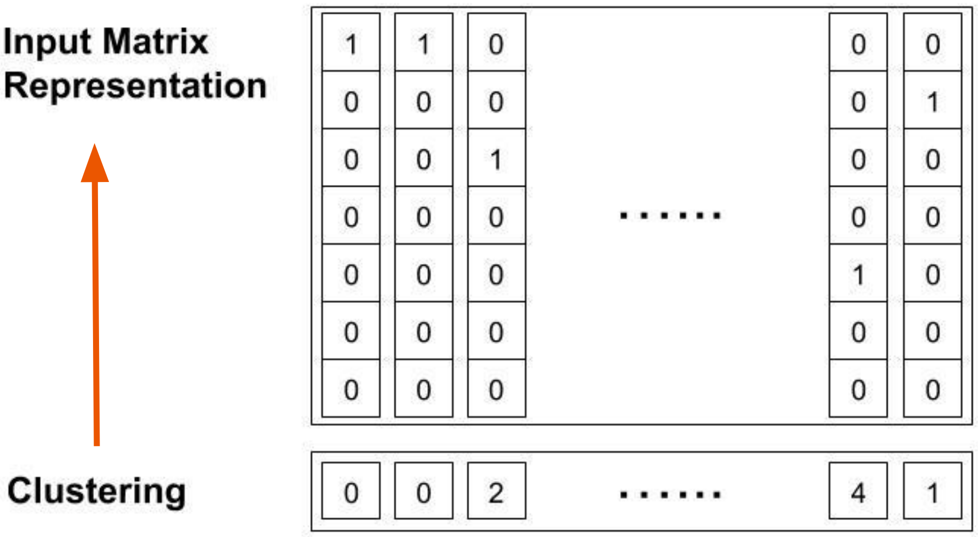


Parameter: [0.09, 0.1, 0.1, 0.71]

Cluster Prediction: Drawbacks of Markov Model

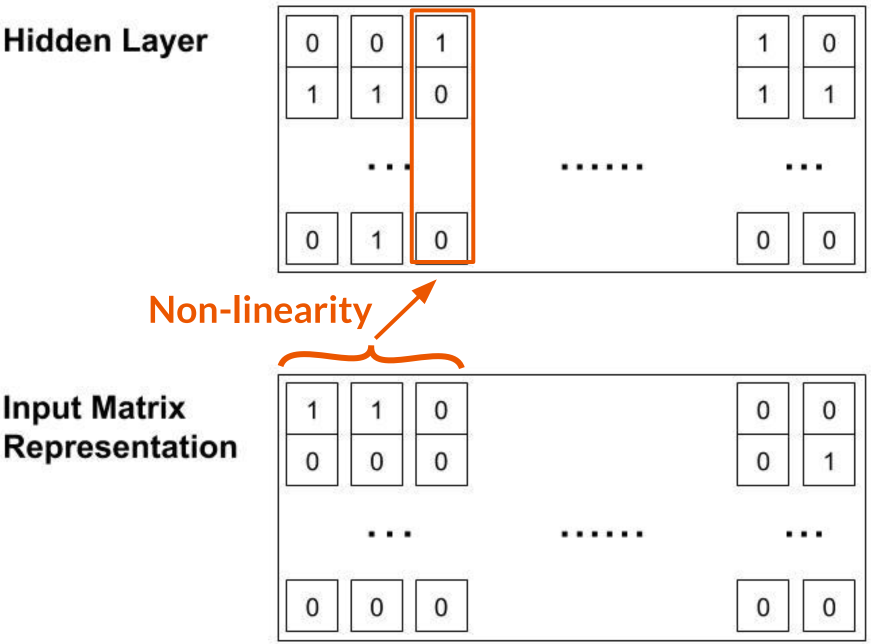
- Robust for short sequences
- Significantly worse predictions for long sequences
  - Long sequences not found in historical data
- Only takes into account of a small fraction of historical data

Recurrent Neural Network (RNN)



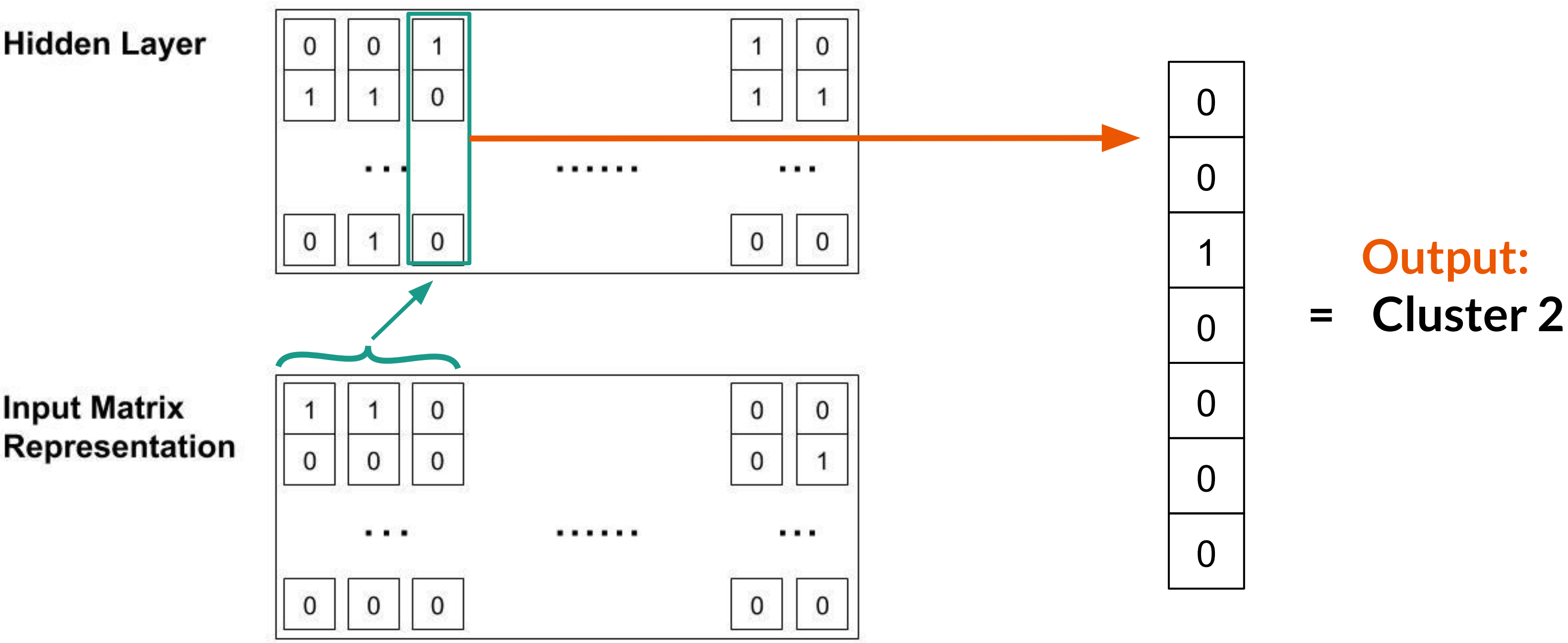
Sequential Data!!

Recurrent Neural Network (RNN)



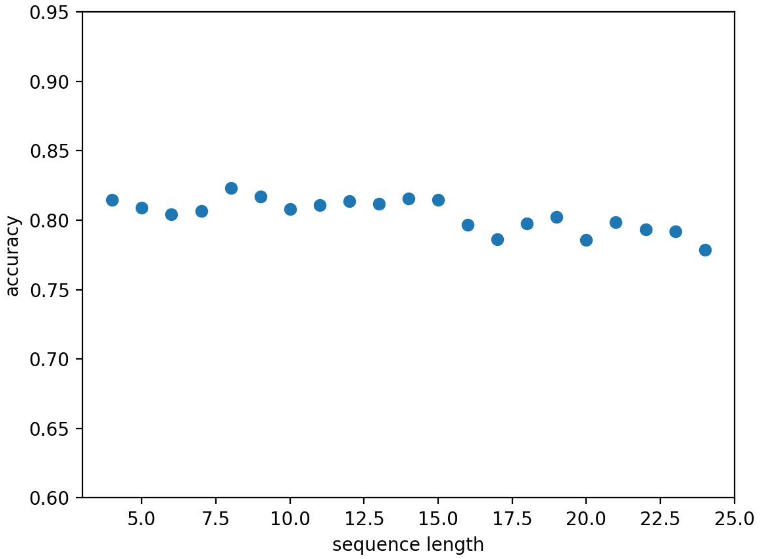
Stack multiple layers to achieve higher coverage

Recurrent Neural Network (RNN)

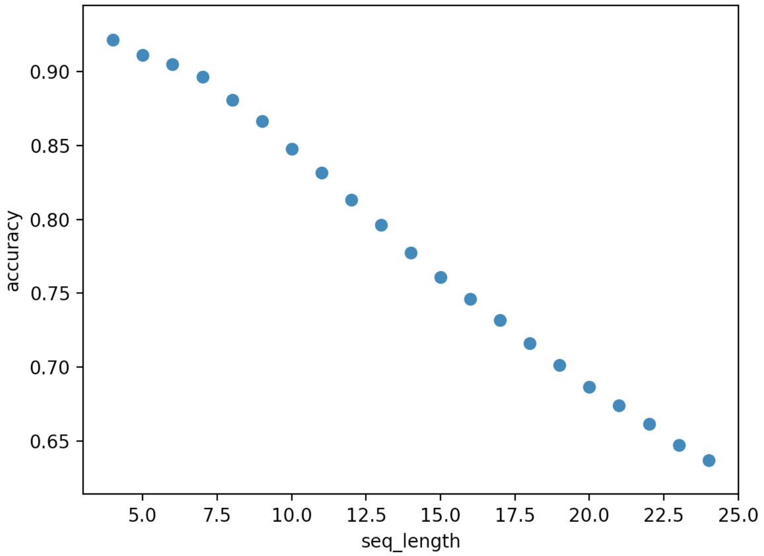


RNN for time series prediction (original)

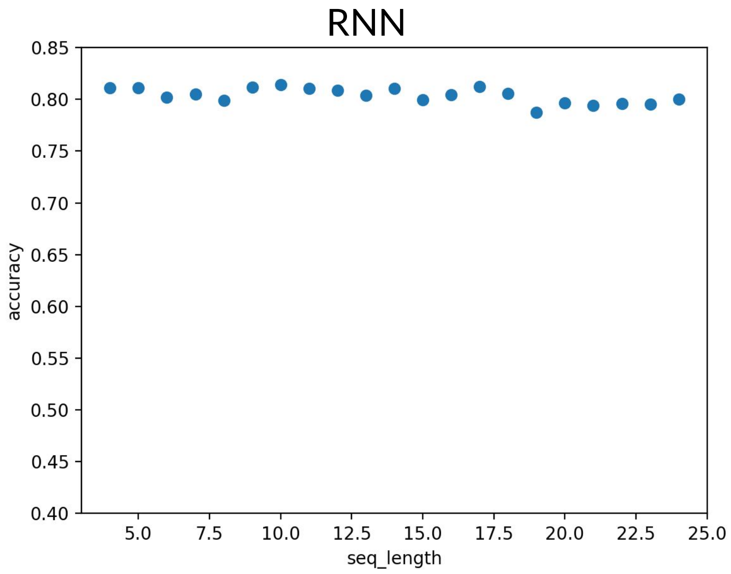
RNN

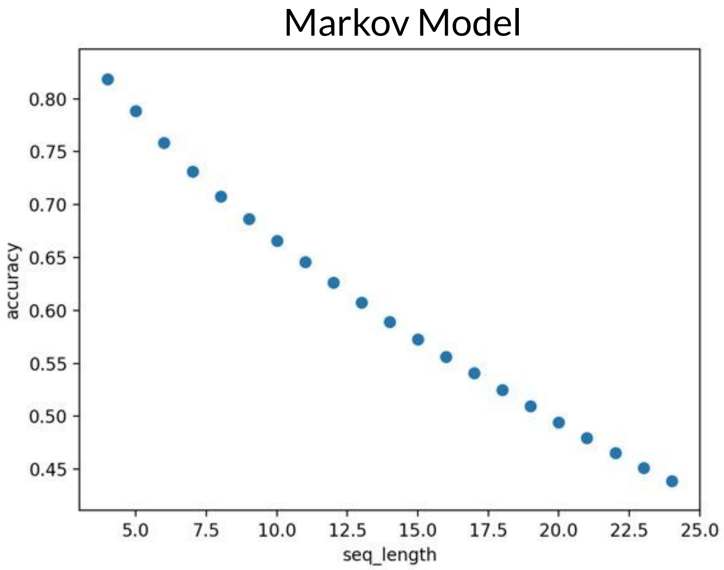


Markov Model



RNN for time series prediction (new variables)





Conclusion

- Using Scipy's optimize function to minimize -silhouette score can effectively be used to identify optimal parameters, which improves clustering
- RNN model is more robust than Markov Model for longer sequences
- Methods used in this project can be applied to different variables and datasets
    - Since the accuracy for RNN is relatively stable for both pairs of variables, it offers more reliable results that can be easily adapted in the future

Future Work

- Compare trend predicted by Volume change vs. Price change and Frac-high vs. Price change model
- Try on other stocks
- Improve on RNN
  - Try other cost functions
  - Optimize parameters
  - Warm-start/retrain the models to achieve higher accuracy
  - Mitigate short term memories using GRU and/or LSTM

References

- Partitioning Around Medoids (Pam), web.archive.org/web/20111002220803/www.unesco.org:80/webworld/idams/advguide/Chapt7_1_1.htm.
- “Optimization (Scipy.optimize)¶.” Optimization (Scipy.optimize) - SciPy v1.4.1 Reference Guide, docs.scipy.org/doc/scipy/reference/tutorial/optimize.html.
- “Text Generation with an RNN : TensorFlow Core.” TensorFlow, www.tensorflow.org/tutorials/text/text_generation.
- “Sklearn.metrics.silhouette_score¶.” Scikit, scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.