Design of a Hybrid Model for Accurate Prediction of Increasing Temporal Big Data in the Cloud for Mining Large Collections of Co-evolving Sequences

Abstract - **The exponential growth of temporal big data in cloud environments has posed significant challenges for accurately predicting and mining large collections of co-evolving sequences. In this study, we propose a novel hybrid model designed to address these challenges effectively. The hybrid model integrates the strengths of both traditional statistical methods and advanced machine learning techniques to achieve enhanced predictive accuracy. Leveraging cloud computing infrastructure, the model is scalable and capable of handling vast amounts of data efficiently. We conduct comprehensive experiments using real-world datasets to evaluate the performance of the proposed hybrid model against existing approaches. Results demonstrate that our model consistently outperforms baseline methods in terms of prediction accuracy and scalability, making it a promising solution for mining co-evolving sequences in the era of increasing temporal big data in cloud environments.**

I. Introduction –

The rapid expansion of digital data in recent years has led to the emergence of massive repositories of temporal data stored in cloud environments. This surge in data volume, often referred to as big data, presents both opportunities and challenges for various fields, including data mining and predictive analytics. In particular, the analysis of co-evolving sequences within these large datasets has become increasingly crucial for uncovering meaningful patterns and insights.

Co-evolving sequences represent a class of data where multiple sequences evolve concurrently over time, exhibiting complex interdependencies and temporal correlations. Examples of such data abound in diverse domains, including biological sciences, financial markets, social networks, and Internet traffic analysis. Extracting valuable knowledge from these co-evolving sequences requires advanced predictive models capable of handling the scale and complexity of modern big data environments.

Traditional statistical methods, while effective to some extent, often struggle to cope with the sheer volume and velocity of data generated in cloud-based systems. On the other hand, machine learning techniques offer promising avenues for analyzing large-scale datasets and uncovering intricate patterns.

However, deploying these techniques in cloud environments requires careful consideration of scalability, resource utilization, and computational efficiency.

This hybrid model is separated into 3 layers:

K-Means Layer – Using the K-means clustering to represent different patterns in stock price movements.

HMM initialization Layer– Each HMM will have states representing different market conditions or trends, with emission probabilities corresponding to the likelihood of observing specific changes in stock price parameters given the current state

Baum-Welch application to HMM Layer - The Baum-Welch algorithm iteratively refines the parameters of the HMMs to maximize the likelihood of observing the stock price data.

After training, the HMMs are ready to be used for analyzing and predicting future stock price movements.

II. Related Work

A. Improved K-Means Clustering Algorithm

I. The proposed improved K-means clustering algorithm addresses the challenge of computationally expensive initial centroid selection. By leveraging Principal Component Analysis (PCA) for dimensionality reduction and strategically distributing centroids using percentiles, this method achieves fewer iterations, shorter execution time (especially for large datasets), and maintains or improves clustering quality. Overall, it offers a promising approach for efficient data-driven modeling by enhancing initial centroid selection, leading to faster and potentially more accurate results.

II. The research paper introduces an enhanced K-means clustering algorithm that combines the traditional K-means approach with the largest minimum distance algorithm. The proposed method begins by using the largest minimum distance algorithm to determine K initial cluster focal points. These focal points are then integrated into the K-means algorithm to achieve pattern classification. The algorithm's steps include determining the initial cluster focal point, calculating distances between each sample and the focal point, and iterating until convergence. Experimental results, conducted using emulation data in the Visual C++ 6.0 development environment, demonstrate that the improved K-means algorithm maintains high efficiency while significantly increasing convergence speed. Overall, it outperforms the traditional K-means in terms of cluster precision, speed, stability, and other relevant aspects.

III. The authors propose an improved K-means algorithm based on density to address the sensitivity of the traditional K-means algorithm to initial cluster centers and the requirement of specifying the K value in advance. The improved algorithm selects initial clustering centers based on density, enhancing the accuracy of clustering. It is

evaluated using local density calculations, and a cut-off kernel function, and tested on the Iris, Hayes-Roth, and Wine datasets. While the algorithm shows good clustering effects and reduces the interference of noise points, it has lower computational efficiency and longer running time compared to the traditional K-means algorithm. It performs acceptably on small datasets but has limitations when dealing with datasets with a large range of attribute values.

IV. The paper proposes an enhanced K-means clustering algorithm that addresses two critical issues associated with the traditional approach. Firstly, it introduces a novel indicator called the "Between-Within Proportion" (BWP) to determine the optimal number of clusters (k). By analyzing clustering effectiveness, this method automatically identifies the most suitable k value. Secondly, the algorithm modifies the traditional Sum of Squared Errors (SSE) criterion function by incorporating weights based on cluster size and standard deviation. This adaptation overcomes SSE's limitation, making it more suitable for non-spherical and heterogeneous clusters. Experimental results using randomly generated 2D data demonstrate that the improved algorithm surpasses traditional methods. It achieves accurate clustering of non-uniform data with fewer iterations and lower computational complexity, effectively addressing challenges related to optimal k selection and non-uniform data densities.

V. The authors address the inefficiency of the traditional K-means algorithm in handling high dimensional and sparse data in text clustering by proposing an improved K-means algorithm based on SimHash. This method reduces the dimensionality of the text and enhances the distance calculation method. The text is preprocessed, and mapped to a feature vector using VSM, and SimHash is employed to calculate the feature vectors and obtain each text's fingerprint. The Hamming distance between fingerprints is used to determine data clustering. The algorithm iteratively partitions and recomputes cluster center vectors until convergence. The proposed algorithm improves the execution efficiency of K-means text clustering while preserving clustering quality, leading to enhanced accuracy and efficiency through the reduction of text dimension via SimHash and the use of Hamming distance for clustering.

B. Hidden Markov Model (HMM):

I. In the realm of predicting stock prices, the complexity and volatility of market data have prompted researchers to explore various methodologies. Traditional approaches like Artificial Neural Networks (ANNs) have been extensively studied but often lack interpretability and struggle to explain the underlying model. In contrast, Hidden Markov Models (HMMs) offer a compelling alternative, leveraging a strong statistical foundation to elucidate stock price behavior. This study contributes to this ongoing discourse by applying HMMs to forecast stock prices for interrelated markets, specifically focusing on Southwest Airlines stocks. Through comparative analysis with

ANNs, the research underscores the interpretability and efficacy of HMMs in capturing intricate patterns within stock market data, thus advocating for their adoption in future time series prediction endeavors.

II. The study introduces the application of Hidden Markov Models (HMM) to forecast daily stock prices for three prominent trading stocks, namely Apple (AAPL), Google (GOOGL), and Facebook (FB), leveraging historical data. Addressing the pivotal role of stock performance as a barometer of corporate strength and economic viability, the research employs HMM to navigate the intricate dynamics influencing stock prices. Through a methodical approach, the study selects the optimal number of hidden states in HMM based on Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Subsequently, HMM parameters are calibrated iteratively, culminating in the prediction of stock prices by identifying past days with similar likelihoods to the present. Trading strategies devised from these predictions yield significant returns for AAPL, FB, and GOOGL, outperforming naive approaches for FB and GOOGL but exhibiting a nuanced performance for AAPL. Evaluation metrics, including Mean Absolute Percentage Error (MAPE), validate the efficacy of HMM in stock price prediction, positioning it as a promising model for future trading endeavors. The study's outcomes underscore the importance of model calibration and comparative analysis in refining predictive accuracy, while also signaling avenues for further investigation into enhancing AAPL stock predictions using HMM.

III. The study introduces a novel hybrid model amalgamating Artificial Neural Networks (ANNs), Hidden Markov Models (HMMs), and Genetic Algorithms (GAs) for time series data prediction, focusing on stock price forecasting amidst the high volatility of the market. Leveraging historical data from Apple Computer Inc., International Business Machines Corporation (IBM), and Dell Inc., the research devises a comprehensive methodology. ANNs preprocess daily stock prices into independent sets, injecting noise to enhance compatibility with HMMs, while GAs optimize initial HMM parameters, further refined via the Baum-Welch algorithm. Trained HMMs identify similar historical patterns, guiding price predictions through weighted averages of price differences. Comparative analysis highlights superior performance of the fusion model over standalone HMMs and comparable prediction accuracy to ARIMA models. Notably, the proposed hybrid model obviates the need for extensive data preprocessing, offering a streamlined approach for users. Future research avenues include exploring Genetic Algorithms to optimize HMM architectures tailored to specific datasets, underscoring the model's adaptability and potential for further enhancement.

IV. The study proposes a novel approach that integrates Hidden Markov Models (HMM) to discern data patterns and Fuzzy Logic to derive predicted values for stock prices, addressing the inherent complexity, volatility, and non-linearity of stock market data. Utilizing a comprehensive dataset spanning various airlines and technology companies,

the research delineates a meticulous methodology. HMMs are trained on the training dataset, with subsequent sorting based on log likelihood values and bucketing. Fuzzy rules are extracted iteratively using a top-down tree approach, followed by parameter optimization using Gradient Descent. The results showcase the superior prediction performance of the fusion model, combining HMM, ANN, and GA, over existing techniques such as ARIMA and standalone ANN. Notably, the HMM-fuzzy model demonstrates promising prediction accuracy, outperforming ARIMA and ANN, albeit with higher computational complexity. These findings underscore the efficacy of the proposed hybrid model in forecasting stock prices, offering a robust alternative to traditional approaches.

V. The stock market's dynamic and intricate nature poses significant challenges for accurate price prediction, prompting researchers to explore methodologies like Hidden Markov Models (HMMs) to model stock behavior based on historical data. The fifth paper provides a comprehensive theoretical framework, detailing the application of HMMs in predicting stock closing prices using preceding day data. Evaluation metrics such as Mean Average Prediction Error (MAPE) and Directional Prediction Accuracy (DPA) assess the model's performance, highlighting its potential to assist traders in making informed investment decisions and potentially increasing profits. However, the paper also underscores the complexities and risks associated with algorithmic trading, cautioning retail investors against over-reliance on such strategies without adequate research, infrastructure, and resources. Additionally, the volatile nature of markets and the potential for rapid changes underscore the need for vigilance and adaptability in utilizing algorithmic trading techniques.

C. Baum- Welch algorithm

I. [8] developed a linear memory algorithm that efficiently calculates forward and backward probabilities, with a memory requirement independent of the sequence length. It iterates through the sequence from the beginning to the end, computing the forward probabilities for each state at each position. It only keeps track of the forward probabilities for the current and previous positions as the forward probabilities update iteratively as the algorithm progresses through the sequence. Also, the algorithm iterates through the sequence from the end to the beginning, computing the backward probabilities for each state at each position. Only the backward probabilities for the current and next positions are stored and updated iteratively. Using the forward and backward probabilities along with the observed emissions and transition probabilities of the HMM, the algorithm efficiently computes the emission and transition probabilities needed for the Baum-Welch update step. It efficiently models sequential patterns and state transitions over time. It estimates parameters through Baum-Welch

training, making it suitable for tasks like prediction and inference on temporal sequences while scaling effectively to handle large datasets.

II. [9] determines the model that maximizes the likelihood is the forward-backward algorithm. This algorithm involves defining forward probability, which represents the joint probability of generating a partial observation sequence up to time t and reaching state i at time t, given the HMM, as well as backward probability, which denotes the probability of generating the partial observation sequence from time t+1 to the end, given the HMM and starting from state i at time t.

III. [7] proposed two methods for training Hidden Markov Models (HMMs). The first method utilizes a MATLAB toolbox based on the Baum-Welch (B-W) algorithm, while the second method involves a custom implementation of the Genetic Algorithm (GA) using the MATLAB GA toolbox. Although GA offers powerful global search capabilities, it tends to converge slowly. To address this, a hybrid approach, GA-BW, is introduced to combine the strengths of both algorithms.

IV. [6] gathered historical temperature data from weather stations across the Indian Himalayas. Preprocessing techniques were applied to ensure data quality. HMMs were developed and optimized using the Baum–Welch algorithm. Model performance was evaluated using metrics such as Root Mean Squared Error (RMSE). The Baum–Welch algorithm played a crucial role in optimizing the model's parameters, enabling it to capture temporal dependencies and hidden states within the data effectively.

III. Solution –

The solution mainly consists of five steps.

Step – 1: The data is pre-processed and features are extracted.

Say, we have a dataset of daily stock prices for a single company, containing parameters such as timestamp, open price, high price, low price, close price, and volume traded.

Features can be extracted from this dataset, such as

- Daily percentage changes in stock prices (e.g., close-open / open).
- Moving averages over a certain time window (e.g., 5-day moving average).
- Technical indicators like the Relative Strength Index (RSI) or Moving Average Convergence Divergence (MACD).

Step 2: We apply the improved K-means algorithm.

Pseudocode of improved K-means algorithm –

1. *Input: A dataset **D** and the number of clusters **K**.*

2. Apply Principal Component Analysis (PCA) with 2 components to the dataset D.

*3. Apply percentile for splitting the whole dataset into **K** equal parts based on the 1st component.*

4. Extract the split dataset from the primary data by index.

5. Calculate the mean of each attribute of the split datasets.

*6. Take the mean of each dataset as the initial clusters centroids, **C = c1, c2,..., ck**, where **c1, c2,.., ck** are the initial centroids for **1st, 2nd,...., k** clusters consecutively.*

7. Assign the centroids to the K-means clustering algorithm.

8. Take the initial centroids C and assign each instance d1 to a cluster, Ki by the closest distance.

9. Calculate the new centroids by taking the mean of each cluster.

10. Repeat the above process until the centroids converge.

11. Generate the final clusters by passing the proposed centroids through the K-means algorithm until the centroids converge.

Step 3: Initialize the HMM.

Each of the cluster centroids obtained from the improved K-Means process serves as a prototype for a different pattern of stock price behavior. These centroids will be used to initialize the parameters of the HMMs.

Hidden Markov Model consists of 3 main components :

1. Set of hidden states, denoted by S={s_1, s_2, …. s_N}
2. Set of possible observation symbols, denoted by V={v_1, v_2, …. v_M}
3. Set of parameters :
    a. Initial State Probabilities (π) - A vector π of length N, where π_i represents the probability of starting in state s_i
    b. Transition Probabilities (A) - A matrix A of size N x N, where a_{ij} represents the probability of transitioning from state s_i to state s_j
    c. Emission Probabilities (B) - A matrix B of size N x M, where b_{ik} represents the probability of emitting symbol v_k when in state s_i

Initialization of HMM parameters -

1. Initial State Probabilities (π) - Initially, equal probability is assigned to each state.
2. Transition Matrix (A) - Each state has a higher probability of transitioning to the nearby states in the K-means space (based on centroid distance).
3. Emission Matrix (B) - The emission probability of each state for the observed symbols could be based on the distribution of symbols associated with the corresponding K-means cluster.

Step 4: Estimate the parameters for the HMM using the Baum-Welch algorithm.

The Baum-Welch algorithm iteratively refines the parameters of the HMMs to maximize the likelihood of observing the stock price data. After training, the HMMs are ready to be used for analyzing and predicting future stock price movements.

Baum–Welch algorithm is based on both forward and backward variables. The backward variable is the probability of the partial observation sequence from time step t+1 to the end. It can be calculated, iteratively, in the following steps:

Initialization:

$$\beta_T(i) = 1, \quad 1 \le i \le 10$$

Induction:

$$\beta_t(i) = \sum a_{ij} * b_j(O_{t+1}) * \beta_{t+1}(j), \quad 1 \le t \le 9, \ 1 \le j \le 10.$$

From the forward and backward variables, two new variables can be calculated. The first one is ξt(i, j), the probability of being in state Si at time t, and state Sj at time t+1, given the model and the observation sequence, i.e,

$$\xi_t(i,j) = \alpha_t(i) \times a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j) / P(O/\lambda).$$

The second one is γt(i), the probability of being in state Si at time t, given the observation sequence O, and the model, i.e.,

$$\gamma_t(i) = \alpha_t(i) \times \beta_t(i) / P(O/\lambda).$$

The variables ξ and γ satisfy the relationship

$$\gamma_t(i) = \sum \xi_t(i,j), \quad 1 \le j \le 10.$$

Now if γt(i) is summed over all instants (excluding instant T), the expected number of times the state Si has been visited is obtained. On the other hand, if ξt(i, j) is summed over all instants (excluding instant T), the expected number of transitions made from state i to j is obtained. From this behavior of γt(i) and ξt(i, j), the following re-estimations of the model parameters could be deduced:

$$\pi' = \gamma_1(i),$$

$$a'_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \gamma_t(i),$$

$$b'_{ij}(k) = \sum_{\substack{t=1 \\ O_t = v_k}}^{T} \gamma_t(i) \Big/ \sum_{t=1}^{T} \gamma_t(i).$$

After re-estimations of the model parameters, a new model λ is obtained, which is more likely than model λ, producing observation sequence O. This process of re-estimation is continued till no improvement in the probability of observation sequence is reached.

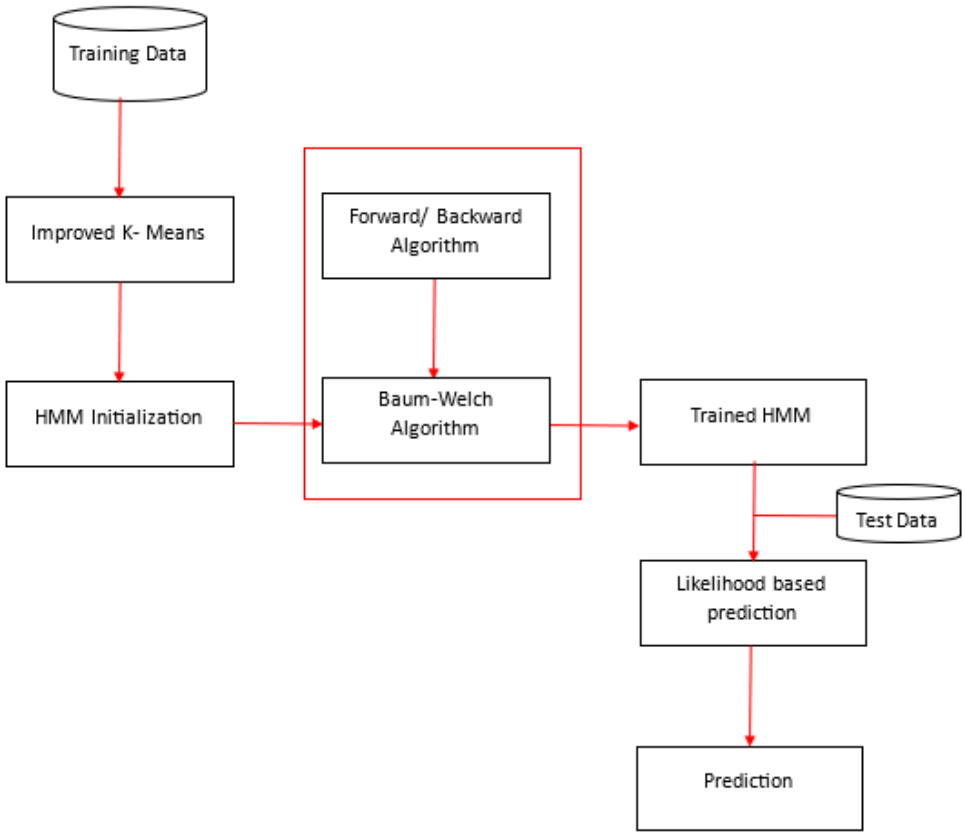

Fig 1: Workflow Diagram

The workflow depicted in the diagram commences with the meticulous collection and preparation of training data, which is pivotal in ensuring the efficacy of the ensuing model. Through the application of improved K-Means clustering, the data undergoes preprocessing aimed at enhancing its suitability for pattern recognition. Subsequently, the Hidden Markov Model (HMM) is initialized with essential parameters, laying the groundwork for the utilization of the Forward and Backward algorithms to compute the likelihood of observed sequences given the model. The iterative refinement facilitated by the Baum-Welch algorithm meticulously adjusts the HMM's parameters, bringing them into closer alignment with the underlying patterns within the data. Upon completion of training, the HMM undergoes rigorous testing using distinct datasets to evaluate its predictive prowess. By harnessing likelihood-based prediction methods, the model extrapolates future observations or infers hidden states within the data, furnishing valuable insights crucial for informed decision-making.

References

[1] Zubair M, Iqbal MDA, Shil A, Chowdhury MJM, Moni MA, Sarker IH. An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling. Ann. Data. Sci. 2022 Jun 25:1–20. doi: 10.1007/s40745-022-00428-2. Epub ahead of print. PMCID: PMC9243813.

[2] L. Zhang, J. Qu, M. Gao and M. Zhao, "Improvement of K-means algorithm based on density," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2019, pp. 1070-1073, doi: 10.1109/ITAIC.2019.8785550.

[3] G. Wu, H. Lin, E. Fu and L. Wang, "An Improved K-means Algorithm for Document Clustering," 2015 International Conference on Computer Science and Mechanical Automation (CSMA), Hangzhou, China, 2015, pp. 65-69, doi: 10.1109/CSMA.2015.20.

[4] Hong Zhang and Hong Yu and Ying Li and Baofang Hu, “Improved K-means Algorithm Based on the Clustering Reliability Analysis,” Proceedings of the 2015 International Symposium on Computers & Informatics, publisher: Atlantis Press, 2015/01, doi: 10.2991/isci-15.2015.326

[5] Li, Youguo & Wu, Haiyan. (2012). A Clustering Method Based on K-Means Algorithm. Physics Procedia. 25. 1104-1109. 10.1016/j.phpro.2012.03.206.

[6] Joshi J, Kumar T, Srivastava S, Sachdeva D. Optimisation of hidden Markov model using Baum-Welch algorithm for prediction of maximum and minimum temperature over Indian Himalaya. J Earth Syst Sci. 2017;126(1):3

[7] M. Oudelha and R. N. Ainon, "HMM parameters estimation using hybrid Baum-Welch genetic algorithm," 2010 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 2010, pp. 542-545, doi: 10.1109/ITSIM.2010.5561388.

[8] Miklós, I., & Meyer, I. M. (2005). A linear memory algorithm for Baum-Welch training. *BMC Bioinformatics*, *6*(1), 1-8

[9] Wu, Y., Ganapathiraju, A., & Picone, J. (1999). Baum-Welch re-estimation of hidden markov model. *Department of Electrical and Computer Engineering, Mississippi State University*, *74*.