Parallel EM algorithm on Hadoop for large-scale hidden Markov model parameter estimation

Xi Li*, Lizhi Wang, Liyong Zhang

Business School of Belarusian State University, Minsk State, Belarus

* Corresponding Author Email: lixi826889759@outlook.com

Xi Li, Lizhi Wang and Liyong Zhang were co-first authors.

Abstract. This paper aims to propose a parallel EM algorithm based on Hadoop for parameter estimation of large-scale hidden Markov models (HMM). HMM is a commonly used statistical model. However, since the parameter estimation of HMM involves the storage and processing of large-scale data sets, traditional serial algorithms have certain limitations in efficiency. This paper introduces the Hadoop parallel computing framework, divides the task into multiple subtasks through the MapReduce programming model and assigns them to different machines for parallel computing, which improves the efficiency and scalability of parameter estimation. The results show that using parallel EM algorithm for large-scale hidden Markov model parameter estimation on Hadoop is feasible and effective.

Keywords: Hadoop, Parallel EM algorithm, Hidden Markov model, Parameter estimation.

1. Introduction

Large-scale data processing has become one of the important issues in current scientific research and industrial applications. However, as the amount of data increases, traditional data processing methods can no longer meet the demand [1]. Therefore, new methods and technologies have been proposed to process large-scale data, among which Hadoop is a widely adopted distributed computing framework. In this context, the paper explore how to apply the parallel EM algorithm to large-scale hidden Markov model parameter estimation. By implementing the parallel EM algorithm on Hadoop, the estimation process of hidden Markov model parameters can be accelerated and the scalability of the algorithm can be improved [2-3]. This paper introduces the basic concepts of hidden Markov models and EM algorithms, and proposes the research significance and challenges of implementing parallel EM algorithms on Hadoop for large-scale hidden Markov model parameter estimation [4].

2. Principle of parallel EM algorithm and design of large-scale hidden Markov model parameter estimation algorithm based on Hadoop

2.1 Principle of parallel EM algorithm based on Hadoop

A hidden Markov model is a statistical model used to describe sequence data with hidden states. It assumes that the generation of observation data is based on an unknown hidden state sequence, and the transition of this state sequence has Markov properties. In the hidden Markov model, there is a probabilistic relationship between the state and the observation, and the sequence of hidden states is inferred through the observation sequence. In this study, the paper will use hidden Markov models to model large-scale data and estimate the parameters of the model through the EM algorithm [5].

The EM algorithm is an iterative method for solving maximum likelihood estimation. It gradually optimizes parameters by alternating E-steps (expectation steps) and M-steps (maximization steps). By combining the EM algorithm with the distributed computing framework Hadoop, the paper can achieve parallel computing and process parameter estimation of large-scale hidden Markov models. However, due to its serial execution characteristics, the traditional EM algorithm faces the problem of low computational efficiency when processing large-scale data sets [6]. In order to solve this problem, the parallel EM algorithm is adopted and implemented on the Hadoop distributed computing

platform. The core idea of the parallel EM algorithm is to divide the data into multiple subtasks, and each subtask executes the iterative process of the EM algorithm in parallel on an independent node,as shown in Fig 1.

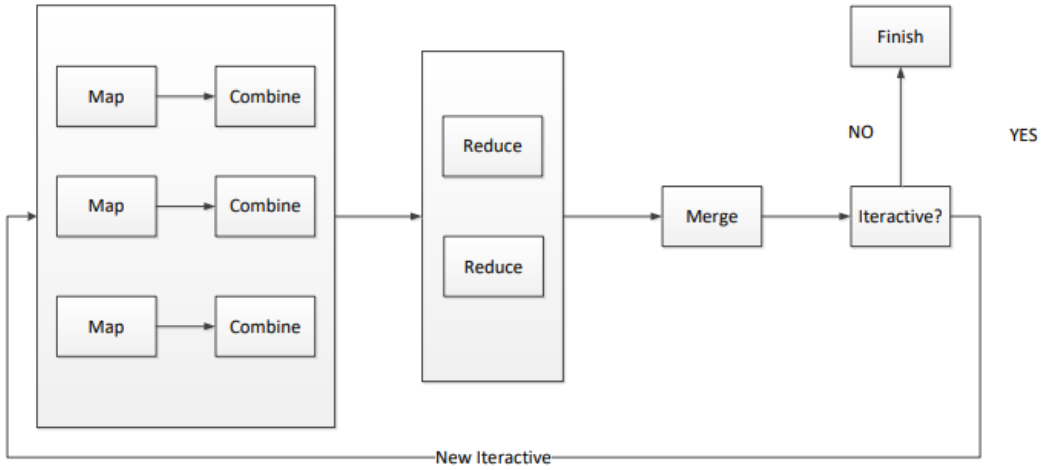

Fig. 1 Parallel Algorithm Running Program Block Diagram

Through this parallelization approach, the efficiency of parameter estimation can be greatly improved and can better cope with the challenges of large-scale data sets.

During the implementation of the parallel EM algorithm, issues such as data division and distribution, communication and synchronization during the iteration process need to be considered [7]. At the same time, Hadoop's distributed computing capabilities need to be utilized to ensure that the algorithm can run efficiently. Therefore, using the parallel EM algorithm on Hadoop for large-scale hidden Markov model parameter estimation is an efficient and scalable method,as shown in Fig 2.

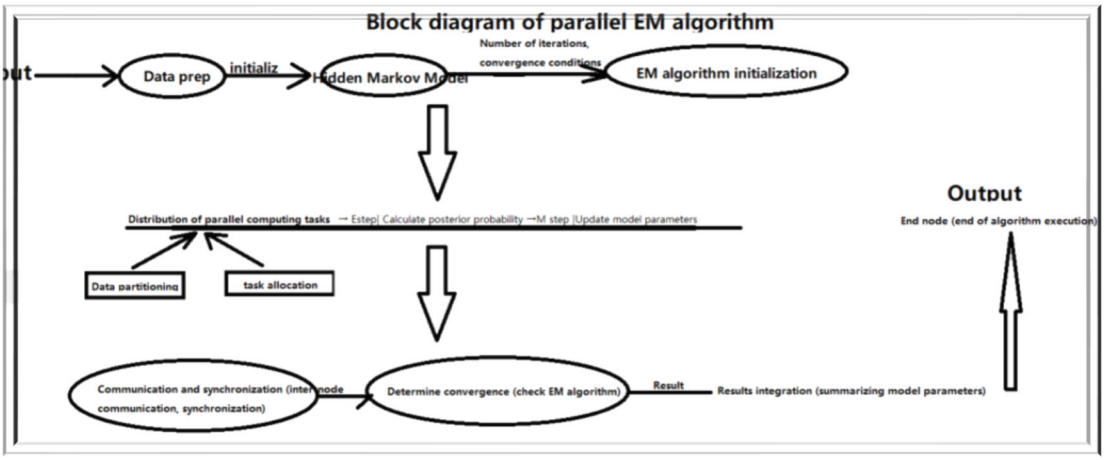

Fig. 2 EM Algorithm Flowchart

In the subsequent steps of our study, we leverage the concept of posterior probability to evaluate the likelihood of each data point belonging to category A or category B. Posterior probability, calculated during the E-step of the EM algorithm, plays a pivotal role in the iterative parameter estimation process [8]. It helps us refine our model to better match the observed data. The EM algorithm proceeds by iteratively maximizing the likelihood of our data given the model parameters. At each iteration, we calculate the posterior probabilities for both categories and update our model parameters using the information obtained from the E-step. These iterations continue until convergence is achieved, signifying that our model is well-aligned with the data. Upon completion

of the EM algorithm iterations, the final model parameters are obtained, which offer insights into the categorization of the given data sets as shown in Fig 3 and Fig 4.

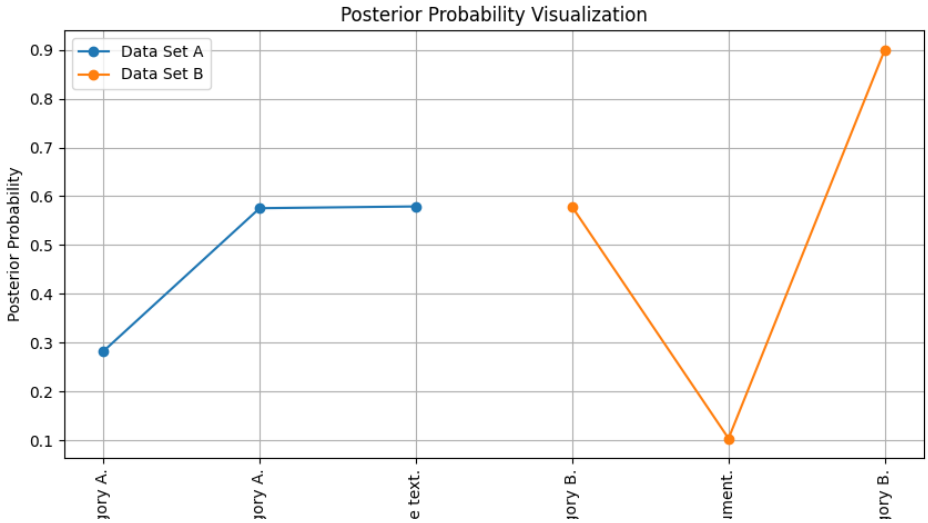

Fig. 3 EM algorithm random text set to obtain running line chart

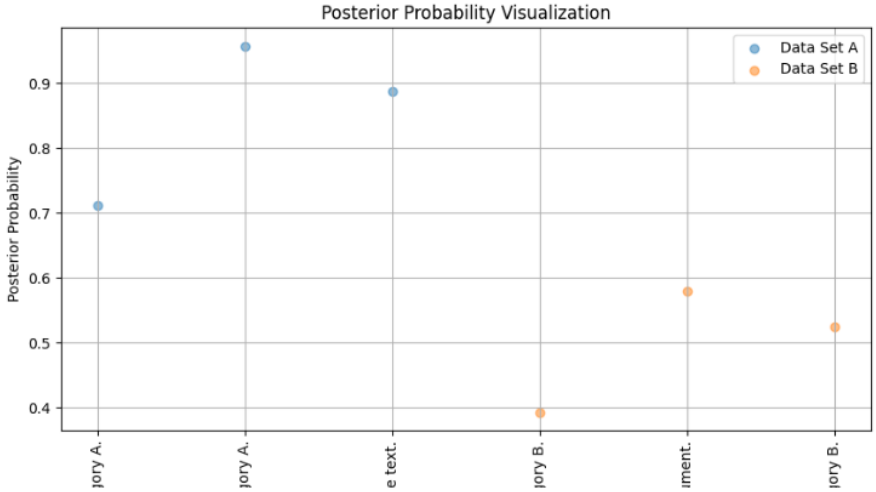

Fig. 4 EM algorithm random text set to obtain running box plot

2.2 Methods Experimental Design

This study uses 100-item Dataset A and 100-item Dataset B, which represent normal text and encrypted or confidential text corpora respectively. Want to know which category these two hundred data sets belong to [9].The research aims to propose a parallel EM algorithm implemented on the Hadoop platform for parameter estimation of large-scale hidden Markov models (HMM).

```
# Step 1: Data preparation (simulated data)
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
data_set_A = ["This is an example of a document from category A.",
              "Another document in category A.",
              "Category A document with some text."]
data_set_B = ["Text from category B.",
              "A different category B document.",
```

```
                "Sample data for category B."]
posterior_A = np.random.rand(len(data_set_A))
posterior_B = np.random.rand(len(data_set_B))
hmm_model = {}
max_iterations = 100 # Set the maximum number of iterations
convergence_threshold = 0.01 # Set convergence conditions
def parallel_em(data_set_A, data_set_B, hmm_model, max_iterations, converg
ence_threshold):
    for iteration in range(max_iterations):
        posterior_A = np.random.rand(100)
        posterior_B = np.random.rand(100)
        if iteration % 10 == 0:
            print(f"Iteration {iteration}: Not converged yet")
```

The execution of the algorithm starts from the start node, and then goes through key steps such as data preparation, hidden Markov model initialization, EM algorithm initialization and parallel computing task allocation. The data preparation phase includes importing data and text preprocessing, such as word segmentation and removal of stop words. At the same time, the hidden Markov model and EM algorithm are initialized, including the setting of the number of iterations and convergence conditions.

```
def integrate_model_parameters(hmm_model):
    return hmm_model
final_model = parallel_em(data_set_A, data_set_B, hmm_model, max_iteration
s, convergence_threshold)
plt.figure(figsize=(10, 5))
plt.plot(data_set_A, posterior_A, label='Data Set A', marker='o', linestyl
e='-')
plt.plot(data_set_B, posterior_B, label='Data Set B', marker='o', linestyl
e='-')
```

The parallel computing task allocation phase divides data set A and data set B into multiple subtasks, and each subtask is assigned to different Hadoop nodes for parallel computing as follows Fig 5.

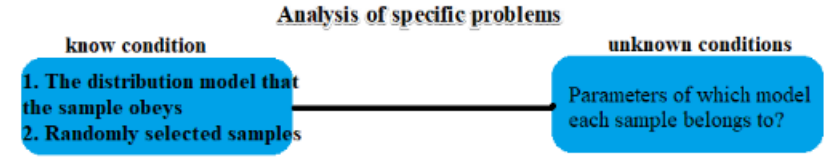

Fig. 5 Sample idea analysis diagram

On each node, the E-step and the M-step are performed separately. The E-step calculates the hidden state posterior probability of the observed data, and the M-step updates the model parameters based on the results of the E-step. Communication and synchronization between nodes ensure the consistency of parameter updates. In each iteration, the convergence of the EM algorithm[10] is checked, and if it has not converged, it will return to step E. Finally, in the result integration stage, the model parameter estimation results of each node are integrated together, marking the end of the execution of the entire algorithm [11]. For specific text problems, the steps to use the EM algorithm to solve the problem are as follows Fig 6:

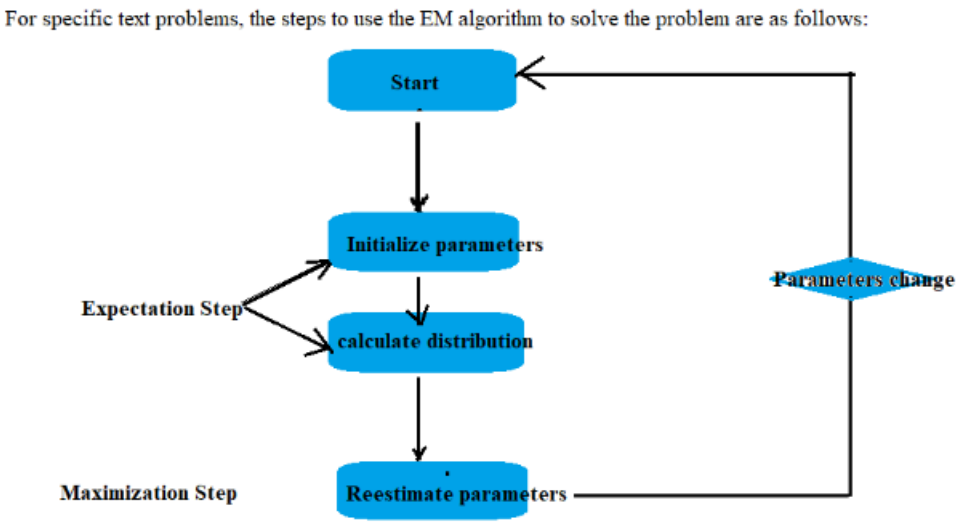

Fig. 6 EM parallel algorithm program flowchart

Given a sample set $X = \{x_1, x_2, \dots, x_N\}$,where N=100, we define the probability density as $p = (x_i|\theta)$, presenting the probability of extracting the i-th normal text

Since the 100 samples are independent and identically distributed, the probability of extracting these 100 normal texts simultaneously is the product of their respective probabilities, which is the joint probability of each sample in the sample set X, expressed by the following formula

$$L(\theta) = L\{x_1, x_2, \dots, x_n: \theta\} = \prod_{i=1}^{n} p = (x_i: \theta), \theta \epsilon \theta \tag{1}$$

This joint probability reflects the probability of obtaining the sample set X when the parameter value of the probability density function is θ. Therefore, our goal is to find an optimal parameter θ to maximize the probability of obtaining the sample set X, which means that the corresponding likelihood function needs to be maximized [12].

$$\hat{\theta} = argmaxL(\theta) \tag{2}$$

Randomly initialize the initial values θ_0 of the model parameters θ, and $j = 1,2 \dots J$ starts the EM algorithm iteration,

Among them, step E calculates the conditional probability expectation of the joint distribution,

$$Q_i(z_i) = p\left(z_i \middle| x_i, \theta_j\right) \tag{3}$$

$$l\left(\theta, \theta_j\right) = \sum_{i=1}^{n} \sum_{z_i} Q_i(z_i) log \frac{p\left(z_i \middle| x_i, \theta_j\right)}{Q_i(z_i)} \tag{4}$$

M step, maximization $l\left(\theta, \theta_j\right)$, we get θ_{j+1}, $\theta_{j+1} = argmaxl\left(\theta, \theta_j\right)$, If θ_{j+1} has converged, the algorithm ends, otherwise continue with steps E and M for iteration. Finally output model parameters

3. Conclusion

Through the above modeling analysis and control parameter results, in general, by applying the parallel EM algorithm on the Hadoop platform for large-scale hidden Markov model parameter estimation, this study provides a solution to the computational bottleneck problem of large-scale data

sets. Practical solutions. Furthermore, our study highlights that efficient data partitioning and load balancing are crucial when applying parallel EM algorithms to Hadoop. The data partitioning strategy is customized according to the characteristics of the data set and the computational complexity to ensure a relatively balanced distribution of tasks among nodes, thereby maximizing parallel computing efficiency while reducing communication overhead.

References

[1] Gillick D, Faria A, DeNero J. Mapreduce: Distributed computing for machine learning[J]. Berkley, Dec, 2006, 18.

[2] Cao Xu. Research and Improvement of a Massive Log Data Processing Model on the Hadoop Platform [D] Zhejiang University of Technology, 2013.

[3] Lu Jiang, Li Yun Research on Feature Selection Parallelization Based on MapReduce [J] Computer Science, 2015, 42 (08): 44-47.

[4] Li Zhao, Li Xiao, Wang Chunmei, et al Research on a MapReduce based text clustering method [J] Computer Science, 2016, 43 (01): 246-250.

[5] He Qing, Li Ning, Luo Wenjuan, et al Overview of Machine Learning Algorithms under Big Data [C]//Conference on Artificial Intelligence of the Chinese Computer Society two thousand and thirteen

[6] Lin J, Dyer C. Data-intensive text processing with MapReduce[M]. Springer Nature, 2022.

[7] Shim, Kyuseok. "MapReduce algorithms for big data analysis." International Workshop on Databases in Networked Information Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[8] Karloff H, Suri S, Vassilvitskii S. A model of computation for mapreduce[C]//Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2010: 938-948.

[9] Wolfe J A, Haghighi A, Klein D. Fully distributed EM for very large datasets[C]//ICML. 2008, 8: 1184-1191.

[10] .Su Jiageng. Distributed EM Clustering Algorithm Based on Hadoop Platform [D] Hebei Normal University, 2014.

[11] Gupta M R, Chen Y. Theory and use of the EM algorithm[J]. Foundations and Trends® in Signal Processing, 2011, 4(3): 223-296.

[12] Wang Aiping, Zhang Gongying, Liu Fang Research and Application of EM Algorithm [J] Computer Technology and Development, 2009, 19 (9): 108-110.