Classification of DNA Sequences into Gene Families Using Hidden Markov Models

MOHAMED MOUHAJIR
ENSIAS, Mohammed V University
Rabat, Morocco,
mohamed_mouhajir@um5.ac.ma

MOHAMMED NECHBA
ENSIAS, Mohammed V University
Rabat, Morocco,
mohammed_nechba@um5.ac.ma

MOHAMED NAOUM
ENSIAS, Mohammed V University
Rabat, Morocco,
mohamed.naoum@ensias.um5.ac.ma

***Abstract*—This report presents the work conducted as part of the project on classifying DNA sequences into their corresponding gene families. The main contribution of this project is the implementation of Hidden Markov Model (HMM) models, including the Forward-Backward algorithm, Baum-Welch algorithm, and Viterbi algorithm ([3], [2]). These algorithms were applied to the problem of classifying DNA sequences into their respective gene families.**

***Index Terms*—Hidden Markov Model (HMM), Forward-Backward algorithm, Baum-Welch algorithm, Viterbi algorithm, DNA sequences classification.**

I. Introduction

DNA sequences encode the genetic characteristics of living organisms and understanding their structure and function is essential in various biological applications. The classification of DNA sequences into gene families is a challenging task due to the high dimensionality of the data and the presence of subtle patterns that differentiate between different gene families.

In this project, we focused on implementing HMM models and applying them to the classification of DNA sequences. The HMM framework provided a powerful tool to capture the underlying structure and dependencies within DNA sequences. By utilizing the Forward-Backward algorithm, we were able to estimate the likelihood of observing a particular DNA sequence given the model parameters. The Baum-Welch algorithm was then employed to iteratively update the model parameters based on the observed data. Finally, the Viterbi algorithm was utilized to predict the most probable gene family for new DNA sequences.

A. Methodology

- **Data Collection**: We obtained a dataset consisting of DNA sequences along with their corresponding gene family labels. The dataset was carefully curated and annotated by domain experts.
- **HMM Model Implementation**: We implemented the HMM models using the Forward-Backward algorithm to calculate the likelihood of observing a sequence and the Viterbi algorithm to predict the most probable gene family for a given sequence.
- **Data Preprocessing**: Prior to model training, the DNA sequences were preprocessed by converting them into a suitable numerical representation, such as one-hot encoding. The dataset was then split into training and testing sets to evaluate the performance of the trained models.
- **Model Training and Parameter Estimation**: The Baum-Welch algorithm was applied to estimate the model parameters, including transition probabilities and emission probabilities, from the training data. This iterative process allowed the model to learn the underlying patterns and dependencies in the DNA sequences.
- **Prediction of Gene Families**: Once the HMM models were trained, the Viterbi algorithm was used to predict the gene family for new DNA sequences. This algorithm determined the most likely sequence of hidden states (gene families) based on the observed sequence.

II. Exploring Hidden Markov Model Algorithms: From Forward-Backward to Viterbi

A. Forward

The forward algorithm is a computational procedure used in Hidden Markov Models (HMMs) to calculate the probability of observing a given sequence of symbols. It takes into account the underlying states of the system and calculates forward probabilities at each time step. By recursively updating these probabilities, the algorithm provides a measure of the likelihood of the observed sequence based on the model's parameters. The forward algorithm is an essential tool for various tasks in HMMs, such as sequence classification and parameter estimation.

Below, we define the formal pseudo code for the forward algorithm :

Algorithm 1: Forward Algorithm

```
INPUT
  • $O$: Sequence of observations
  • $\pi$: Initial state distribution
  • $A$: Transition matrix
  • $B$: Emission matrix
Result: $\alpha$: Forward probabilities
Initialize $\alpha_1(i) = \pi(i) \cdot B(i, O_1)$ for all states $i$;
for $t = 2$ to $T$ do
    for each state $j$ do
        $\alpha_t(j) = \left(\sum_i \alpha_{t-1}(i) \cdot A(i, j)\right) \cdot B(j, O_t)$;
    end
end
Return $\alpha$
```

B. Backward Algorithm

The backward algorithm is a computational procedure used in Hidden Markov Models (HMMs) to calculate the probability of observing a future sequence of symbols given the current state. It works in the opposite direction of the forward algorithm, starting from the last observation and moving backwards in time. By recursively updating backward probabilities, it provides information about the likelihood of future observations given the current state. The backward algorithm is a key component of HMMs, used in tasks such as sequence alignment, parameter estimation, and decoding.

Algorithm 2: Backward Algorithm

```
INPUT
  • $O$: Sequence of observations
  • $A$: Transition matrix
  • $B$: Emission matrix
Result: $\beta$: Backward probabilities
Initialize $\beta_T(i) = 1$ for all states $i$;
for $t = T - 1$ to $1$ do
    for each state $i$ do
        $\beta_t(i) = \sum_j (A(i, j) \cdot B(j, O_{t+1}) \cdot \beta_{t+1}(j))$;
    end
end
Return $\beta$
```

C. forward-backward

The forward-backward algorithm is a dynamic programming algorithm used in the context of Hidden Markov Models (HMMs). It allows us to compute the probability of observing a sequence of observations given the model parameters.

In the forward step, the algorithm calculates the probability of being in each state at each time step, taking into account the previous state probabilities and the transition and emission probabilities. This information is then used to compute the probability of the entire observation sequence.

The backward step, on the other hand, computes the probability of observing the remaining part of the sequence starting from each state at each time step. It uses similar recursive computations as the forward step but in the opposite direction.

By combining the forward and backward probabilities, the algorithm provides an estimation of the likelihood of the observed sequence. This estimation is useful for tasks such as parameter estimation, sequence alignment, and decoding in HMM-based applications.

Overall, the forward-backward algorithm is a fundamental tool for analyzing and working with Hidden Markov Models, providing insights into the probabilities and dependencies of observed sequences

Algorithm 3: Forward-Backward Algorithm

```
Input:
  • $O$: Observation sequence of length $T$
  • $\lambda$: HMM model parameters ($A$, $B$, $\pi$)
Output:
  • $P$: Probability of observing the sequence $O$ given the
    model $\lambda$
Forward Procedure:
  1) Initialize the forward variables $\alpha$ with
     $\alpha_1(i) = \pi_i \cdot b_i(O_1)$ for all states $i$.
  2) Recursively compute the forward variables
     $\alpha_t(i) = \left(\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ji}\right) \cdot b_i(O_t)$ for $t = 2$ to $T$
     and all states $i$.
  3) Compute the probability of the observation sequence
     as $P = \sum_{i=1}^{N} \alpha_T(i)$.
Backward Procedure:
  1) Initialize the backward variables $\beta$ with $\beta_T(i) = 1$ for
     all states $i$.
  2) Recursively compute the backward variables
     $\beta_t(i) = \sum_{j=1}^{N} (\beta_{t+1}(j) \cdot a_{ij} \cdot b_j(O_{t+1}))$ for $t = T - 1$
     to 1 and all states $i$.
Output:
  • $P$: Probability of observing the sequence $O$ given the
    model $\lambda$.
```

D. Baum-Welch Algorithm

The Baum-Welch algorithm, also known as the expectation-maximization algorithm, is an iterative procedure used to estimate the parameters of a hidden Markov model (HMM) based on observed sequences. It is an unsupervised learning algorithm that aims to maximize the likelihood of the observed data given the HMM model.

```
Algorithm 4: Baum-Welch Algorithm
Input:
  • $O$: Set of observation sequences
  • $N$: Number of states in the HMM
  • $M$: Number of possible observations
  • $A$: Transition probability matrix
  • $B$: Emission probability matrix
  • $\pi$: Initial state distribution
  • $nb\_iter$: number of iteration
Output:
  • $A'$: Updated transition probability matrix
  • $B'$: Updated emission probability matrix
Initialization: Initialize $A'$, $B'$, and $\pi'$ with random values
for $iter \in \llbracket 1, nb\_iter \rrbracket$ do
    for $o \in O$ do
        • $\alpha = \text{forward}(o, A', B', \pi')$
        • $\beta = \text{backward}(o, A', B', \pi')$
        • Compute the state occupation and state transition probabilities:
          for $t \in \llbracket 1, M \rrbracket$ do
              for $i \in \llbracket 1, N \rrbracket$ do
                  $$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$
                  for $j \in \llbracket 1, N \rrbracket$ do
                      $$\xi_t(i,j) = \frac{\alpha_t(i)A_{ij}B_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$$
                  end
              end
          end
        for $i \in \llbracket 1, N \rrbracket$ do
            for $j \in \llbracket 1, N \rrbracket$ do
                $$A'_{ij} = \frac{\sum_{t=1}^{M-1} \xi_t(i,j)}{\sum_{t=1}^{M-1} \sum_m \xi_t(i,m)}$$
                for $k \in \llbracket 1, M \rrbracket$ do
                    $$B'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j)\delta_{o_t,k}}{\sum_{t=1}^{T} \gamma_t(j)}$$
                end
            end
        end
        for $i \in \llbracket 1, N \rrbracket$ do
            $$\pi'_i = \frac{\gamma_1(i)}{\sum_O 1}$$
        end
    end
end
Return $A'$, $B'$
```

E. Viterbi Algorithm

The Viterbi algorithm is a programming technique used to find the most likely sequence of hidden states in a Hidden Markov Model (HMM) given an observed sequence. It efficiently calculates the probabilities of different state sequences by considering the previous state probabilities and transition probabilities. By iteratively determining the most probable path to each state at each time step, the algorithm selects the state with the highest probability and traces back to retrieve the sequence of hidden states with the maximum likelihood. The Viterbi algorithm is widely applied in fields like speech recognition, natural language processing, and bioinformatics for tasks such as part-of-speech tagging and gene sequence analysis due to its efficiency and accuracy in decoding hidden states in HMMs.

```
Algorithm 5: Viterbi Algorithm
Input:
  • $O$: Observed sequence
  • $N$: Number of states in the HMM
  • $S$: Set of states
  • $A$: State transition matrix
  • $B$: Emission probability matrix
  • $\pi$: Initial state probabilities
Output:
  • $Q^*$: Most likely sequence of hidden states
Initialization:
for $i \in \llbracket 1, N \rrbracket$ do
    $$V_1(i) = \pi(i) \cdot B(i, O_1)$$
    $$T_1(i) = 0$$
end
Recursion:
for $t \in \llbracket 1, M \rrbracket$ do
    for $j \in \llbracket 1, N \rrbracket$ do
        $V_t(j) = \max_{1 \le i \le N}(V_{t-1}(i) \cdot A(i,j) \cdot B(j, O_t))$
        $T_t(j) = \arg\max_{1 \le i \le N}(V_{t-1}(i) \cdot A(i,j))$
    end
end
Termination:
    $$P^* = \max_{1 \le i \le N} V_T(i)$$
    $$Q^*_T = \arg\max_{1 \le i \le N} V_T(i)$$
Path Backtracking:
for $t \in \llbracket 1, M-1 \rrbracket$ do
    $Q^*_t = T_{t+1}(Q^*_{t+1})$
end
Return $Q^*$
```

III. Experimentation

A. Description of DNA Sequence Classification Problem

The problem in our dataset is to classify DNA sequences into their corresponding gene families. We have a dataset that contains DNA sequences along with the associated class or gene family for each sequence.

Our objective is to develop a machine learning model capable of learning to recognize specific motifs or characteristics in DNA sequences that can predict which gene family a given sequence belongs to.

Classifying gene families from DNA sequences is a complex task for several reasons. Firstly, DNA sequences are high-dimensional data, and there can be significant variability within sequences even within the same gene family. Secondly, there may exist subtle motifs or features that differentiate between different gene families.

By using machine learning techniques such as Hidden Markov Models (HMMs) and the Baum-Welch algorithm, we can construct a model that learns to represent and exploit the relationships between DNA sequences and gene families. The Baum-Welch algorithm will allow us to estimate the parameters of the HMM model from the observed data, while the Viterbi algorithm will enable us to predict the corresponding class for a new DNA sequence.

The ultimate goal of this problem is to achieve accurate classification of DNA sequences into their corresponding gene families, which can enhance our understanding of gene function and structure, as well as identify relationships with specific diseases or characteristics.

It is important to note that the success of the classification will depend on the quality of the training data, the representativeness of the different gene families in the dataset, and the appropriate choice of model and learning parameters.

B. Description of the data

The dataset available at [1] consists of DNA sequences and their corresponding labels. The DNA sequences are extracted from various organisms, including humans, animals, and plants. Each sequence is represented as a string of nucleotide bases (A, C, G, and T).

The dataset is labeled, with each DNA sequence assigned a specific category or class. These labels represent different attributes associated with the DNA sequences, such as the organism type, gene function, or genetic mutation.

This dataset provides a valuable resource for exploring and analyzing DNA sequences and their relationships with various biological phenomena. It can be used for tasks such as DNA sequence classification, gene function prediction, or investigating genetic variations across different organisms.

TABLE I
DESCRIPTION OF THE DNA CLASSIFICATION DATASET

Gene family	**Number**	Class label
G protein coupled receptors	531	0
Tyrosine kinase	534	1
Tyrosine phosphatase	349	2
Synthetase	672	3
Synthas	711	4
lon channel	240	5
Transcription factor	1343	6

C. Results

For a fixed number of iterations (2), the achieved accuracy was only 9%, indicating a significantly low performance. However, the potential for improvement exists through hyper-parameter tuning, a step that was not feasible in our current study. To address this limitation, one viable approach is the utilization of parallelization techniques and the implementation of parallel versions of these algorithms, particularly on hardware accelerators such as GPUs. This strategy can enhance computational efficiency and potentially lead to more accurate classifications in DNA sequence analysis tasks.

IV. Conclusion

In conclusion, this project implemented Hidden Markov Model (HMM) models and their associated algorithms, including the Forward-Backward, Baum-Welch, and Viterbi algorithms, for the classification of DNA sequences into gene families. The models provided insights into DNA sequence analysis and have potential applications in gene function prediction, disease-related gene identification, and understanding biological processes. Further improvements can be made in terms of computational efficiency and accuracy. Overall, this project contributes to the field of bioinformatics and lays the groundwork for future research in DNA sequence classification.

References

[1] DNA Sequence Dataset. https://www.kaggle.com/datasets/nageshsingh/dna-sequence-dataset. Accessed on 2023-05-07.

[2] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE signal processing letters*, 10(1):11–14, 2003.

[3] Shun-Zheng Yu and Hisashi Kobayashi. Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE Transactions on Signal Processing*, 54(5):1947–1951, 2006.