Hidden Markov Model

10/31/2023

Agenda

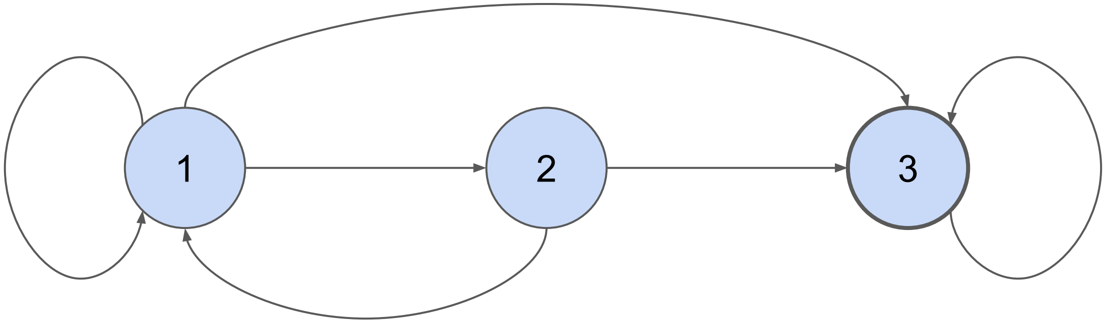
- Announcements
    - Project page will be public on our course website today
    - Some updates will be added in the future
- Main topic: Hidden Markov Model

A Simple State Machine

A state machine is a machine's AI logic in graph form.

Each node is a state, each directed edge represents a path from one state to another.

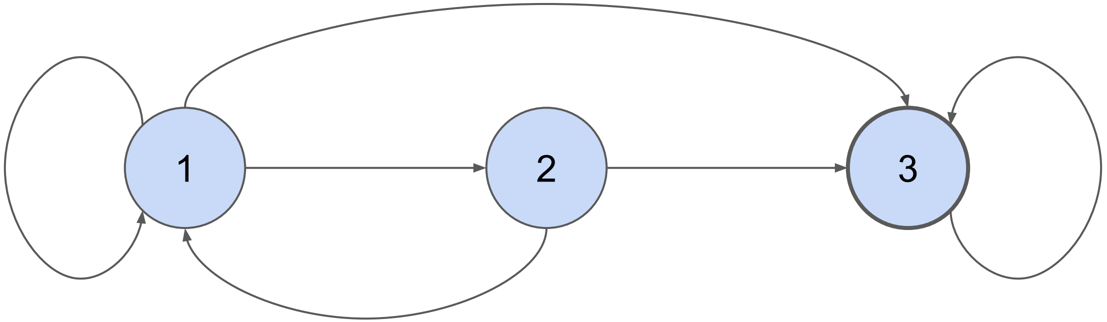
The initial state can be any state, but there is only one terminal state: the AI can only stop and return the output at the terminal state.



Finite State Machine/Automata

If there are finitely many states, the structure is called **Finite State Machine** (**FSM**) or **Finite State Automata** (**FSA**).

FSM/FSA is widely used in computational linguistics.



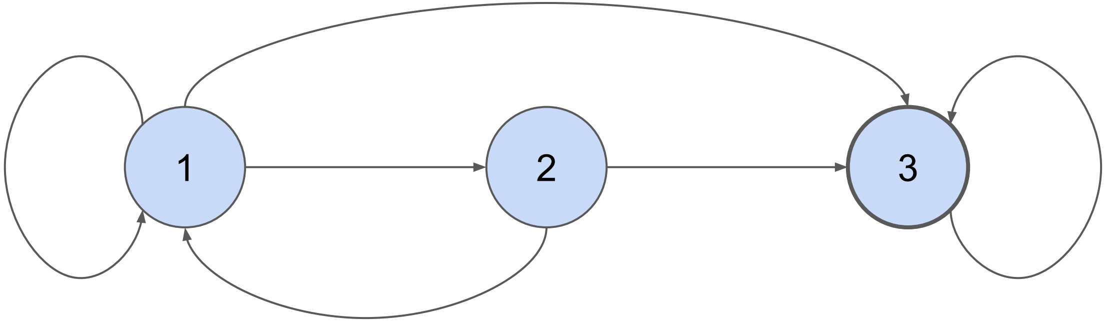
Finite State Machine/Automata

Each state can output a **symbol**. Symbols can be characters, words, tags, etc..

The symbols in a FSM/FSA form a **vocabulary**.

Then, a FSM/FSA can generate all possible sequences of the symbols in the vocabulary. The set of all these sequences is called **language**.

We can also verify if a sequence can be generated by a FSM/FSA.

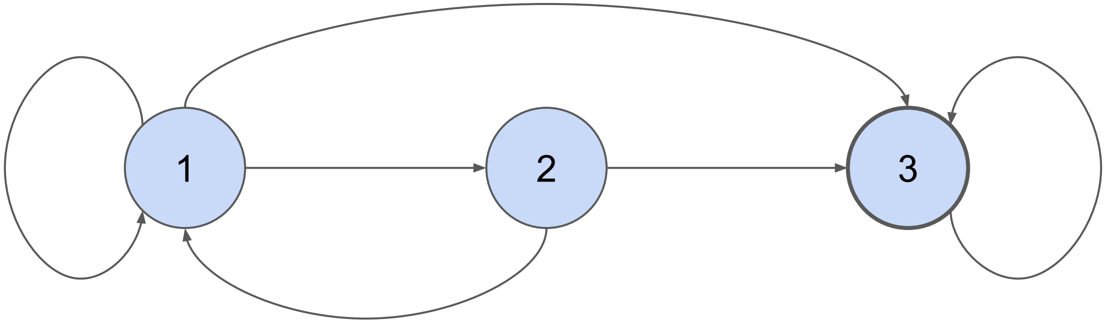


Finite State Machine/Automata

The simple structure of FSM/FSA limits its application in modeling.

When used as a generator, it outputs a possibly infinite set.

When used as a recognizer, it can only return True of False.

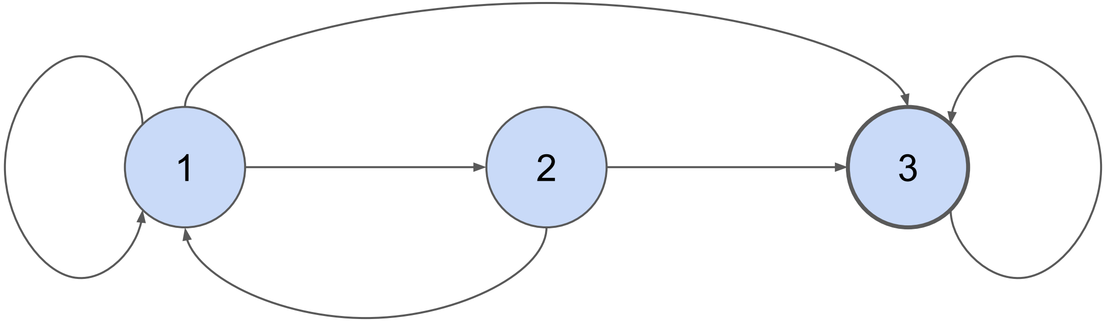


Finite State Machine/Automata

However, we often want more than these.

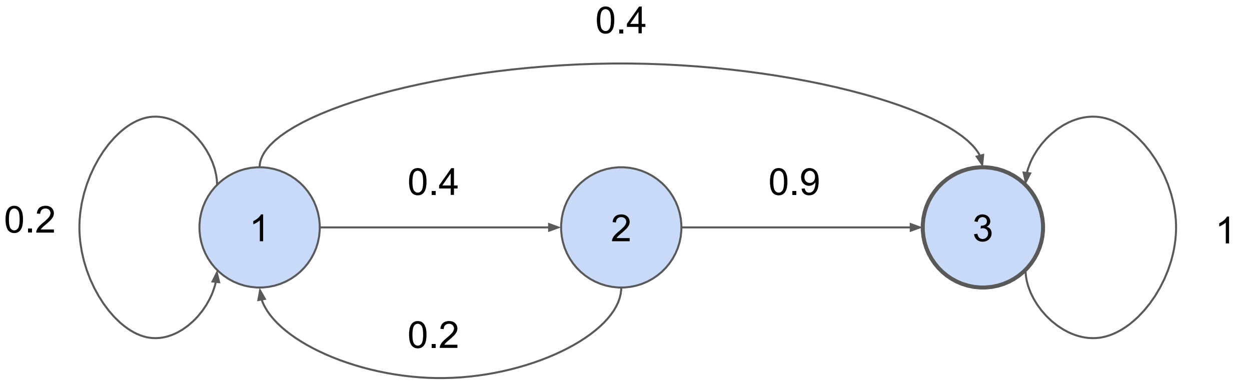
For example, what are the odds of the sequences and what are the likelihoods of True or False.

This leads to the adaptation of probabilities to the transitions.



Markov Model

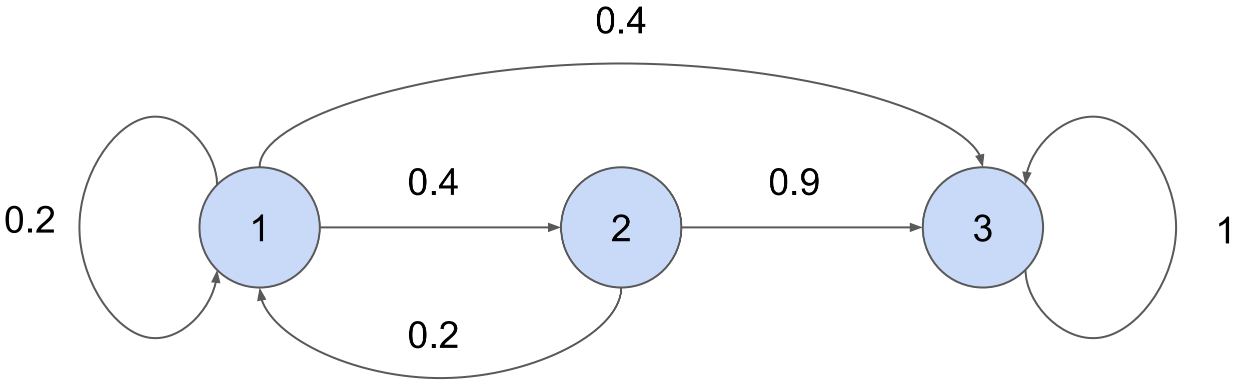
The resulting model is a **Markov Model**, as each transition depends on the outgoing state but not the ones prior to it, and the sum of all outgoing transition probabilities of any state is one.



Markov Model

The resulting model is a **Markov Model**, as each transition depends on the outgoing state but not the ones prior to it, and the sum of all outgoing transition probabilities of any state is one.

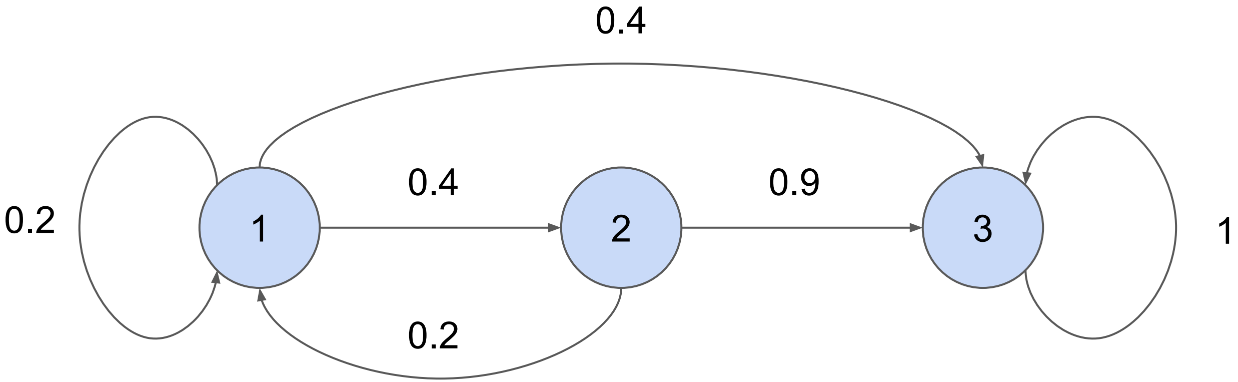
Furthermore, we can make it more robust by allowing termination at any state.



(Hidden) Markov Model

At this point, each state outputs and must output one symbol, making the state outputs deterministic (**observable**).

However, if instead, each state can output any symbol with certain probability, then the outputs are non-deterministic (**hidden**) and the resulting model is called **Hidden Markov Model** (**HMM**).



HMM - What It Is

At a high level, a HMM is a Markov model with Markov transition process and non-observable (hidden) states.

Note that the sum of output probabilities in each state must be one.

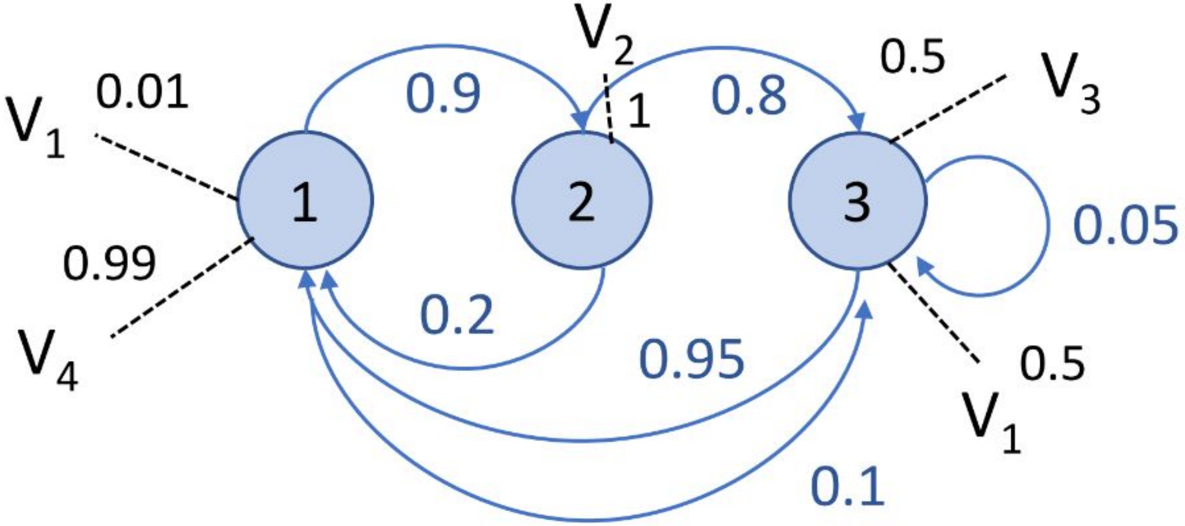


Image source: Prof. Betke’s original presentation

HMM - How It Works

1. Choose a random state by some initial probability
2. Choose and output a symbol by some symbol probability in the current state
3. Choose the next state by some transitional probability and repeat from 2 till satisfied

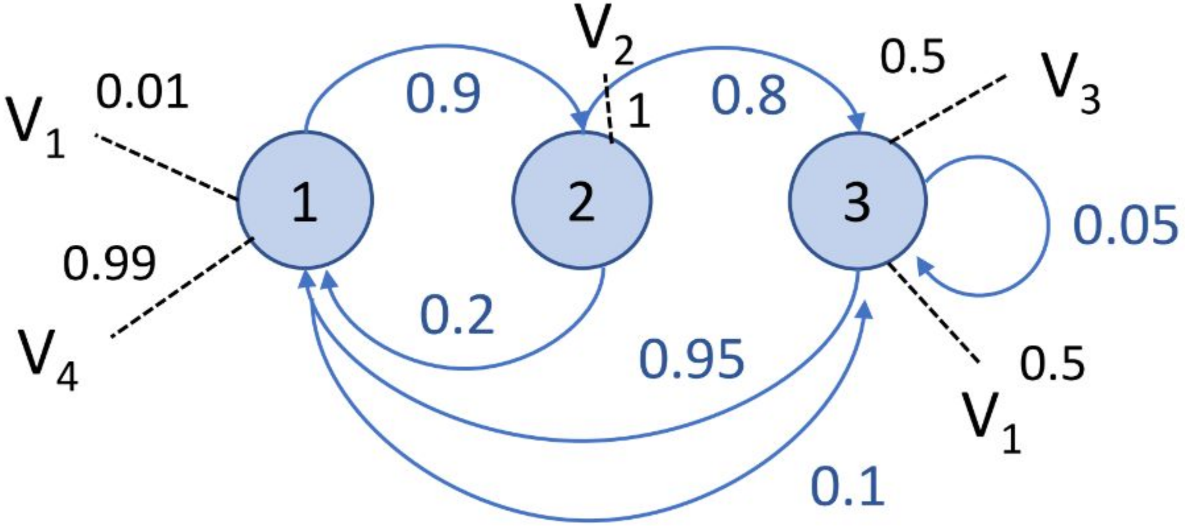


Image source: Prof. Betke’s original presentation

HMM Applications

- Weather forecasting
- Financial analysis
- Part of speech tagging

HMM Notations

The following notations are commonly used in HMM literature:

- States and the number of states: S = [S_1, S_2, S_3, …, S_N]
- State at time t: q_t
- Vocabulary, symbols, and the size of vocabulary: V = [V_1, V_2, V_3, …, V_M]
    - sometimes symbols are in lower case
- Transition probabilities as a matrix: A = {a_{ij}}
- The probability to yield symbol k in state j: B = {$b_j(k)$} = {P(V_k at j at t | $q_t = S_j$)}
- Initial probabilities: $\boldsymbol{\pi}$ = [$\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_3$, …, $\boldsymbol{\pi}_N$]
- Sequence of observations (observed symbols) at time T: O = $O_1O_2O_3 \ldots O_T$
- Everything above except for q_t and O: $\boldsymbol{\lambda}$ = (S, V, A, B, $\boldsymbol{\pi}$) or more commonly just (A, B, $\boldsymbol{\pi}$)

HMM Example - States and Symbols

$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

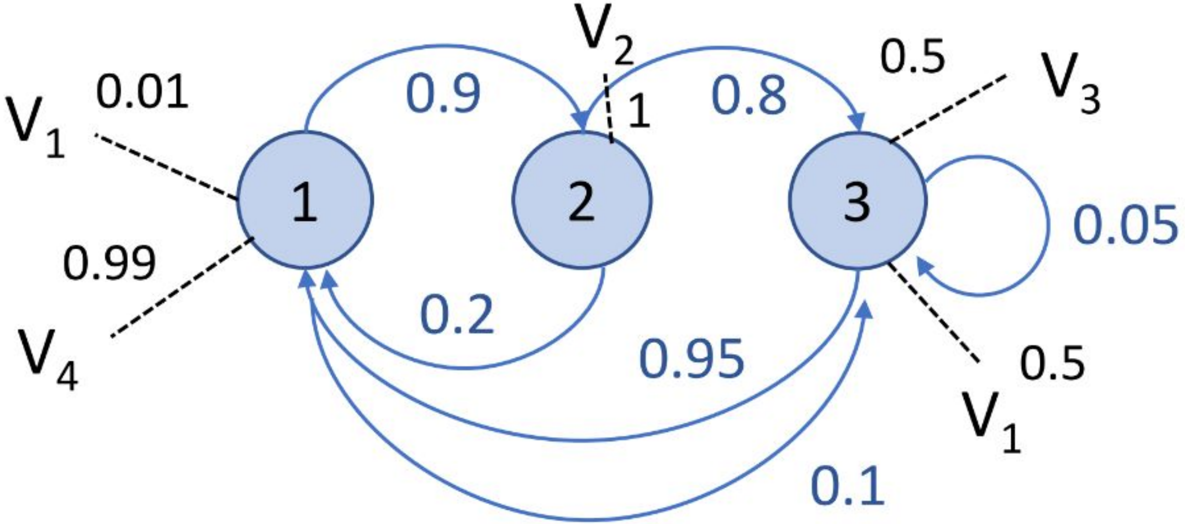


Image source: Prof. Betke’s original presentation

HMM Example - Transitions

A[1, :] = [0, 0.9, 0.1], A[2, :] = [0.2, 0, 0.8], A[3, :] = [0.95, 0, 0.05]

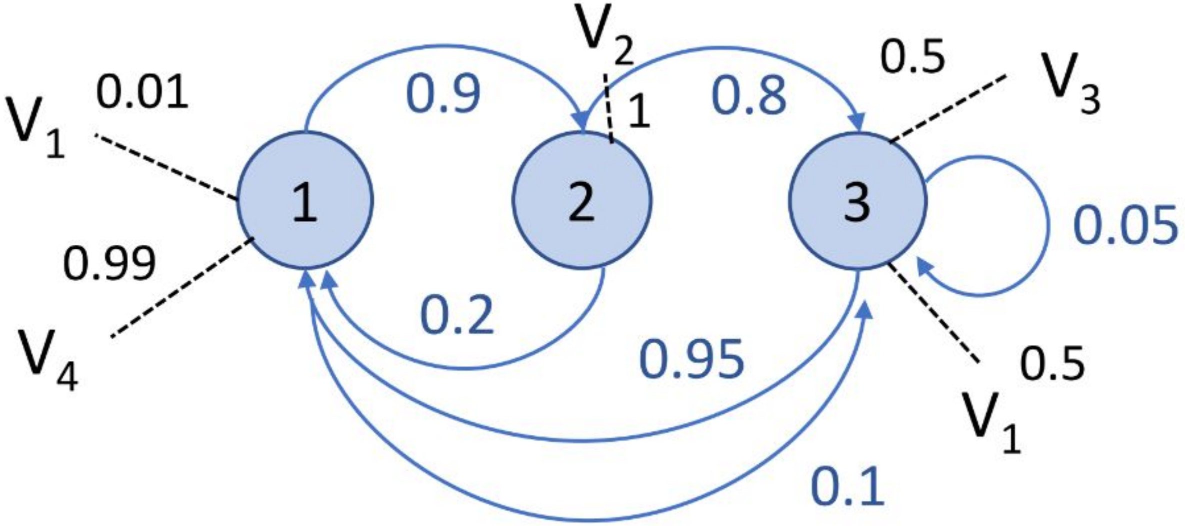


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

$b_1(V_1) = 0.01$, $b_1(V_2) = 0$, $b_1(V_3) = 0$, $b_1(V_4) = 0.99$

$b_2(V_1) = 0$, $b_2(V_2) = 1$, $b_2(V_3) = 0$, $b_2(V_4) = 0$

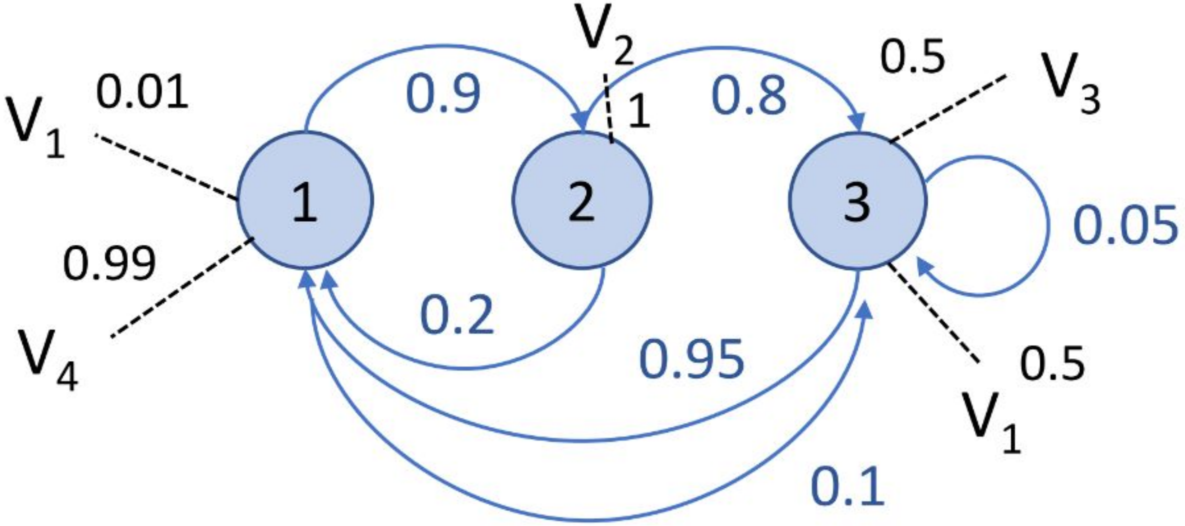


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P(V_2)?

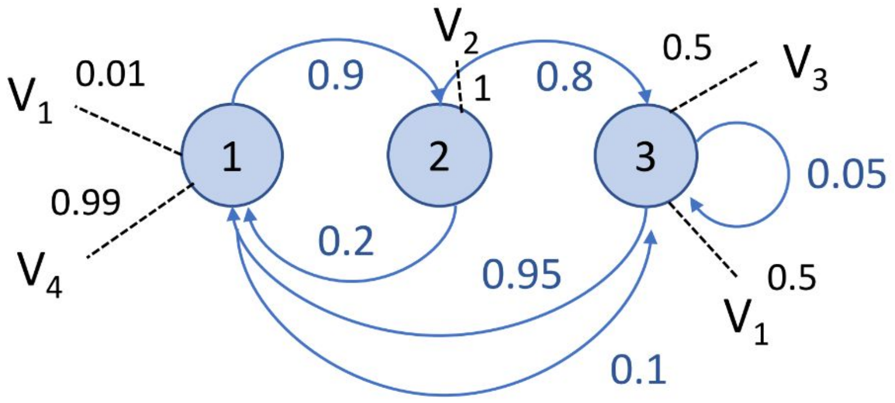
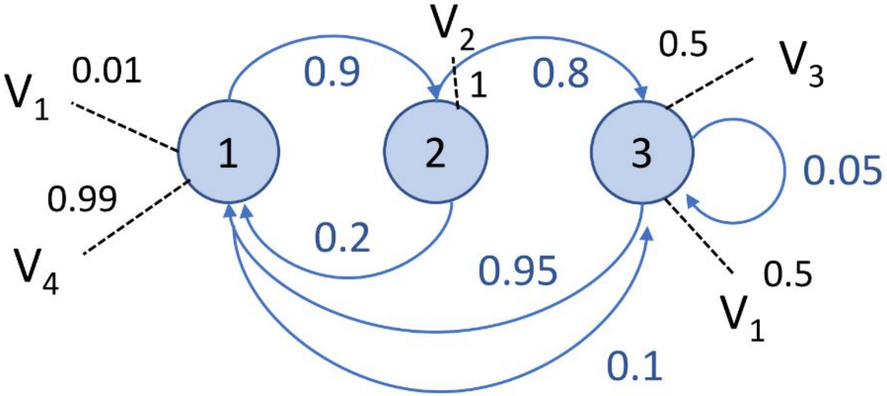


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose $\boldsymbol{\pi}$ = [0.5, 0.2, 0.3]

What is P(O) = P(V_2)?



$$\begin{aligned} P(V_2) &= \pi_1 \cdot b_1(V_2) + \pi_2 \cdot b_2(V_2) + \pi_3 \cdot b_3(V_2) \\ &= 0 + 0.2 \cdot 1 + 0 \\ &= 0.2 \end{aligned}$$

Image source: Prof. Betke's original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P(V_2V_2)?

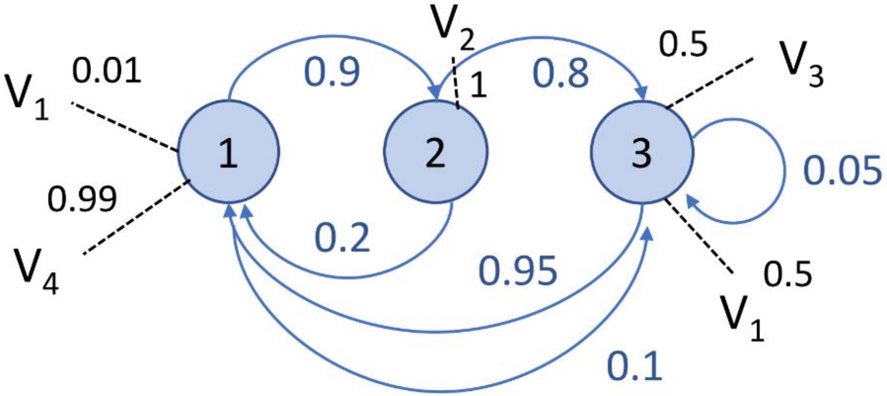
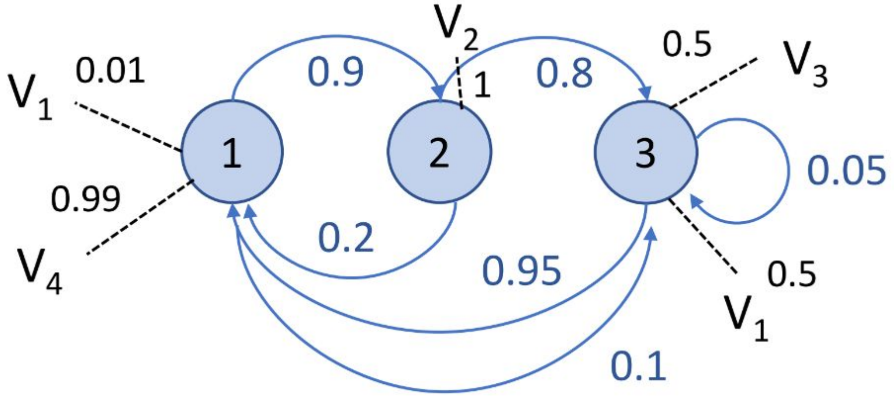


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P(V_2V_2)?



$$\begin{aligned}P(V_2V_2) &= \pi_1 \cdot b_1(V_2) \cdot a_{11} \cdot b_1(V_2) + \pi_1 \cdot b_1(V_2) \cdot a_{12} \cdot b_2(V_2) + \pi_1 \cdot b_1(V_2) \cdot a_{13} \cdot b_3(V_2) + \\ &\quad \pi_2 \cdot b_2(V_2) \cdot a_{21} \cdot b_1(V_2) + \pi_2 \cdot b_2(V_2) \cdot a_{22} \cdot b_2(V_2) + \pi_2 \cdot b_2(V_2) \cdot a_{23} \cdot b_3(V_2) + \\ &\quad \pi_3 \cdot b_3(V_2) \cdot a_{31} \cdot b_1(V_2) + \pi_3 \cdot b_3(V_2) \cdot a_{32} \cdot b_2(V_2) + \pi_3 \cdot b_3(V_2) \cdot a_{33} \cdot b_3(V_2) \\ &= 0 + 0 + 0 \\ &= 0\end{aligned}$$

Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P(V_2V_1)?

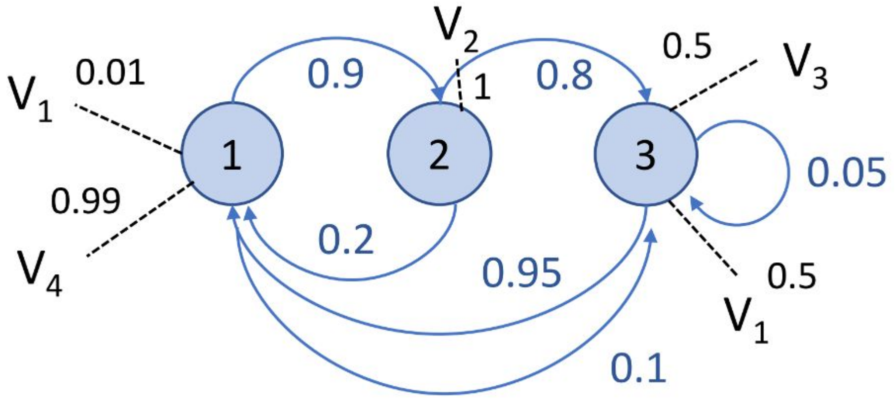
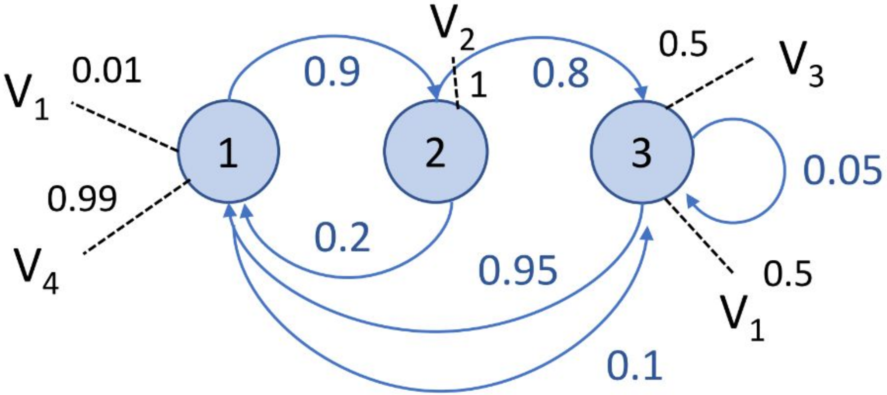


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P(V_2V_1)?



$$P(V_2V_1) = \pi_1 \cdot b_1(V_2) \cdot a_{11} \cdot b_1(V_1) + \pi_1 \cdot b_1(V_2) \cdot a_{12} \cdot b_2(V_1) + \pi_1 \cdot b_1(V_2) \cdot a_{13} \cdot b_3(V_1) +$$
$$\pi_2 \cdot b_2(V_2) \cdot a_{21} \cdot b_1(V_1) + \pi_2 \cdot b_2(V_2) \cdot a_{22} \cdot b_2(V_1) + \pi_2 \cdot b_2(V_2) \cdot a_{23} \cdot b_3(V_1) +$$
$$\pi_3 \cdot b_3(V_2) \cdot a_{31} \cdot b_1(V_1) + \pi_3 \cdot b_3(V_2) \cdot a_{32} \cdot b_2(V_1) + \pi_3 \cdot b_3(V_2) \cdot a_{33} \cdot b_3(V_1)$$
$$= 0 + (0.2 * 1 * 0.2 * 0.01 + 0 + 0.2 * 1 * 0.8 * 0.5) + 0$$
$$= 0.0804$$

Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P($V_2V_1V_3$)?

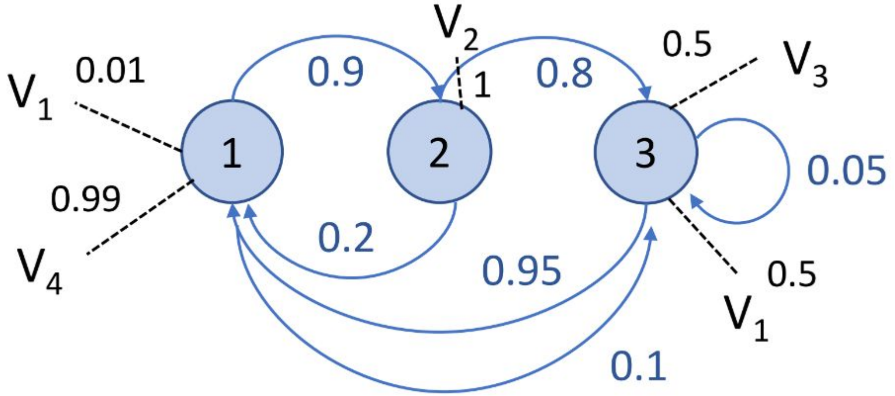
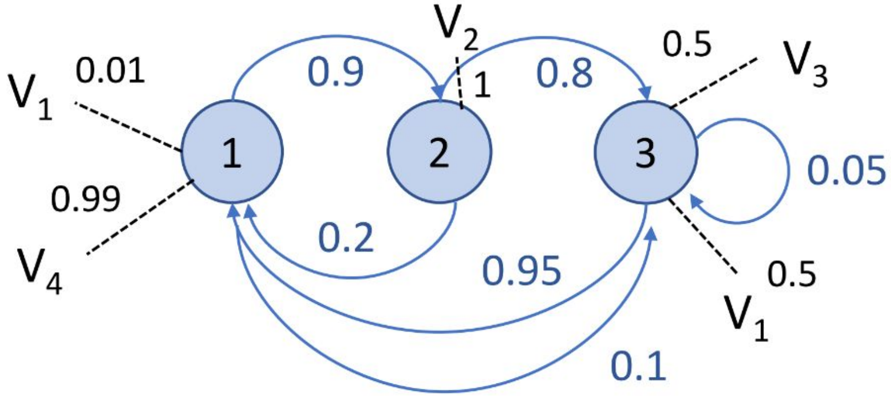


Image source: Prof. Betke’s original presentation

HMM Example - Symbol Probabilities

Suppose π = [0.5, 0.2, 0.3]

What is P(O) = P($V_2V_1V_3$)?



In your homework/exam, you only need to consider the non-zero terms.

Image source: Prof. Betke’s original presentation

HMM - Three Problems

There are three basic questions to answer about HMM before we can apply it.

- Evaluation problem
  - Given a sequence of observations O and the model $\boldsymbol{\lambda}$, what is $P(O \mid \boldsymbol{\lambda})$?
- Recognition problem
  - Given a sequence of observations O and the model $\boldsymbol{\lambda}$, what is the optimal state sequence $Q = q_1 q_2 q_3 \ldots q_T$?
- Learning/training problem
  - Given a sequence of observations O and the model $\boldsymbol{\lambda}$, how to adjust $\boldsymbol{\lambda}$ to maximize $P(O \mid \boldsymbol{\lambda})$?

HMM Problem 1: Evaluation Problem

We have just seen how fast the math gets ugly.

In general, the formula is

$$P(O|\lambda) = \sum_{Q} P(O|Q, \lambda) P(Q|\lambda)$$

where Q is a sequence of states.

The time complexity is exponential.

HMM Problem 1: Evaluation Problem

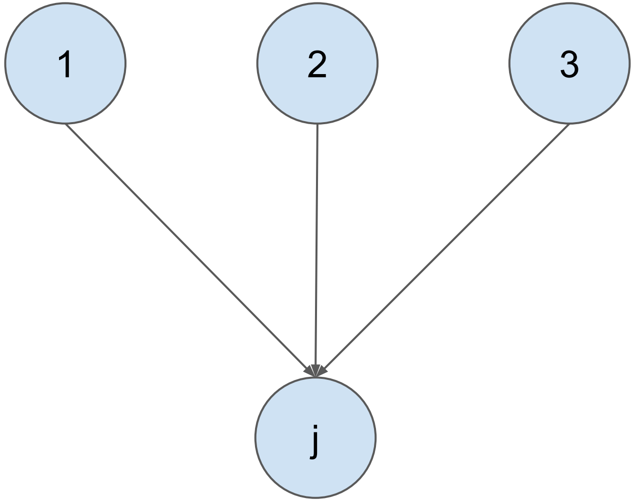
Suppose that for each state S_i, we know the quantity

$$\alpha_t(i) = P(O_1 O_2 ... O_t, q_t = S_i | \lambda)$$

That is, suppose we know the probability that the model outputs the partial sequence $O_1 O_2 … O_t$ when it reaches state S_i at time t for all states S_i, can we compute $\alpha_{t + 1}$(j) for some state S_j?

$$\alpha_{t+1}(j) = (\sum_i \alpha_t(i) \cdot a_{ij}) \cdot b_j(O_{t+1})$$

HMM Problem 1: Evaluation Problem



$$\alpha_t(i) = P(O_1 O_2 ... O_t, q_t = S_i | \lambda)$$

$$\alpha_{t+1}(j) = (\sum_i \alpha_t(i) \cdot a_{ij}) \cdot b_j(O_{t+1})$$

HMM Problem 1: Evaluation Problem

This observation leads to an inductive algorithm.

Initially, for all *i*

$$\alpha_1(i) = \pi_i b_i(O_1)$$

Next, for all *j* at each step *t* < *T*,

$$\alpha_{t+1}(j) = (\textstyle\sum_{i=1}^{N} \alpha_t(i)a_{ij})b_j(O_{t+1})$$

Finally, when *t* = *T*

$$P(O|\lambda) = \textstyle\sum_{i=1}^{N} \alpha_T(i)$$

Time complexity? Polynomial!

HMM Problem 1: Evaluation Problem

This is called the **forward procedure**.

Consequently, there is a **backward procedure**.

HMM Problem 1: Evaluation Problem

Define

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T|q_t = S_i, \lambda)$$

This is the probability of observing the future subsequence $O_{t+1}O_{t+2}\ldots O_T$, given that the current state is S_i at time step t.

It can be computed using future β_{t+1} (suppose we know these future values) as follows.

$$\beta_t(i) = \sum_{j=1} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

HMM Problem 1: Evaluation Problem

Therefore, the backward procedure can be described as follows.

Initially, for all *i*

$$\beta_T(i) = 1$$

Next, for all *i* at each step ***t*** = ***T - 1**, **T - 2**, **…**, **1***,

$$\beta_t(i) = \sum_{j=1} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Finally**,**

$$P(O|\lambda) = \sum_{i} \pi_i b_i(O_1) \beta_1(i)$$

HMM Problem 1: Evaluation Problem

Together these two are called the **Forward-Backward Procedure**.

Note that their outputs must be the same, as they are computing the same quantity from different directions.

HMM Problem 2: Recognition

Problem statement:

How to measure optimality?

Given a sequence of observations O and the model $\boldsymbol{\lambda}$, what is the optimal state sequence $Q = q_1q_2q_3\ldots q_T$?

Before trying to answer this question, what is the right question to ask?

HMM Problem 2: Recognition

The most common measurement: $P(Q \mid O, \boldsymbol{\lambda})$, the probability of the state sequence (or **path**) given the observed symbol sequence.

In other words, to solve problem 2, we want to find Q^* such that $P(Q^* \mid O, \boldsymbol{\lambda})$ is maximized.

But this requires looping through all possible paths, which we have known is not feasible.

However, since $P(Q \mid O, \boldsymbol{\lambda})\ P(O \mid \boldsymbol{\lambda}) = P(Q, O \mid \boldsymbol{\lambda})$, we can use $P(Q, O \mid \boldsymbol{\lambda})$ instead.

Optimal Path

Let’s take a break by considering this simply problem.

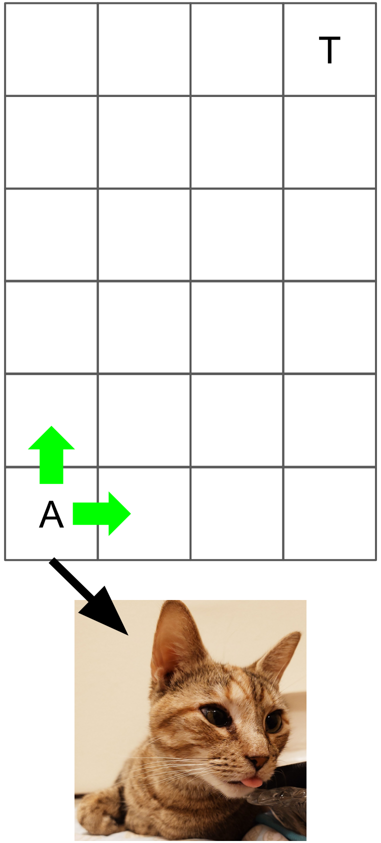
Say Anya hits the wall while chasing a ghost in the house and got isekai’ed to a grid world.

She is initially located at the bottom-left corner and she finds her favourite treat at the top-right corner.

The only available actions are up and right.

Moving between blocks incurs a cost and the costs are not uniform.

How to find the path with the lowest cost?



HMM Problem 2: Recognition

New problem statement:

Given an observation sequence O and model **λ**, what is the state sequence Q that maximizes the probability P(Q, O | **λ**)?

Define the best score representation

$$\delta_t(i) = \max_{Q_{t-1}} P(Q_{t-1}, q_t = S_i, O_t | \lambda)$$

This is the highest probability of a sequence Q_t whose first t - 1 terms are Q_{t-1}, the current state is S_i, and the outputs are O_t. Therefore,

$$P(Q^*, O|\lambda) = \max_i \delta_T(i)$$

HMM Problem 2: Recognition

Definition:

$$\delta_t(i) = \max_{Q_{t-1}} P(Q_{t-1}, q_t = S_i, O_t|\lambda)$$

Initially,

$$\delta_1(i) = \pi_i b_i(O_1)$$

Then, $\boldsymbol{\delta}_T$(i) can be computed recursively:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1})$$

This is the **Viterbi** algorithm.

The path can be retrieved by backtracking the chosen states

HMM Problem 3: Learning/Training

Problem statement:

Given an observation sequence O and model $\boldsymbol{\lambda}$, how do we adjust $\boldsymbol{\lambda}$ to maximize $P(O \mid \boldsymbol{\lambda})$?

If we can solve this problem, then we can train a model starting from some random parameters.

But there is no optimal way to estimate the parameters.

One can at best use some iterative procedure to locally maximize the probabilities.

HMM Problem 3: Learning/Training

First, define a new function

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

This is the probability of being in state S_i at time t and state S_j at time t + 1, given the observations and the model.

We can sum over j

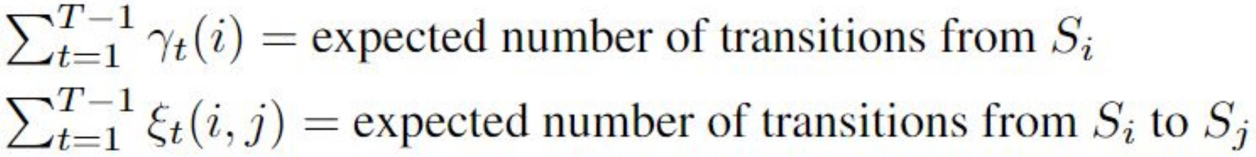
$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$

This is the probability of being in state S_i at time t, given the observations and the model.

HMM Problem 3: Learning/Training

- $\gamma_t(i)$: probability of being in state S_i at time t, given the observations and the model
- $\xi_t(i, j)$: probability of being in state S_i at time t and state S_j at time t + 1, given the observations and the model

What are the sums of these two quantities over time steps t from 1 to T - 1?



$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j$$

This is because they follow Poisson binomial distribution.

HMM Problem 3: Learning/Training

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from } S_i \text{ to } S_j$$

Wait, how do we compute this?

We can then update **λ** as

$$\overline{\pi_i} = \gamma_1(i)$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\overline{b_j(v_k)} = \frac{\sum_{t=1}^{T-1} \mathbf{1}(v_k{=}O_t)\gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

HMM Problem 3: Learning/Training

$$\begin{aligned}\xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{P(O|\lambda)} \\ &= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{\sum_i \sum_j P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}\end{aligned}$$

where

$$P(q_t = S_i, q_{t+1} = S_j, O|\lambda) = \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

HMM Problem 3: Learning/Training

The process is called “**Baum-Welch Re-estimation Algorithm**”.

It is repeated till stable.