**Continuous Observation Hidden Markov Model**

Loc Nguyen

Sunflower Soft Company, An Giang, Vietnam

## Abstract

Hidden Markov model (HMM) is a powerful mathematical tool for prediction and recognition but it is not easy to understand deeply its essential disciplines. Previously, I made a full tutorial on HMM in order to support researchers to comprehend HMM. However HMM goes beyond what such tutorial mentioned when observation may be signified by continuous value such as real number and real vector instead of discrete value. Note that state of HMM is always discrete event but continuous observation extends capacity of HMM for solving complex problems. Therefore, I do this research focusing on HMM in case that its observation conforms to a single probabilistic distribution. Moreover, mixture HMM in which observation is characterized by the mixture model of partial probability density functions is also mentioned. Mathematical proofs and practical techniques relevant to continuous observation HMM are main subjects of the research.

**Keywords**: hidden Markov model, continuous observation, mixture model, evaluation problem, uncovering problem, learning problem

## I. Hidden Markov model

The research produces a full tutorial on hidden Markov model (HMM) in case of continuous observations and so it is required to introduce essential concepts and problems of HMM. The main reference of this tutorial is the article “A tutorial on hidden Markov models and selected applications in speech recognition” by author (Rabiner, 1989). Section [I](#_III.2.4._Hidden_Markov) – the first section is summary of the tutorial on HMM by author (Nguyen, 2016) whereas sections [II](#_IV.5._Continuously_observational) and [III](#_III._Mixture_hidden) are main ones of the research. Section [IV](#_IV._Discussion_and) is the discussion and conclusion. The main problem that needs to be solved is how to learn HMM parameters when discrete observation probability matrix is replaced by continuous density function. In section [II](#_IV.5._Continuously_observational), I propose practical technique to calculate essential quantities such as forward variable *αt*, backward variable *βt*, and joint probabilities *ξt*, *γt* which are necessary to train HMM with regard to continuous observations. Moreover, from expectation maximization (EM) algorithm which was used to learn traditional discrete HMM, I derive the general equation whose solutions are optimal parameters. Such equation specified by formulas [II.5](#_Formula_IV.5.5._Equation) and [III.7](#_Formula_IV.5.13._Equation) is described in sections [II](#_IV.5._Continuously_observational), [III](#_III._Mixture_hidden) and discussed more in section [IV](#_IV._Discussion_and). My reasoning is based on EM algorithm and Lagrangian function for solving optimization problem.

As a convention, all equations are called formulas and they are entitled so that it is easy for researchers to look up them. Tables, figures, and formulas are numbered according to their sections. For example, formula [I.1.1](#_Formula_III.2.1._Forward) is the first formula in sub-section [I.1](#_IV.4.1._HMM_evaluation). Most common notations “*exp*” and “*ln*” denote exponential function and natural logarithm function.

There are many real-world phenomena (so-called states) that we would like to model in order to explain our observations. Often, given sequence of observations symbols, there is demand of discovering real states. For example, there are some states of weather: *sunny*, *cloudy*, *rainy* (Fosler-Lussier, 1998, p. 1). Suppose you are in the room and do not know the weather outside but you are notified observations such as wind speed, atmospheric pressure, humidity, and temperature from someone else. Basing on these observations, it is possible for you to forecast the weather by using HMM. Before discussing about HMM, we should glance over the definition of Markov model (MM). First, MM is the statistical model which is used to model the stochastic process. MM is defined as below (Schmolze, 2001):

* Given a finite set of state *S*={*s*1, *s*2,…, *sn*} whose cardinality is *n*. Let ∏ be the *initial state distribution* where *πi*∏ represents the probability that the stochastic process begins in state *si*. In other words *πi* is the initial probability of state *si*, where .
* The stochastic process which is modeled gets only one state from *S* at all time points. This stochastic process is defined as a finite vector *X*=(*x*1, *x*2,…, *xT*) whose element *xt* is a state at time point *t*. The process *X* is called *state stochastic process* and *xt* *S* equals some state *si* *S*. Note that *X* is also called *state sequence*. Time point can be in terms of second, minute, hour, day, month, year, etc. It is easy to infer that the initial probability *πi* = *P*(*x*1*=si*) where *x*1 is the first state of the stochastic process. The state stochastic process *X* must meet fully the *Markov property*, namely, given previous state *xt*–1 of process *X*, the conditional probability of current state *xt* is only dependent on the previous state *xt*–1, not relevant to any further past state (*xt*–2, *xt*–3,…, *x*1). In other words, *P*(*xt | xt*–1, *xt*–2, *xt*–3,…, *x*1) = *P*(*xt | xt*–1) with note that *P*(.) also denotes probability in this research. Such process is called first-order Markov process.
* At each time point, the process changes to the next state based on the *transition probability distribution* *aij*, which depends only on the previous state. So *aij* is the probability that the stochastic process changes current state *si* to next state *sj*. It means that *aij* = *P*(*xt=sj* | *xt–*1*=si*) = *P*(*xt+*1*=sj* | *xt=si*). The probability of transitioning from any given state to some next state is 1*,* we have. All transition probabilities *aij* (s) constitute the *transition probability matrix* *A*. Note that *A* is *n* by *n* matrix because there are *n* distinct states. It is easy to infer that matrix *A* represents state stochastic process *X*. It is possible to understand that the initial probability matrix ∏ is degradation case of matrix *A*.

Briefly, MM is the triple 〈*S*, *A*,∏〉. In typical MM, states are observed directly by users and transition probabilities (*A* and ∏) are unique parameters. Otherwise, hidden Markov model (HMM) is similar to MM except that the underlying states become hidden from observer, they are hidden parameters. HMM adds more output parameters which are called observations. Each state (hidden parameter) has the conditional probability distribution upon such observations. HMM is responsible for discovering hidden parameters (states) from output parameters (observations), given the stochastic process. The HMM has further properties as below (Schmolze, 2001):

* Suppose there is a finite set of possible observations Φ = {*φ*1, *φ*2,…, *φm*} whose cardinality is *m*. There is the second stochastic process which produces *observations* correlating with hidden states. This process is called *observable stochastic process*, which is defined as a finite vector *O* = (*o*1, *o*2,…, *oT*) whose element *ot* is an observation at time point *t*. Note that *ot* Φ equals some *φk*. The process *O* is often known as *observation sequence*.
* There is a probability distribution of producing a given observation in each state. Let *bi*(*k*) be the probability of observation *φk* when the state stochastic process is in state *si*. It means that *bi*(*k*) = *bi*(*ot=φk*) = *P*(*ot=φk* | *xt=si*). The sum of probabilities of all observations which observed in a certain state is 1, we have. All probabilities of observations *bi*(*k*) constitute the *observation probability matrix* *B*. It is convenient for us to use notation *bik* instead of notation *bi*(*k*). Note that *B* is *n* by *m* matrix because there are *n* distinct states and *m* distinct observations. While matrix *A* represents state stochastic process *X*, matrix *B* represents observable stochastic process *O*.

Thus, HMM is the 5-tuple ∆ = 〈*S*, Φ, *A*, *B*,∏〉. Note that components *S*,Φ, *A*, *B*,and∏ are often called parameters of HMM in which *A*, *B*,and∏ are essential parameters. Going back weather example, suppose you need to predict how weather tomorrow is: *sunny*, *cloudy* or *rainy* since you know only observations about the humidity: *dry*, *dryish*, *damp*, *soggy*. The HMM is totally determined based on its parameters *S*,Φ, *A*, *B*,and∏ according to weather example. We have *S* = {*s*1=*sunny*, *s*2=*cloudy*, *s*3=*rainy*}, Φ = {*φ*1=*dry*, *φ*2=*dryish*, *φ*3=*damp*, *φ*4=*soggy*}. Transition probability matrix *A* is shown in table [I.1](#_Table_III.2.1._Transition).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Weather current day  (Time point *t*) | | |
| *sunny* | *cloudy* | *rainy* |
| Weather previous day  (Time point *t* –1) | *sunny* | *a*11=0.50 | *a*12=0.25 | *a*13=0.25 |
| *cloudy* | *a*21=0.30 | *a*22=0.40 | *a*23=0.30 |
| *rainy* | *a*31=0.25 | *a*32=0.25 | *a*33=0.50 |

###### **Table I.1.** Transition probability matrix *A*

From table [I.1](#_Table_III.2.1._Transition), we have *a*11+*a*12+*a*13=1, *a*21+*a*22+*a*23=1, *a*31+*a*32+*a*33=1.

Initial state distribution specified as uniform distribution is shown in table [I.2](#_Table_III.2.2._Uniform).

|  |  |  |
| --- | --- | --- |
| *sunny* | *cloudy* | *rainy* |
| *π*1=0.33 | *π*2=0.33 | *π*3=0.33 |

###### **Table I.2.** Uniform initial state distribution ∏

From table [I.2](#_Table_III.2.2._Uniform), we have *π*1+*π*2+*π*3=1.

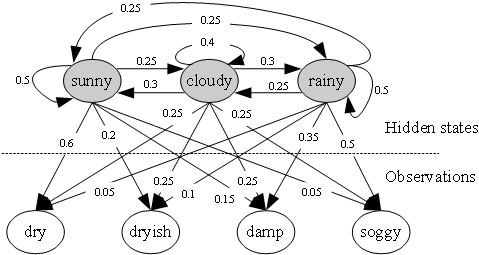
Observation probability matrix *B* is shown in table [I.3](#_Table_III.2.3._Observation).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Humidity | | | |
| *dry* | *dryish* | *damp* | *soggy* |
| Weather | *sunny* | *b*11=0.60 | *b*12=0.20 | *b*13=0.15 | *b*14=0.05 |
| *cloudy* | *b*21=0.25 | *b*22=0.25 | *b*23=0.25 | *b*24=0.25 |
| *rainy* | *b*31=0.05 | *b*32=0.10 | *b*33=0.35 | *b*34=0.50 |

###### **Table I.3.** Observation probability matrix *B*

From table [I.3](#_Table_III.2.3._Observation), we have *b*11+*b*12+*b*13+*b*14=1, *b*21+*b*22+*b*23+*b*24=1, *b*31+*b*32+*b*33+*b*34=1.

The whole weather HMM is depicted in figure [I.1](#_Figure_III.2.4._HMM).



**Figure I.1.** HMM of weather forecast (hidden states are shaded)

There are three problems of HMM (Schmolze, 2001) (Rabiner, 1989, pp. 262-266):

1. Given HMM ∆ and an observation sequence *O* = {*o*1, *o*2,…, *oT*} where *ot* Φ, how to calculate the probability *P*(*O*|*∆*) of this observation sequence. Such probability *P*(*O*|*∆*) indicates how much the HMM ∆ affects on sequence *O*. This is *evaluation problem* or *explanation problem*. Note that it is possible to denote *O* = {*o*1 → *o*2 →…→ *oT*} and the sequence *O* is aforementioned observable stochastic process.
2. Given HMM ∆ and an observation sequence *O* = {*o*1, *o*2,…, *oT*} where *ot* Φ, how to find the sequence of states *X* = {*x*1, *x*2,…, *xT*} where *xt* *S* so that *X* is most likely to have produced the observation sequence *O*. This is *uncovering problem*. Note that the sequence *X* is aforementioned state stochastic process.
3. Given HMM ∆ and an observation sequence *O* = {*o*1, *o*2,…, *oT*} where *ot* Φ, how to adjust parameters of ∆ such as initial state distribution ∏, transition probability matrix *A*, and observation probability matrix *B* so that the quality of HMM ∆ is enhanced. This is *learning problem*.

These problems will be mentioned in sub-sections [I.1](#_IV.4.1._HMM_evaluation), [I.2](#_IV.4.2._HMM_uncovering), and [I.3](#_IV.4.3._HMM_learning), in turn.

### I.1. HMM evaluation problem

The essence of evaluation problem is to find out the way to compute the probability *P*(*O*|*∆*) most effectively given the observation sequence *O* = {*o*1, *o*2,…, *oT*}. For example, given HMM ∆ whose parameters *A*, *B*,and∏ specified in tables [I.1](#_Table_III.2.1._Transition), [I.2](#_Table_III.2.2._Uniform), and [I.3](#_Table_III.2.3._Observation), which is designed for weather forecast. Suppose we need to calculate the probability of event that humidity is *soggy* and *dry* in days 1 and 2, respectively. This is evaluation problem with sequence of observations *O* = {*o*1=*φ*4=*soggy*, *o*2=*φ*1=*dry*, *o*3=*φ*2=*dryish*}. There is a complete set of 33=27 mutually exclusive cases of weather states for three days; for example, given a case in which weather states in days 1, 2, and 3 are *sunny*, *sunny*, and *sunny* then, state stochastic process is *X* = {*x*1=*s*1=*sunny*, *x*2=*s*1=*sunny*, *x*3=*s*1=*sunny*}. It is easy to recognize that it is impossible to browse all combinational cases of given observation sequence *O* = {*o*1, *o*2,…, *oT*} as we knew that it is necessary to survey 33=27 mutually exclusive cases of weather states with a tiny number of observations {*soggy*, *dry*, *dryish*}. Exactly, given *n* states and *T* observations, it takes extremely expensive cost to survey *nT* cases. According to (Rabiner, 1989, pp. 262-263), there is a so-called *forward-backward procedure* to decrease computational cost for determining the probability *P*(*O|*Δ). Let *αt*(*i*) be the joint probability of partial observation sequence {*o*1, *o*2,…, *ot*} and state *xt*=*si* where , specified by formula [I.1.1](#_Formula_III.2.1._Forward).

**Formula I.1.1.** Forward variable

The joint probability *αt*(*i*) is also called *forward variable* at time point *t* and state *si*. Formula [I.1.2](#_Formula_III.2.2._Recurrence) specifies recurrence property of forward variable (Rabiner, 1989, p. 262).

**Formula I.1.2.** Recurrence property of forward variable

Where *bj*(*ot+*1) is the probability of observation *ot+*1 when the state stochastic process is in state *sj*, please see an example of observation probability matrix shown in table [I.3](#_Table_III.2.3._Observation). Please pay attention to recurrence property of forward variable specified by formula [I.1.2](#_Formula_III.2.2._Recurrence) because this formula is essentially to build up Markov chain.

According to the forward recurrence formula [I.1.2](#_Formula_III.2.2._Recurrence), given observation sequence *O* = {*o*1, *o*2,…, *oT*}, we have:

The probability *P*(*O|*Δ) is sum of *αT*(*i*) over all *n* possible states of *xT*, specified by formula [I.1.3](#_Formula_III.2.3._Probability).

**Formula I.1.3.** Probability *P*(*O|*Δ) based on forward variable

The forward-backward procedure to calculate the probability *P*(*O|*Δ), based on forward formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.3](#_Formula_III.2.3._Probability), includes three steps as shown in table [I.1.1](#_Table_III.2.4._Forward-backward) (Rabiner, 1989, p. 262).

|  |
| --- |
| 1. Initialization step: Initializing *α*1(*i*) = *bi*(*o*1)*πi* for all 2. Recurrence step: Calculating all *αt+*1(*j*) for all and according to formula [I.1.2](#_Formula_III.2.2._Recurrence). 3. Evaluation step: Calculating the probability |

###### **Table I.1.1.** Forward-backward procedure based on forward variable to calculate the probability *P*(*O|*Δ)

There is interesting thing that the forward-backward procedure can be implemented based on so-called *backward variable*. Let *βt*(*i*) be the backward variable which is conditional probability of partial observation sequence {*ot*, *ot+*1,…, *oT*} given state *xt*=*si* where , specified by formula [I.1.4](#_Formula_III.2.4._Backward).

**Formula I.1.4.** Backward variable

The recurrence property of backward variable specified by formula [I.1.5](#_Formula_III.2.5._Recurrence) (Rabiner, 1989, p. 263).

**Formula I.1.5.** Recurrence property of backward variable

Where *bj*(*ot+*1) is the probability of observation *ot+*1 when the state stochastic process is in state *sj*, please see an example of observation probability matrix shown in table [I.3](#_Table_III.2.3._Observation). The construction of backward recurrence formula [I.1.5](#_Formula_III.2.5._Recurrence) is essentially to build up Markov chain.

The probability *P*(*O|*Δ) is sum of product *πibi*(*o*1)*β*1(*i*) over all *n* possible states of *x*1=*si*, specified by formula [I.1.6](#_Formula_III.2.6._Probability).

**Formula I.1.6.** Probability *P*(*O|*Δ) based on backward variable

The forward-backward procedure to calculate the probability *P*(*O|*Δ), based on backward formulas [I.1.5](#_Formula_III.2.5._Recurrence) and [I.1.6](#_Formula_III.2.6._Probability), includes three steps as shown in table [I.1.2](#_Table_III.2.5._Forward-backward) (Rabiner, 1989, p. 263).

|  |
| --- |
| 1. Initialization step: Initializing *βT*(*i*) = 1for all 2. Recurrence step: Calculating all *βt*(*i*) for all and *t=T*–1, *t=T*–2,…, *t=*1, according to formula [I.1.5](#_Formula_III.2.5._Recurrence). 3. Evaluation step: Calculating the probability *P*(*O|*Δ) according to formula [I.1.6](#_Formula_III.2.6._Probability), |

###### **Table I.1.2.** Forward-backward procedure based on backward variable to calculate the probability *P*(*O|*Δ)

Now the uncovering problem is mentioned particularly in successive sub-section [I.2](#_IV.4.2._HMM_uncovering).

### I.2. HMM uncovering problem

Recall that given HMM ∆ and observation sequence *O* = {*o*1, *o*2,…, *oT*} where *ot* Φ, how to find out a state sequence *X* = {*x*1, *x*2,…, *xT*} where *xt S* so that *X* is most likely to have produced the observation sequence *O*. This is the uncovering problem: which sequence of state transitions is most likely to have led to given observation sequence. In other words, it is required to establish an *optimal criterion* so that the state sequence *X* leads to maximizing such criterion. The simple criterion is the conditional probability of sequence *X* with respect to sequence *O* and model ∆, denoted *P*(*X*|*O*,∆). We can apply brute-force strategy: “go through all possible such *X* and pick the one leading to maximizing the criterion *P*(*X|O*,∆)”.

This strategy is impossible if the number of states and observations is huge. Another popular way is to establish a so-called *individually optimal criterion* (Rabiner, 1989, p. 263) which is described right later.

Let *γt*(*i*) be joint probability that the stochastic process is in state *si* at time point *t* with observation sequence *O* = {*o*1, *o*2,…, *oT*}, formula [I.2.1](#_Formula_IV.4.2.1._Probability) specifies this probability based on forward variable *αt* and backward variable *βt*.

**Formula I.2.1.** Joint probability of being in state *si* at time point *t* with observation sequence *O*

The variable *γt*(*i*) is also called *individually optimal criterion* with note that forward variable *αt* and backward variable *βt* are calculated according to recurrence formulas [I.1.2](#_Formula_IV.4.1.2._Recurrence) and [I.1.5](#_Formula_IV.4.1.5._Recurrence), respectively.

Because the probability is not relevant to state sequence *X*, it is possible to remove it from the optimization criterion. Thus, formula [I.2.2](#_Formula_III.2.7._Optimal) specifies how to find out the optimal state *xt* of *X* at time point *t*.

**Formula I.2.2.** Optimal state at time point *t*

Note that index *i* is identified with state according to formula [I.2.2](#_Formula_III.2.7._Optimal). The optimal state *xt* of *X* at time point *t* is the one that maximizes product *αt*(*i*) *βt*(*i*) over all values *si*. The procedure to find out state sequence *X* = {*x*1, *x*2,…, *xT*} based on individually optimal criterion is called *individually optimal procedure* that includes three steps, shown in table [I.2.1](#_Table_III.2.6._Individually).

|  |
| --- |
| 1. Initialization step:  * Initializing *α*1(*i*) = *bi*(*o*1)*πi* for all * Initializing *βT*(*i*) = 1for all  1. Recurrence step:  * Calculating all *αt+*1(*i*) for all and according to formula [I.1.2](#_Formula_IV.4.1.2._Recurrence). * Calculating all *βt*(*i*) for all and *t=T*–1, *t=T*–2,…, *t=*1, according to formula [I.1.5](#_Formula_IV.4.1.5._Recurrence). * Calculating all *γt*(*i*)=*αt*(*i*)*βt*(*i*) for all and according to formula [I.2.1](#_Formula_IV.4.2.1._Probability). * Determining optimal state *xt* of *X* at time point *t* is the one that maximizes *γt*(*i*) over all values *si*.  1. Final step: The state sequence *X* = {*x*1, *x*2,…, *xT*} is totally determined when its partial states *xt* (s) where are found in recurrence step. |

###### **Table I.2.1.** Individually optimal procedure to solve uncovering problem

The individually optimal criterion *γt*(*i*) does not reflect the whole probability of state sequence *X* given observation sequence *O* because it focuses only on how to find out each partially optimal state *xt* at each time point *t*. Thus, the individually optimal procedure is heuristic method. Viterbi algorithm (Rabiner, 1989, p. 264) is alternative method that takes interest in the whole state sequence *X* by using joint probability *P*(*X*,*O|*Δ) of state sequence and observation sequence as optimal criterion for determining state sequence *X*. Let *δt*(*i*) be the maximum joint probability of observation sequence *O* and state *xt*=*si* over *t*–1 previous states. The quantity *δt*(*i*) is called *joint optimal criterion* at time point *t*, which is specified by formula  [I.2.3](#_Formula_III.2.8._Joint).

**Formula I.2.3.** Joint optimal criterion at time point *t*

The recurrence property of *joint optimal criterion* is specified by formula  [I.2.4](#_Formula_III.2.9._Recurrence) (Rabiner, 1989, p. 264).

**Formula I.2.4.** Recurrence property of joint optimal criterion

The semantic content of joint optimal criterion *δt* is similar to the forward variable *αt*. Given criterion *δt+*1(*j*), the state *xt+*1=*sj* that maximizes *δt+*1(*j*) is stored in the backtracking state *qt+*1(*j*) that is specified by formula [I.2.5](#_Formula_III.2.10._Backtracking).

**Formula I.2.5.** Backtracking state

Note that index *i* is identified with state according to formula [I.2.5](#_Formula_III.2.10._Backtracking). The Viterbi algorithm based on joint optimal criterion *δt*(*i*) includes three steps described in table [I.2.2](#_Table_III.2.7._Viterbi) (Rabiner, 1989, p. 264).

|  |
| --- |
| 1. Initialization step:  * Initializing *δ*1(*i*) = *bi*(*o*1)*πi* for all * Initializing *q*1(*i*) = 0 for all  1. Recurrence step:  * Calculating all for all and according to formula [I.2.4](#_Formula_III.2.9._Recurrence). * Keeping tracking optimal states for all and according to formula [I.2.5](#_Formula_III.2.10._Backtracking).  1. State sequence backtracking step: The resulted state sequence *X* = {*x*1, *x*2,…, *xT*} is determined as follows:  * The last state * Previous states are determined by backtracking: *xt* = *qt+*1(*xt+*1) for *t=T*–1, *t=T*–2,…, *t=*1. |

###### **Table I.2.2.** Viterbi algorithm to solve uncovering problem

Now the uncovering problem is described thoroughly in this sub-section [I.2](#_IV.4.2._HMM_uncovering). Successive sub-section [I.3](#_IV.4.3._HMM_learning) will mention the last problem of HMM that is the learning problem.

### I.3. HMM learning problem

The learning problem is to adjust parameters such as initial state distribution ∏, transition probability matrix *A*, and observation probability matrix *B* so that given HMM ∆ gets more appropriate to an observation sequence *O* = {*o*1, *o*2,…, *oT*} with note that ∆ is represented by these parameters. In other words, the learning problem is to adjust parameters by maximizing probability of observation sequence *O*, as follows:

The Expectation Maximization (EM) algorithm is applied successfully into solving HMM learning problem, which is equivalently well-known Baum-Welch algorithm by authors Leonard E. Baum and Lloyd R. Welch (Rabiner, 1989). The successive sub-section [I.3.1](#_IV.4.3.1._EM_algorithm) describes shortly EM algorithm before going into Baum-Welch algorithm.

#### I.3.1. EM algorithm

Expectation Maximization (EM) is effective parameter estimator in case that incomplete data is composed of two parts: observed part and hidden part (missing part). EM is iterative algorithm that improves parameters after iterations until reaching optimal parameters. Each iteration includes two steps: E(xpectation) step and M(aximization) step. In E-step the hidden data is estimated based on observed data and current estimate of parameters; so the lower-bound of likelihood function is computed by the expectation of complete data. In M-step new estimates of parameters are determined by maximizing the lower-bound. Please see document (Sean, 2009) for short tutorial of EM. This sub-section [I.3.1](#_IV.4.3.1._EM_algorithm) focuses on practice general EM algorithm; the theory of EM algorithm is described comprehensively in article “Maximum Likelihood from Incomplete Data via the EM algorithm” by authors (Dempster, Laird, & Rubin, 1977).

Suppose *O* and *X* are observed data and hidden data, respectively. Note *O* and *X* can be represented in any form such as discrete values, scalar, integer number, real number, vector, list, sequence, sample, and matrix. Let represent parameters of probability distribution. Concretely, includes initial state distribution ∏, transition probability matrix *A*, and observation probability matrix *B* inside HMM*.* In other words, represents HMM Δ itself. EM algorithm aims to estimate by finding out which maximizes the likelihood function = .

Where is the optimal estimate of parameters which is called usually *parameter estimate*. Note that notation “*ln*” denotes natural logarithm function.

The expression is essentially expectation of given conditional probability distribution when is totally determined. Let denote this conditional expectation, formula [I.3.1.1](#_Formula_III.2.14._EM) specifies EM optimization criterion for determining the parameter estimate, which is the most important aspect of EM algorithm (Sean, 2009, p. 8).

Where,

**Formula I.3.1.1.** EM optimization criterion based on conditional expectation

If is continuous density function, the continuous version of this conditional expectation is:

Finally, the EM algorithm is described in table [I.3.1.1](#_Table_III.2.9._General).

|  |
| --- |
| Starting with initial parameter , each iteration in EM algorithm has two steps:   1. *E-step*: computing the conditional expectation based on the current parameter according to formula [I.3.1.1](#_Formula_III.2.14._EM). 2. *M-step*: finding out the estimate that maximizes such conditional expectation. The next parameter is assigned by the estimate , we have:   Of course becomes current parameter for next iteration. How to maximize the conditional expectation is optimization problem which is dependent on applications. For example, the popular method to solve optimization problem is Lagrangian duality (Jia, 2013, p. 8).  EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter and next parameter is smaller than some pre-defined threshold *ε*.  In addition, it is possible to define a custom terminating condition. |

###### **Table I.3.1.1.** General EM algorithm

In general, it is easy to calculate the EM expectation but finding out the estimate based on maximizing such expectation is complicated optimization problem. It is possible to state that the essence of EM algorithm is to determine the estimate . Now the EM algorithm is introduced to you. How to apply it into solving HMM learning problem is described in successive sub-section [I.3.2](#_IV.4.3.2._Applying_EM).

#### I.3.2. Applying EM algorithm into solving learning problem

Now going back the HMM learning problem, the EM algorithm is applied into solving this problem, which is equivalently well-known Baum-Welch algorithm by authors Leonard E. Baum and Lloyd R. Welch (Rabiner, 1989). The parameter becomes the HMM model Δ = (*A*, *B*, ∏). Recall that the learning problem is to adjust parameters by maximizing probability of observation sequence *O*, as follows:

*Where* , , *are parameter estimates and so, the purpose of HMM learning problem is to determine them*.

The observation sequence *O* = {*o*1, *o*2,…, *oT*} and state sequence *X* = {*x*1, *x*2,…, *xT*} are observed data and hidden data within context of EM algorithm, respectively. Note *O* and *X* is now represented in sequence. According to EM algorithm, the parameter estimate is determined as follows:

Where Δ*r* = (*Ar*, *Br*, ∏*r*) is the known parameter at the current iteration. Note that we use notation Δ*r* instead of popular notation Δ*t* in order to distinguish iteration indices of EM algorithm from time points inside observation sequence *O* and state sequence *X*.

It is conventional that where *x*0 is pseudo-state, formula [I.3.2.1](#_Formula_IV.4.3.2.1._General) specifies general EM conditional expectation for HMM:

**Formula I.3.2.1.** General EM conditional expectation for HMM

Note that notation “*ln*” denotes natural logarithm function.

Because of the convention , matrix ∏ is degradation case of matrix *A* at time point *t=*1. In other words, the initial probability *πj* is equal to the transition probability *aij* from pseudo-state *x*0 to state *x*1=*sj*.

Note that *n*=|*S*| is the number of possible states and *m*=|Φ| is the number of possible observations. Let and are two index functions so that

The EM conditional expectation for HMM is specified by formula [I.3.2.2](#_Formula_III.2.16._EM).

**Formula I.3.2.2.** EM conditional expectation for HMM

Where,

Note that the conditional expectation is function of Δ. There are two constraints for HMM as follows:

Maximizing with subject to these constraints is optimization problem that is solved by Lagrangian duality theorem (Jia, 2013, p. 8). Original optimization problem mentions minimizing target function but it is easy to infer that maximizing target function shares the same methodology. Let *l*(Δ, *λ*, *μ*) be Lagrangian function constructed from together with these constraints (Ramage, 2007, p. 9), we have formula [I.3.2.3](#_Formula_IV.4.3.4._Lagrangian) for specifying HMM Lagrangian function as follows:

**Formula I.3.2.3.** Lagrangian function for HMM

Where *λ* is *n*-component vector *λ* = (*λ*1, *λ*2,…, *λn*) and *μ* is *m*-component vector *μ* = (*μ*1, *μ*2,…, *μm*). Factors *λi* ≥ 0 and *μj* ≥ 0 are called Lagrange multipliers or Karush-Kuhn-Tucker multipliers (Wikipedia, Karush–Kuhn–Tucker conditions, 2014) or dual variables. The expectation is specified by formula [I.3.2.2](#_Formula_III.2.16._EM).

The parameter estimate is extreme point of the Lagrangian function. According to Lagrangian duality theorem (Boyd & Vandenberghe, 2009, p. 216) (Jia, 2013, p. 8), we have:

The parameter estimate is determined by setting partial derivatives of *l*(Δ, *λ*, *μ*) with respect to *aij* and *bj*(*k*) to be zero.

By solving these equations, we have formula [I.3.2.4](#_Formula_IV.4.3.5._HMM) for specifying HMM parameter estimate given current parameter Δ = (*aij*, *bj*(*k*), *πj*) as follows:

**Formula I.3.2.4.** HMM parameter estimate

The parameter estimate is the ultimate solution of the learning problem. As seen in formula [I.3.2.4](#_Formula_IV.4.3.5._HMM), it is necessary to calculate probabilities *P*(*O*, *xt–*1=*si*, *xt*=*sj*) and *P*(*O*, *xt–*1*=si*) when other probabilities *P*(*O*, *xt*=*sj*), *P*(*O*, *x*1=*si*), and *P*(*O*, *x*1=*sj*) are represented by the joint probability *γt* specified by formula [I.2.1](#_Formula_IV.4.2.1._Probability).

Let *ξt*(*i*, *j*) is the joint probability that the stochastic process receives state *si* at time point *t*–1 and state *sj* at time point *t* given observation sequence *O* (Rabiner, 1989, p. 264).

Formula [I.3.2.5](#_Formula_IV.4.3.6._Joint) determines the joint probability *ξt*(*i*, *j*) based on forward variable *αt* and backward variable *βt*.

**Formula I.3.2.5.** Joint probability *ξt*(*i*, *j*)

Where forward variable *αt* and backward variable *βt* are calculated by previous recurrence formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.5](#_Formula_III.2.5._Recurrence).

Recall that *γt*(*j*) is the joint probability that the stochastic process is in state *sj* at time point *t* with observation sequence *O* = {*o*1, *o*2,…, *oT*}, specified by previous formula [I.2.1](#_Formula_IV.4.2.1._Probability).

According to total probability rule, it is easy to infer that *γt* is sum of *ξt* over all states with , as seen in following formula [I.3.2.6](#_Formula_IV.4.3.2.5._Relationship).

**Formula I.3.2.6.** The *γt* is sum of *ξt* over all states

Deriving from formulas [I.3.2.5](#_Formula_IV.4.3.6._Joint) and [I.3.2.6](#_Formula_IV.4.3.2.5._Relationship), we have:

By extending formula [I.3.2.4](#_Formula_IV.4.3.5._HMM), we receive formula [I.3.2.7](#_Formula_IV.4.3.7._HMM) for specifying HMM parameter estimate given current parameter Δ = (*aij*, *bi*(*k*), *πi*) in detailed.

**Formula I.3.2.7.** HMM parameter estimate in detailed

The formula [I.3.2.7](#_Formula_IV.4.3.7._HMM) and its proof are found in (Ramage, 2007, pp. 9-12). It is easy to infer that the parameter estimate is based on joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) which, in turn, are based on current parameter Δ = (*aij*, *bj*(*k*), *πj*). The EM conditional expectation is determined by joint probabilities *ξt*(*i*, *j*) and *γt*(*j*); so, the main task of E-step in EM algorithm is essentially to calculate the joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) according to formulas [I.3.2.5](#_Formula_IV.4.3.6._Joint) and [I.2.1](#_Formula_IV.4.2.1._Probability). The EM conditional expectation gets maximal at estimate and so, the main task of M-step in EM algorithm is essentially to calculate according to formula [I.3.2.7](#_Formula_IV.4.3.7._HMM). The EM algorithm is interpreted in HMM learning problem, as shown in table [I.3.2.1](#_Table_IV.4.3.2._EM).

|  |
| --- |
| Starting with initial value for Δ, each iteration in EM algorithm has two steps:   1. *E-step*: Calculating the joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) according to formulas [I.3.2.5](#_Formula_IV.4.3.6._Joint) and [I.2.1](#_Formula_IV.4.2.1._Probability) given current parameter Δ = (*aij*, *bj*(*k*), *πj*).   Where forward variable *αt* and backward variable *βt* are calculated by previous recurrence formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.5](#_Formula_III.2.5._Recurrence).   1. *M-step*: Calculating the estimate based on the joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) determined at E-step, according to formula [I.3.2.7](#_Formula_IV.4.3.7._HMM).   The estimate becomes the current parameter for next iteration.  EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter Δ and next parameter is insignificant. It is possible to define a custom terminating condition. |

###### **Table I.3.2.1.** EM algorithm for HMM learning problem

The algorithm to solve HMM learning problem shown in table [I.3.2.1](#_Table_IV.4.3.2._EM) is known as Baum-Welch algorithm by authors Leonard E. Baum and Lloyd R. Welch (Rabiner, 1989). Please see document “Hidden Markov Models Fundamentals” by (Ramage, 2007, pp. 8-13) for more details about HMM learning problem. As aforementioned in previous sub-section [I.3.1](#_IV.4.3.1._EM_algorithm), the essence of EM algorithm applied into HMM learning problem is to determine the estimate .

As seen in table [I.3.2.1](#_Table_IV.4.3.2._EM), it is not difficult to run E-step and M-step of EM algorithm but how to determine the terminating condition is considerable problem. It is better to establish a computational terminating criterion instead of applying the general statement “EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter Δ and next parameter is insignificant”. Therefore, author (Nguyen L. , Tutorial on Hidden Markov Model, 2016) proposes the probability *P*(*O|*Δ) as the terminating criterion. Calculating criterion *P*(*O|*Δ) is evaluation problem described in sub-section [I.1](#_IV.4.1._HMM_evaluation). Criterion *P*(*O|*Δ) is determined according to forward-backward procedure; please see tables [I.1.1](#_Table_III.2.4._Forward-backward) and [I.1.2](#_Table_IV.4.1.2._Forward-backward) for more details about forward-backward procedure.

|  |
| --- |
| 1. Initialization step: Initializing *α*1(*i*) = *bi*(*o*1)*πi* for all 2. Recurrence step: Calculating all *αt+*1(*j*) for all and according to formula [I.1.2](#_Formula_III.2.2._Recurrence). 3. Evaluation step: Calculating the probability |

Concretely, when EM algorithm results out forward variables in E-step, the forward-backward procedure takes advantages of such forward variables so as to determine criterion *P*(*O|*Δ) the at the same time. As a result, the speed of EM algorithm does not decrease. However, there is always a redundant iteration; suppose that the terminating criterion approaches to maximal value at the end of the *rth* iteration but the EM algorithm only stops at the E-step of the (*r*+1)*th* iteration when it really evaluates the terminating criterion. In general, the terminating criterion *P*(*O|*Δ) is calculated based on the current parameter Δ at E-step instead of the estimate at M-step. Table [I.3.2.2](#_Table_IV.4.3.2.2._Proposed) (Nguyen, Tutorial on Hidden Markov Model, 2016) shows the proposed implementation of EM algorithm with terminating criterion *P*(*O|*Δ). Pseudo-code like programming language C is used to describe the implementation of EM algorithm. Note, variables are marked as *italic words*, programming language keywords (*while*, *for*, *if*, [], *==*, !=, &&, //, etc.) are marked blue and comments are marked gray. For example, notation [] denotes array index operation; concretely, *α*[*t*][*i*] denotes forward variable *αt*(*i*) at time point *t* with regard to state *si*.

|  |
| --- |
| Input:  HMM with current parameter Δ = {*aij*, *πj*, *bjk*}  Observation sequence *O* = {*o*1, *o*2,…, *oT*}  Output:  HMM with optimized parameter Δ = {*aij*, *πj*, *bjk*}  Allocating memory for two matrices *α* and *β* representing forward variables and backward variables.  *previous\_criterion* = –1  *current\_criterion* = –1  *iteration* = 0  //Pre-defined number *MAX\_ITERATION* is used to prevent from infinite loop.  *MAX\_ITERATION* = 10000  While (*iteration* < *MAX\_ITERATION*)    //Calculating forward variables and backward variables  For *t* = 1 to *T*  For *i* = 1 to *n*  Calculating forward variables *α*[*t*][*i*] and backward variables *β*[*T*–*t*+1][*i*] based on observation sequence *O* according to formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.5](#_Formula_III.2.5._Recurrence).  End for *i*  End for *t*    //Calculating terminating criterion *current\_criterion* = *P*(*O|*Δ)  *current\_criterion* = 0  For *i* = 1 to *n*  *current\_criterion* = *current\_criterion* + *α*[*T*][*i*]  End for i  //Terminating condition  If *previous\_criterion* >= 0 && *previous\_criterion* == *current\_criterion* then  break //breaking out the loop, the algorithm stops  Else  *previous\_criterion* = *current\_criterion*  End if    //Updating transition probability matrix  For *i* = 1 to *n*  *denominator* = 0  Allocating *numerators* as a 1-dimension array including *n* zero elements.  For *t* = 2 to *T*  For *k* = 1 to *n*  *ξ* = *α*[*t*–1][*i*] \* *aik* \* *bk*(*ot*) \* *β*[*t*][*k*]  *numerators*[*k*] = *numerators*[*k*] + *ξ*  *denominator* = *denominator* + *ξ*  End for *k*  End for *t*    If *denominator* != 0 then  For *j* = 1 to *n*  *aij* = *numerators*[*j*] / *denominator*  End for *j*  End if    End for *i*    //Updating initial probability matrix  Allocating *g* as a 1-dimension array including *n* elements.  *sum* = 0  For *j* = 1 to *n*  *g*[*j*] = *α*[1][*j*] \* *β*[1][*j*]  *sum* = *sum* + *g*[j]  End for *j*    If *sum* != 0 then  For *j* = 1 to *n*  *πj* = *g*[*j*] / *sum*  End for *j*  End if  //Updating observation probability distribution  For *j* = 1 to *n*  Allocating *γ* as a 1-dimension array including *T* elements.  *denominator* = 0  For *t* = 1 to *T*  *γ*[*t*] = *α*[*t*][*j*] \* *β*[*t*][*j*]  *denominator* = *denominator* + *γ*[*t*]  End for *t*    Let *m* be the columns of observation distribution matrix *B*.  For *k* = 1 to *m*  *numerator* = 0  For *t* = 1 to *T*  If *ot* == *k* then  *numerator* = *numerator* + *γ*[*t*]  End if  End for *t*    *bjk* = *numerator* / *denominator*  End for *k*  End for *j*  *iteration* = *iteration* + 1  End while |

###### **Table I.3.2.2.** Proposed implementation of EM algorithm for learning HMM with terminating criterion *P*(*O|*Δ)

According to table [I.3.2.2](#_Table_IV.4.3.2.2._Proposed), the number of iterations is limited by a pre-defined maximum number, which aims to solve a so-called infinite loop optimization. Although it is proved that EM algorithm always converges, maybe there are two different estimates and at the final convergence. This situation causes EM algorithm to alternate between and in infinite loop. Therefore, the final estimate or is totally determined but the EM algorithm does not stop. This is the reason that the number of iterations is limited by a pre-defined maximum number.

Now three main problems of HMM are described; please see an excellent document “A tutorial on hidden Markov models and selected applications in speech recognition” written by author (Rabiner, 1989) for advanced details about HMM. The next section [II](#_IV.5._Continuously_observational) described a HMM whose observations are continuous.

## II. Continuous observation hidden Markov model

Observations of normal HMM mentioned in previous sub-section [I](#_III.2.4._Hidden_Markov) are quantified by discrete probability distribution that is concretely observation probability matrix *B*. In the general situation, observation *ot* is continuous variable and matrix *B* is replaced by probability density function (PDF). Formula [II.1](#_Formula_IV.5.1._Probability_1) specifies the PDF of continuous observation *ot* given state *sj*.

**Formula II.1.** Probability density function (PDF) of observation

Where the PDF *pj*(*ot|θj*) belongs to any probability distribution, for example, normal distribution, exponential distribution, etc. The notation *θj* denotes probabilistic parameters, for instance, if *pj*(*ot|θj*) is normal distribution PDF, *θj* includes mean *mj* and variance *σj*2. The HMM now is specified by parameter Δ = (*aij*, *θj*, *πj*), which is called *continuous observation HMM* (Rabiner, 1989, p. 267). The PDF *pj*(*ot|θj*) is known as *single PDF* because it is atom PDF which is not combined with any other PDF. We will research so-called mixture model PDF that is constituted of many partial PDF (s) later. We still apply EM algorithm known as Baum-Welch algorithm into learning continuous observation HMM. In the field of continuous-speech recognition, authors (Lee, Rabiner, Pieraccini, & Wilpon, 1990) proposed Bayesian adaptive learning for estimating mean and variance of continuous density HMM. Authors (Huo & Lee, 1997) proposed a framework of quasi-Bayes (QB) algorithm based on approximate recursive Bayes estimate for learning HMM parameters with Gaussian mixture model; they described that “The QB algorithm is designed to incrementally update the hyper-parameters of the approximate posterior distribution and the continuous density HMM parameters simultaneously” (Huo & Lee, 1997, p. 161). Authors (Sha & Saul, 2009) and (Cheng, Sha, & Saul, 2009) used the approach of large margin training to learn HMM parameters. Such approach is different from Baum-Welch algorithm when it firstly establishes discriminant functions for correct and incorrect label sequences and then, finds parameters satisfying the margin constraint that separates the discriminant functions as much as possible (Sha & Saul, 2009, pp. 106-108). Authors (Cheng, Sha, & Saul, 2009, p. 4) proposed a fast online algorithm for large margin training, in which “the parameters for discriminant functions are updated according to an online learning rule with given learning rate”. Large margin training is very appropriate to speech recognition, which was proposed by authors (Sha & Saul, 2006) in the article “Large Margin Hidden Markov Models for Automatic Speech Recognition”. Some other authors used different learning approaches such as conditional maximum likelihood and minimizing classification error, mentioned in (Sha & Saul, 2009, pp. 104-105).

Methods to solve evaluation problem and uncovering problem mentioned previous sub-sections [I.1](#_IV.4.1._HMM_evaluation), [I.2](#_IV.4.2._HMM_uncovering), and [I.3](#_IV.4.3._HMM_learning) are kept intact by using the observation PDF specified by formula [II.1](#_Formula_IV.5.1._Probability_1). For example, forward-backward procedure (based on forward variable, shown in table [I.1.1](#_Table_IV.4.1.1._Forward-backward)) that solves evaluation problem is based on the recurrence formula [I.1.2](#_Formula_III.2.2._Recurrence) as follows:

In order to apply forward-backward procedure into continuous observation HMM, it is simple to replace the discrete probability *bj*(*ot+*1) by the single PDF specified by formula [II.1](#_Formula_IV.5.1._Probability).

However, there is a change in solution of learning problem. Recall that the essence of EM algorithm applied into HMM learning problem is to determine the estimate . Formulas for calculating estimates and are kept intact, as aforementioned in formula [I.3.2.7](#_Formula_IV.4.3.7._HMM).

Where joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) are modified based on replacing discrete probability *bj*(*ot*) by the single PDF *pj*(*ot|θj*) given current parameter Δ = (*aij*, *bj*(*k*), *πj*).

Where forward variable *αt* and backward variable *βt* are calculated by recurrence formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.5](#_Formula_III.2.5._Recurrence). Recall that *γt*(*j*) is joint probability that the stochastic process is in state *sj* at time point *t* with observation sequence *O* and *ξt*(*i*, *j*) is the joint probability that the stochastic process receives state *si* at time point *t–*1 and state *sj* at time point *t* given observation sequence *O*.

Your attention please, quantities *ξt*(*i*, *j*), *γt*(*j*), *αt*(*i*), and *βt*(*j*) are essentially continuous functions because they are based on PDF *pj*(*ot|θj*). Their values on a concrete observation *ot* are zero because the value of PDF *pj*(*ot|θj*) given such concrete observation *ot* is zero. Therefore, in practice, these quantities are calculated according to integral of PDF *pj*(*ot|θj*) in *ε*-vicinity of *ot* where *ε* is very small positive number. The number *ε* can reflect inherent attribute of observation data with regard to measure bias, for example, if atmosphere humidity at time point *t* is , the measure bias is 0.01 and so we have *ε=*0.01. In addition, the number *ε* can be pre-defined fixedly by arbitrary very small number. For example, given *ε=*0.01 we have:

If all *ot* are intervals, for example, then, the integral of PDF *pj*(*ot|θj*) is calculated directly over such *ot*.

Given the PDF *pj*(*ot|θj*) conforms normal distribution, it is easy to calculate the probability of *ot* in *ε*-vicinity as the integral of PDF *pj*(*ot|θj*) in *ε*-vicinity of *ot* as follows:

The best way is to standardize the normal PDF where into cumulative standard normal distribution (Montgomery & Runger, 2003, p. 653). Let Φ be cumulative standard normal distribution (Montgomery & Runger, 2003, p. 653), we have:

The quantities and are standardized values of *b* and *a* given PDF , respectively. The function Φ is always evaluated in popular. For instance, appendix A of the book “Applied Statistics and Probability for Engineers” by authors (Montgomery & Runger, 2003, p. 653) is a good reference for looking up some values of Φ. Please distinguish the function Φ from the set of possible discrete observations Φ = {*φ*1, *φ*2,…, *φm*} aforementioned at the beginning of section [I](#_III.2.4._Hidden_Markov) when they share the same notation.

Formula [II.2](#_Formula_IV.5.2._Joint) specifies quantities *ξt*(*i*, *j*), *γt*(*j*) according to integral of PDF *pj*(*ot|θj*).

Where forward variable *αt* and backward variable *βt* are calculated based on recurrence formulas [I.1.2](#_Formula_III.2.2._Recurrence) and [I.1.5](#_Formula_III.2.5._Recurrence).

**Formula II.2.** Joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) based on single PDF

Note that *mj* and *σj*2 are mean and variance of the normal PDF *pj*(*ot|θj*) and Φ is cumulative standard normal distribution (Montgomery & Runger, 2003, p. 653).

As a convention, quantities *ξt*(*i*, *j*), *γt*(*j*), *αt+*1(*j*), and *βt*(*i*) are still referred as joint probabilities, forward variable,and backward variable. This convention help us to describe traditional HMM and continuous convention HMM in coherent way.

Now it is necessary to determine the estimate. Derived from formula [I.3.2.2](#_Formula_III.2.16._EM), the expectation is modified based on replacing discrete probability *bj*(*ot*) by the continuous PDF *pj*(*ot|θj*), as seen in following formula [II.3](#_Formula_IV.5.2._EM) given current parameter Δ*r*.

**Formula II.3.** EM conditional expectation for continuous observation HMM with single PDF

Where and are index functions so that

Note that notation “*ln*” denotes natural logarithm function. Derived from formula [I.3.2.3](#_Formula_IV.4.3.2.3._Lagrangian), the HMM Lagrangian function for continuous observation HMM with single PDF is specified by formula [II.4](#_Formula_IV.5.2._Lagrangian).

**Formula II.4.** Lagrangian function for continuous observation HMM with single PDF

Where *λ* is *n*-component vector *λ* = (*λ*1, *λ*2,…, *λn*). Factors *λi* ≥ 0 are called Lagrange multipliers or Karush-Kuhn-Tucker multipliers (Wikipedia, Karush–Kuhn–Tucker conditions, 2014) or dual variables.

The parameter estimate which is extreme point of the Lagrangian function *l*(Δ, *λ*) is determined by setting partial derivatives of *l*(Δ, *λ*) with respect to *θj* to be zero. The partial derivative of *l*(Δ, *λ*) with respect to *θj* is:

(Due to total probability rule)

(Due to multiplication rule)

Setting the partial derivative to be zero, we get the equation whose solution is estimate , specified by formula [II.5](#_Formula_IV.5.5._Equation).

**Formula II.5.** Equation of single PDF parameter

Note that notation “*ln*” denotes natural logarithm function.

It is possible to solve the above equation (formula [II.5](#_Formula_IV.5.5._Equation)) by Newton-Raphson method (Burden & Faires, 2011, pp. 67-69) – a numeric analysis method but it is easier and simpler to find out more precise solution if the PDF *pj*(*ot|θj*) belongs to well-known distributions: normal distribution (Montgomery & Runger, 2003, pp. 109-110), exponential distribution (Montgomery & Runger, 2003, pp. 122-123), etc. According to author (Couvreur, 1996, p. 32), the estimate is determined by the more general formula as follows:

The easy way to find out is to solve formula [II.5](#_Formula_IV.5.5._Equation) by taking advantages of derivatives.

Suppose *pj*(*ot|θj*) is normal PDF whose parameter is *θj* = (*mj*, *σj*2) where *mj* and *σj*2 are mean and variance, respectively.

Note that notation “*exp*” denotes exponential function. The equation specified by formula [II.5](#_Formula_IV.5.5._Equation) is re-written with regard to parameter *mj* as follows:

Therefore, the estimate is

The equation specified by formula [II.5](#_Formula_IV.5.5._Equation) is re-written with regard to parameter *σj*2 as follows:

It implies that given the estimate , the estimate is:

In general, the normal parameter estimate is:

Suppose *pj*(*ot|θj*) is exponential PDF whose parameter is *θj* = *κj* as follows:

Note that notations “*exp*” and “*e*(.)” denote exponential function. The equation specified by formula [II.5](#_Formula_IV.5.5._Equation) is re-written with regard to parameter *κj* as follows:

Therefore, the exponential parameter estimate is

Shortly, the continuous observation HMM parameter estimate with single PDF given current parameter Δ = (*aij*, *bi*(*k*), *πi*) is specified by formula [II.6](#_Formula_IV.5.6._Continuously).

**Formula II.6.** Continuous observation HMM parameter estimate with single PDF

Where joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) based on single PDF *pj*(*ot|θj*) is specified by formula [II.2](#_Formula_IV.5.2._Joint).

The EM algorithm applied into learning continuous observation HMM parameter with single PDF is described in table [II.1](#_Table_IV.5.1._EM).

|  |
| --- |
| Starting with initial value for Δ, each iteration in EM algorithm has two steps:   1. *E-step*: Calculating the joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) according to formula [II.2](#_Formula_IV.5.2._Joint). 2. *M-step*: Calculating the estimate based on the joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) determined at E-step, according to formula [II.6](#_Formula_IV.5.6._Continuously).The estimate becomes the current parameter for next iteration.   EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter Δ and next parameter is insignificant. It is possible to define a custom terminating condition. The terminating criterion *P*(*O|*Δ) described in table [I.3.2.2](#_Table_IV.4.3.2.2._Proposed) is a suggestion. |

###### **Table II.1.** EM algorithm applied into learning continuous observation HMM parameter with single PDF

Going back the weather example, there are some states of weather: *sunny*, *cloudy*, and *rainy*. Suppose you are in the room and do not know the weather outside but you are notified air humidity measures as observations from someone else. You can forecast weather based on humidity. However, humidity is not still categorized into discrete values such as *dry*, *dryish*, *damp*, and *soggy*. The humidity is now continuous real number, which is used to illustrate continuous observation HMM. It is required to discuss humidity a little bit.

*Absolute humidity* of atmosphere is measured as amount of water vapor (kilogram) in 1 cubic meter (m3) volume of air (Gallová & Kučerka).

Where *mw* is the amount of water vapor and *V* is the volume of air. The SI unit (NIST, 2008) of absolute humidity is kg/m3.

The amount of water vapor in the air conforms to environment conditions such as temperature and pressure. Given environment conditions, there is a saturation point at which absolute humidity becomes maximal, denoted . *Relative humidity* is ratio of the absolute humidity *h* to its maximal value (Gallová & Kučerka).

The relative humidity *rh* is always less than or equal to 1. Relative humidity *rh* is near to 0 then, the air is dry. Relative humidity *rh* is near to 1 then, the air is soggy. It is comfortable for human if relative humidity is between 0.5 and 0.7. Relative humidity is used in our weather example instead of absolute humidity. Suppose continuous observation sequence is

*O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38}

These observations are relative humidity measures. The bias for all measures is *ε=*0.01, for example, the first observation *o*1=0.88 ranges in interval [0.88–0.01, 0.88+0.01]. Given weather HMM ∆ whose parameters *A* and∏ specified in tables [I.1](#_Table_IV.4.1._Transition) and [I.2](#_Table_IV.4.2._Uniform) is shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Weather current day  (Time point *t*) | | |
| *sunny* | *cloudy* | *rainy* |
| Weather previous day  (Time point *t* –1) | *sunny* | *a*11=0.50 | *a*12=0.25 | *a*13=0.25 |
| *cloudy* | *a*21=0.30 | *a*22=0.40 | *a*23=0.30 |
| *rainy* | *a*31=0.25 | *a*32=0.25 | *a*33=0.50 |

|  |  |  |
| --- | --- | --- |
| *sunny* | *cloudy* | *rainy* |
| *π*1=0.33 | *π*2=0.33 | *π*3=0.33 |

The observation probability distribution *B* now includes three normal PDF (s): , , and corresponding to three states: *s*1*=sunny*, *s*2*=cloudy*, and *s*3=*rainy*. Following is the specification of these normal PDF (s).

Where , , and are means and variances of , , and .

As a convention, observation PDF (s) such as , , and are represented by theirs means and variances , , and . These means and variances are also called *observation probability parameters* that substitute for discrete matrix *B*. Table [II.2](#_Table_IV.5.2._Observation) shows observation probability parameters for our weather example.

|  |  |  |
| --- | --- | --- |
| *p*1(*ot|θ*1) | *m*1 = 0.87 | *σ*12 = 0.9 |
| *p*2(*ot|θ*2) | *m*2 = 0.14 | *σ*12 = 0.9 |
| *p*3(*ot|θ*3) | *m*3 = 0.39 | *σ*12 = 0.9 |

###### **Table II.2.** Observation probability parameters for weather example

Obviously, we have:

EM algorithm described in table [II.1](#_Table_IV.5.1._EM) is applied into calculating the parameter estimate given continuous observation sequence *O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38} and continuous normal PDF (s) whose means and variances shown in table [II.2](#_Table_IV.5.2._Observation). For convenience, all floating-point values are rounded off until ten decimal numbers.

As a convention, let

**At the first iteration** (*r=*1) we have:

Within the E-step of the first iteration (*r=*1), the terminating criterion *P*(*O|*Δ) is calculated according to forward-backward procedure (see table [I.1.1](#_Table_III.2.4._Forward-backward)) as follows:

Within the E-step of the first iteration (*r=*1), the joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) are calculated based on formula [II.2](#_Formula_IV.5.2._Joint) as follows:

Within the M-step of the first iteration (*r=*1), the estimate is calculated based on joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) determined at E-step.

**At the second iteration** (*r=*2), the current parameter Δ = (*aij*, *θj*, *πj*) is received values from the previous estimate , as seen in table [II.3](#_Table_IV.5.3._Continuously).

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000004341 | | |

###### **Table II.3.** Continuous observation HMM parameters resulted from the first iteration of EM algorithm

We have:

Within the E-step of the second iteration (*r=*2), the terminating criterion *P*(*O|*Δ) is calculated according to forward-backward procedure (see table [I.1.1](#_Table_III.2.4._Forward-backward)) as follows:

Within the E-step of the second iteration (*r=*2), the joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) are calculated based on formula [II.2](#_Formula_IV.5.2._Joint) as follows:

Within the M-step of the second iteration (*r=*2), the estimate is calculated based on joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) determined at E-step.

Table [II.4](#_Table_IV.5.4._Continuously) summarizes HMM parameters resulted from the first iteration and the second iteration of EM algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | HMM parameters | | |
| 1st |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000004341 | | |
|  | | | |
| 2nd |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000037839 | | |

###### **Table II.4.** Continuous observation HMM parameters resulted from the first iteration and the second iteration of EM algorithm

As seen in table [II.4](#_Table_IV.5.4._Continuously), the EM algorithm does not converge yet when it produces two different terminating criteria at the first iteration and the second iteration. It is necessary to run more iterations so as to gain the most optimal estimate. Within this example, the EM algorithm converges absolutely after 14 iterations when the criterion *P*(*O*|Δ) approaches to the same value 1 at the 13rd and 14th iterations. Table [II.5](#_Table_IV.5.5._Continuously) shows HMM parameter estimates along with terminating criterion *P*(*O|*Δ) at the 1st, 2nd, 13rd, and 14th iterations of EM algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | HMM parameters | | |
| 1st |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000004341 | | |
| 2nd |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000037839 | | |
| 13rd |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 1 | | |
| 14th |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 1 | | |

###### **Table II.5.** Continuous observation HMM parameters along with terminating criteria after 14 iterations of EM algorithm

Note that the format like “4.2*e* – 09” indicates scientific notation for real number, namely, 4.2*e* – 09 = 4.2\*10–9.

As a result, the learned parameters *A*, *B*, and ∏ are shown in table [II.6](#_Table_IV.5.6._Continuously):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Weather current day  (Time point *t*) | | |
| *sunny* | *cloudy* | *rainy* |
| Weather previous day  (Time point *t* –1) | *sunny* | *a*11=0 | *a*12=1 | *a*13=0 |
| *cloudy* | *a*21=0 | *a*22=0 | *a*23=1 |
| *rainy* | *a*31=0 | *a*32=0 | *a*33=1 |

|  |  |  |
| --- | --- | --- |
| *sunny* | *cloudy* | *rainy* |
| *π*1=1 | *π*2=0 | *π*3=0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Humidity | | |
| Weather | *sunny* | *p*1(*ot|θ*1) | *m*1 = 0.88 | *σ*12 = 4.2*e* – 09 |
| *cloudy* | *p*2(*ot|θ*2) | *m*2 = 0.13 | *σ*12 = 1.4*e* –14 |
| *rainy* | *p*3(*ot|θ*3) | *m*3 = 0.38 | *σ*12 = 1.0*e* –21 |

###### **Table II.6.** Continuous observation HMM parameters of weather example learned from EM algorithm

Such learned parameters are more appropriate to the continuous observation sequence *O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38} than the original ones shown in tables [I.1](#_Table_III.2.1._Transition), [I.2](#_Table_III.2.2._Uniform), and [II.2](#_Table_IV.5.2._Observation). In this section, observation conforms to a single PDF. Next section mentions mixture HMM in which observation is characterized by the mixture model of partial PDF (s).

## III. Mixture hidden Markov model

Suppose observation *ot* is result of experiment that is done at laboratory with *K* methods or equipments. Each observation is product of *K* methods or equipments. Given aforementioned weather example, humidity is measured at time point *t* by *K*=2 equipments. Equipment 1 and equipment 2 produces two humidity measures *a*1=0.63 and *a*2=0.88, respectively at time point *t*. Hence, the observation *ot* receives a random value among *a*1=0.63 and *a*2=0.88. If there is 60% that equipment 1 is selected then, we have *ot*=*a*1=0.63 with confidence 60%. The percentage 60% is called normalized weight of equipment 1 and so, it is easy to infer that normalized weight of equipment 2 is 40%.

Shortly, observation *ot* is created from mixture of *K* methods or equipments. If the observation *ot* is characterized by a PDF as aforementioned in formula [II.1](#_Formula_IV.5.1._Probability_1), this PDF is constituted of *K* partial PDF (s) and each partial PDF corresponds to one method or equipment. Such PDF is called *mixture model PDF*, specified by formula [III.1](#_Formula_IV.5.7._Mixture) (Rabiner, 1989, p. 267).

Where are non-negative *normalized weights*, such that

**Formula III.1.** Mixture model probability density function (PDF) of observation

Each *partial PDF* belongs to any probability distribution, for example, normal distribution, exponential distribution, etc. It is not specified that *K* partial PDF (s) constructing the formula [III.1](#_Formula_IV.5.7._Mixture) must have the same type of distribution but they often share the same distribution type in practice. For example, humidity is measured by *K*=2 equipments where the first equipment produces values conforming normal distribution with mean 0 and variance 1 while the second equipment produces values conforming normal distribution with mean 0.5 and variance 2. The case that two equipments measuring the same metric like humidity produce values in accordance with two different distributions (for example, normal distribution and exponential distribution) is very rare or impossible. The notation denotes *partial probabilistic parameters*, for instance, if is normal distribution PDF, includes mean and variance . The HMM now is specified by parameter , which is called *mixture continuous observation HMM*. The main subject of learning problem is to calculate partial estimates and when estimates and are kept intact, as aforementioned in formula [I.3.2.7](#_Formula_IV.4.3.7._HMM).

The joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) according to formulas [II.2](#_Formula_IV.5.2._Joint).

Let *γt*(*j*, *k*) be joint probability that the stochastic process counting for *kth* partial PDF is in state *si* at time point *t* with observation sequence *O*. Let *ξt*(*i*, *j*, *k*) be the joint probability that the stochastic process counting for *kth* receives state *si* at time point *t–*1 and state *sj* at time point *t* given observation sequence *O*. Formula [III.2](#_Formula_IV.5.8._Partial) specifies partial *γt*(*j*, *k*) and *ξt*(*i*, *j*, *k*).

Where partial forward variable *αt+*1(*j*, *k*) and partial backward variable *βt*(*i*, *k*) count for *kth* partial PDF.

**Formula III.2.** Partial joint probabilities *ξt*(*i*, *j*, *k*) and *γt*(*j*, *k*) based on mixture model PDF

Note that and are mean and variance of the normal PDF and Φ is cumulative standard normal distribution (Montgomery & Runger, 2003, p. 653).

After comparing formula [II.2](#_Formula_IV.5.2._Joint) with formula [III.2](#_Formula_IV.5.8._Partial), it is easy to draw the relationship between quantities *ξt*(*i*, *j*), *γt*(*j*) and partial quantities *ξt*(*i*, *j*, *k*), *γt*(*j*, *k*), as seen in formula [III.3](#_Formula_IV.5.9._Relationship).

**Formula III.3.** Relationship between quantities *ξt*(*i*, *j*), *γt*(*j*) and partial quantities *ξt*(*i*, *j*, *k*), *γt*(*j*, *k*)

The ratio reflects how much the partial PDF contributes to the mixture model PDF . Following is the proof of formula [III.3](#_Formula_IV.5.9._Relationship).

Given *t* ≥ 2, we have:

(The integral shown in formula [III.2](#_Formula_IV.5.8._Partial) is turned back the original PDF )

(due to according to formula [III.1](#_Formula_IV.5.7._Mixture))

We also have:

(due to according to formula [I.3.2.6](#_Formula_IV.4.3.2.6._The))

(due to according to formula [I.3.2.6](#_Formula_IV.4.3.2.6._The))

Because estimates and are kept intact, it is necessary to calculate partial estimates and . Formula [III.4](#_Formula_IV.5.10._EM) specifies the EM expectation for continuous observation HMM given mixture model PDF.

**Formula III.4.** EM conditional expectation for continuous observation HMM given mixture model PDF

Where is index function such that,

Note that notation “*ln*” denotes natural logarithm function.

Derived from formula [I.3.2.3](#_Formula_IV.4.3.2.3._Lagrangian), the HMM Lagrangian function for continuous observation HMM given mixture model PDF and constraint   
 is specified by formula [III.5](#_Formula_IV.5.11._Lagrangian).

**Formula III.5.** Lagrangian function for continuous observation HMM with single PDF

Where *λ* is *n*-component vector *λ* = (*λ*1, *λ*2,…, *λn*) and *μ* is *n*-component vector *μ* = (*μ*1, *μ*2,…, *μn*). Factors *λi* ≥ 0 and *μj* ≥ 0 are called Lagrange multipliers or Karush-Kuhn-Tucker multipliers (Wikipedia, Karush–Kuhn–Tucker conditions, 2014) or dual variables.

The estimate which is extreme point of the Lagrangian function *l*(Δ, *λ*, *μ*) is determined by setting partial derivatives of *l*(Δ, *λ*, *μ*) with respect to to be zero. The partial derivative of *l*(Δ, *λ*) with respect to is:

(Due to total probability rule)

Where is index function such that,

It implies

(Due to multiplication rule)

(Where is the joint probability of state *xt=sj* and observation sequence *O* counting for *kth* PDF)

(Due to according to formula [II.8](#_Formula_IV.5.8._Partial))

Setting the partial derivative to be zero, we get the equation whose solution is estimate , as follows:

Summing the expression over *K* mixture components, we have:

(Due to constraint )

Substituting into equation , we have:

It implies that the weight estimate is specified by formula [III.6](#_Formula_IV.5.12._Weight):

**Formula III.6.** Weight estimate of partial PDF

The partial joint probabilities *γt*(*j*, *k*) is determined by formula [III.2](#_Formula_IV.5.8._Partial).

The parameter estimate which is extreme point of the Lagrangian function *l*(Δ, *λ*, *μ*) is determined by setting partial derivatives of *l*(Δ, *λ*, *μ*) with respect to to be zero. The partial derivative of *l*(Δ, *λ*, *μ*) with respect to is:

(Due to total probability rule)

Where is index function such that,

It implies

(Due to multiplication rule)

(Where is the joint probability of state *xt=sj* and observation sequence *O* counting for *kth* PDF)

(Due to according to formula [III.2](#_Formula_IV.5.8._Partial))

Setting the partial derivative to be zero, we get the equation whose solution is estimate , specified by formula [III.7](#_Formula_IV.5.13._Equation).

**Formula III.7.** Equation of partial PDF parameter

Note that notation “*ln*” denotes natural logarithm function.

The equation specified by formula is similar to the one specified by formula [II.5](#_Formula_IV.5.5._Equation) except that the single PDF is replaced by partial PDF . It is possible to solve the above equation (formula [III.7](#_Formula_IV.5.13._Equation)) by Newton-Raphson method (Burden & Faires, 2011, pp. 67-69) – a numeric analysis method but it is easier and simpler to find out more precise solution if the PDF belongs to well-known distributions: normal distribution (Montgomery & Runger, 2003, pp. 109-110), exponential distribution (Montgomery & Runger, 2003, pp. 122-123), etc. Please see formula [II.5](#_Formula_IV.5.5._Equation) for more details of such specific solutions.

Therefore, without of replication, if is normal PDF, its parameter estimate where is mean estimate and is variance estimate is:

If is exponential PDF, its parameter estimate is:

Shortly, the mixture continuous observation HMM estimate with mixture PDF given current parameter is specified by formula [III.8](#_Formula_IV.5.14._Continuously).

**Formula III.8.** Continuous observation HMM parameter estimate with mixture PDF

The joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) follow formulas [II.2](#_Formula_IV.5.2._Joint).

The partial joint probabilities *γt*(*j*, *k*) is determined by formula [III.2](#_Formula_IV.5.8._Partial).

The EM algorithm applied into learning continuous observation HMM parameter with mixture PDF is described in table [III.1](#_Table_IV.5.6._EM).

|  |
| --- |
| Starting with initial value for Δ, each iteration in EM algorithm has two steps:   1. *E-step*: Calculating the joint probabilities *ξt*(*i*, *j*), *γt*(*j*), and *γt*(*j*, *k*) according to formulas [II.2](#_Formula_IV.5.2._Joint) and [III.2](#_Formula_IV.5.8._Partial). 2. *M-step*: Calculating the estimate based on the joint probabilities *ξt*(*i*, *j*), *γt*(*j*), and *γt*(*j*, *k*) determined at E-step, according to formula [III.8](#_Formula_IV.5.14._Continuously). The estimate becomes the current parameter for next iteration.   EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter Δ and next parameter is insignificant. It is possible to define a custom terminating condition. The terminating criterion *P*(*O|*Δ) described in table [I.3.2.2](#_Table_IV.4.3.2.2._Proposed) is a suggestion. |

###### **Table III.1.** EM algorithm applied into learning continuous observation HMM parameter with mixture PDF

Going back given HMM Δ whose parameters *A* and∏ are specified in tables [I.1](#_Table_III.2.1._Transition) and [I.2](#_Table_III.2.2._Uniform), suppose continuous observation sequence is *O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38, the EM algorithm described in table [III.1](#_Table_IV.5.6._EM) is applied into calculating the parameter estimate .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Weather current day  (Time point *t*) | | |
| *sunny* | *cloudy* | *rainy* |
| Weather previous day  (Time point *t* –1) | *sunny* | *a*11=0.50 | *a*12=0.25 | *a*13=0.25 |
| *cloudy* | *a*21=0.30 | *a*22=0.40 | *a*23=0.30 |
| *rainy* | *a*31=0.25 | *a*32=0.25 | *a*33=0.50 |

|  |  |  |
| --- | --- | --- |
| *sunny* | *cloudy* | *rainy* |
| *π*1=0.33 | *π*2=0.33 | *π*3=0.33 |

Recall that observations are relative humidity measures. The bias for all measures is *ε=*0.01, for example, the first observation *o*1=0.88 ranges in interval [0.88–0.01, 0.88+0.01]. Humidity is measured by *K*=2 equipments. The normalized weights of equipments 1 and 2 are 60% and 40%, respectively. Shortly, observation *ot* is created from mixture of *K*=2 equipments. The observation probability distribution *B* is now includes three normal mixture PDF (s): , , and corresponding to three states: *s*1*=sunny*, *s*2*=cloudy*, and *s*3=*rainy*. Following is specification of these normal PDF (s).

Note that , , and are normalized weights; , , and are normal partial PDF (s).

Where , , and are means and variances of , , and .

As a convention, normal mixture PDF (s) such as , , and are represented by theirs weights, partial means and partial variances.

These weights, means and variances are also called observation probability parameters that substitute for discrete matrix *B*. Table [III.2](#_Table_IV.5.8._Observation) shows observation probability parameters for our weather example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *p*1(*ot|θ*1) |  |  |  |  |  |  |
| *p*2(*ot|θ*2) |  |  |  |  |  |  |
| *p*3(*ot|θ*3) |  |  |  |  |  |  |

###### **Table III.2.** Observation probability parameters for weather example in case of mixture model

Obviously, we have:

EM algorithm described in table [III.1](#_Table_IV.5.6._EM) is applied into calculating the parameter estimate given continuous observation sequence *O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38} and mixture normal PDF (s) whose means and variances shown in table [III.2](#_Table_IV.5.8._Observation). For convenience, all floating-point values are rounded off until ten decimal numbers.

As a convention, let

**At the first iteration** (*r=*1) we have:

Within the E-step of the first iteration (*r=*1), the terminating criterion *P*(*O|*Δ) is calculated according to forward-backward procedure (see table [I.1.1](#_Table_III.2.4._Forward-backward)) as follows:

Within the E-step of the first iteration (*r=*1), the joint probabilities *ξt*(*i*,*j*), *γt*(*j*), and *γt*(*j*,*k*) are calculated based on formulas [II.2](#_Formula_IV.5.2._Joint) and [III.2](#_Formula_IV.5.8._Partial) as follows:

Within the M-step of the first iteration (*r=*1), the estimate is calculated based on joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) determined at E-step.

**At the second iteration** (*r=*2), the current parameter Δ = (*aij*, *θj*, *πj*) is received values from the previous estimate , as seen in table [III.3](#_Table_IV.5.9._Mixture).

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000003847 | | |

###### **Table III.3.** Mixture HMM parameters resulted from the first iteration of EM algorithm

We have:

Within the E-step of the second iteration (*r=*2), the terminating criterion *P*(*O|*Δ) is calculated according to forward-backward procedure (see table [I.1.1](#_Table_III.2.4._Forward-backward)) as follows:

Within the E-step of the second iteration (*r=*2), the joint probabilities *ξt*(*i*,*j*), *γt*(*j*), and *γt*(*j*,*k*) are calculated based on formulas [II.2](#_Formula_IV.5.2._Joint) and [III.2](#_Formula_IV.5.8._Partial) as follows:

Within the M-step of the second iteration (*r=*2), the estimate is calculated based on joint probabilities *ξt*(*i*,*j*) and *γt*(*j*) determined at E-step.

Table [III.4](#_Table_IV.5.10._Mixture) summarizes mixture HMM parameters resulted from the first iteration and the second iteration of EM algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | HMM parameters | | |
| 1st |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000003847 | | |
|  | | | |
| 2nd |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000037929 | | |

###### **Table III.4.** Mixture HMM parameters resulted from the first iteration and the second iteration of EM algorithm

As seen in table [III.4](#_Table_IV.5.10._Mixture), the EM algorithm does not converge yet when it produces two different terminating criteria at the first iteration and the second iteration. It is necessary to run more iterations so as to gain the most optimal estimate. Within this example, the EM algorithm converges absolutely after 11 iterations when the criterion *P*(*O*|Δ) approaches to value 1 at the 10th and 11st iterations. Table [III.5](#_Table_IV.5.11._Mixture) shows mixture HMM parameter estimates along with terminating criterion *P*(*O|*Δ) at the 1st, 2nd, 10th, and 11st iterations of EM algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | HMM parameters | | |
| 1st |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000003847 | | |
|  | | | |
| 2nd |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 0.0000037929 | | |
|  | | | |
| 10th |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 1 | | |
|  | | | |
| 11st |  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | | |
|  |  |  |
|  | | |
| Terminating criterion *P*(*O|*Δ) = 1 | | |

###### **Table III.5.** Mixture HMM parameters along with terminating criteria after 14 iterations of EM algorithm

Note that the format like “4.1*e* – 16” indicates scientific notation for real number, namely, 4.1*e* – 16 = 4.1\*10–16.

As a result, the learned parameters *A*, *B*, and ∏ are shown in table [III.6](#_Table_IV.5.12._Mixture):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Weather current day  (Time point *t*) | | |
| *sunny* | *cloudy* | *rainy* |
| Weather previous day  (Time point *t* –1) | *sunny* | *a*11=0.998665 | *a*12=0 | *a*13=0.001335 |
| *cloudy* | *a*21=0 | *a*22=0 | *a*23=1 |
| *rainy* | *a*31=1 | *a*32=0 | *a*33=0 |

|  |  |  |
| --- | --- | --- |
| *sunny* | *cloudy* | *rainy* |
| *π*1=0 | *π*2=1 | *π*3=0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Humidity | | | |
| Weather | *sunny* | *p*1(*ot|θ*1) |  |  |  |
|  |  |  |
| *cloudy* | *p*2(*ot|θ*2) |  |  |  |
|  |  |  |
| *rainy* | *p*3(*ot|θ*3) |  |  |  |
|  |  |  |

###### **Table III.6.** Mixture HMM parameters of weather example learned from EM algorithm

Such learned parameters are more appropriate to the continuous observation sequence *O* = {*o*1=0.88, *o*2=0.13, *o*3=0.38} than the original ones shown in tables [I.1](#_Table_III.2.1._Transition), [I.2](#_Table_III.2.2._Uniform), and [III.2](#_Table_IV.5.8._Observation). This section ends up full description of continuous observation HMM. Next section is the discussion and conclusion.

## IV. Discussion and Conclusion

My main contribution is to propose the practical technique to calculate basic quantities such as forward variable *αt*, backward variable *βt*, and joint probabilities *ξt*, *γt* based on integral of observation PDF within a very small velocity of observation value. These quantities are necessary to evaluation problem, uncovering problem, and learning problem of HMM. However, the most hazardous issue of continuous observation HMM is to determine parameter estimate of the observation PDF *p*(*ot|θ*). In previous sections, it is easy to solve the equation specified by formulas [II.5](#_Formula_IV.5.5._Equation) and [III.7](#_Formula_IV.5.13._Equation) to find out if the PDF *p*(*ot|θ*) belongs to normal/exponential distributions when the natural logarithm function eliminates “exponent form” of these distributions, which in turn leads to take derivatives easier. Otherwise, if such PDF belongs to complicated distributions, a new hazardous problem is issued when equation [II.5](#_Formula_IV.5.5._Equation) is relevant to complex derivatives as follows:

The suggestion of using Newton-Raphson method (Burden & Faires, 2011, pp. 67-69) to solve such equation is not optimal solution. Your attention please, we should concern the “exponent form” of normal/exponential distributions. The recognition of “exponent form” is very important. In fact, probabilistic distributions such as normal and exponent belong to exponential family of distribution. We should take advantages of exponential family instead of finding the most general method to solve formula [II.5](#_Formula_IV.5.5._Equation). Therefore, this section focuses on exponential family of observation PDF. Later on, we will know that exponential family ranges over most of common distributions. In other words, solving successfully formula [II.5](#_Formula_IV.5.5._Equation) with regard to exponential family is close to finding the general method for determining parameter estimate .

According to (Wikipedia, Exponential family, 2016), exponential family refers to a set of probabilistic distributions whose PDF (s) have the same exponential form as follows:

Where,

* The *θ* = (*θ*1, *θ*2,…, *θn*) is vector of *n* parameters *θi* (s). Note that a scalar known as a number is considered as 1-element vector.
* The *h*(*x*) is a function of *x*, which is called base measure.
* The *η*(*θ*) is function of *θ*, which is a vector. If *η*(*θ*) = *θ*, the PDF is called canonical. The function *η*(*θ*) is known as natural parameter *η* whose dimension may be larger than *θ*.
* The *T*(*x*) is function of *x*, which is sufficient statistic.
* The notation denotes scalar product of *η*(*θ*) and *T*(*x*). Note that the scalar product of two scalars is multiplication of such two scalars.
* The *A*(*θ*) is log-partition function which is used to normalize PDF. It will be mentioned later on.

All functions *η*(*θ*), *T*(*x*), and *A*(*η*(*θ*)) are known. Formula [IV.1](#_Formula_IV.1._Canonical) specifies the exponential family PDF with note that function *η*(*θ*) is considered as natural parameter *η*.

Where *η* is natural parameter which is function of *θ* and so *A*(*η*) is always specified by function of *θ*.

**Formula IV.1.** Exponential family PDF

According to (Wikipedia, Exponential family, 2016), most of common distributions belong to exponential family, for instance: normal, log-normal, exponential, gamma, chi-squared, beta, Dirichlet, Bernoulli, categorical, Poisson, geometric, inverse Gaussian, von Mises, von Mises-Fisher, Wishart, Inverse Wishart, binomial with fixed number of trials, multinomial with fixed number of trials, negative binomial with fixed number of failures.

The log-partition function *A*(*η*) is determined based on a property of PDF that integral of any PDF from negative infinity to positive infinity is 1.

It implies *A*(*η*) is determined by formula [IV.2](#_Formula_IV.2._Generating) (Jebara, 2015, p. 7).

**Formula IV.2.** Log-partition function *A*(*η*)

According to author (Jebara, 2015, p. 7), the first derivative of *A*(*η*) with regard to *η* is:

In general, the first derivative of *A*(*η*) is the expectation of *T*(*x*) given exponential family PDF according to formula [IV.3](#_Formula_IV.3._First) as follows:

**Formula IV.3.** First derivative of log-partition function *A*(*η*) with regard to *η*

Note that the exponential family PDF is specified by formula [IV.1](#_Formula_IV.1._Canonical) and it is possible to determine the derivative if *A*(*η*) is known; for example, function *A*(*η*) of normal PDF with mean *μ* and variance *σ*2 is . It is not always to calculate based on the expectation .

Let *LnL*(*η*) be the log-likelihood of exponential family PDF which is natural logarithm of exponential family PDF as seen in formula [IV.4](#_Formula_IV.4._Log-likelihood).

**Formula IV.4.** Log-likelihood function of exponential family PDF

Note that *LnL*(*η*) is function of parameters *η* and it is also a vector. The first derivative of *LnL*(*η*) is specified by formula [IV.5](#_Formula_IV.5._Derivative).

**Formula IV.5.** Derivative of log-likelihood function of exponential family PDF

The formula [II.5](#_Formula_IV.5.5._Equation) becomes formula [IV.6](#_Formula_IV.6._Equation) as follows:

**Formula IV.6.** Equation of exponential family PDF parameters

Where *ot* is continuous observation, *T* is the number of observations and *γt* is the joint probability specified by formula [I.2.1](#_Formula_IV.4.2.1._Probability). As seen in section [II](#_IV.5._Continuously_observational), formulas [II.5](#_Formula_IV.5.5._Equation), [III.7](#_Formula_IV.5.13._Equation) and [IV.6](#_Formula_IV.6._Equation) are derived from EM algorithm and so they are essentially maximum likelihood equations with additional information of joint probabilities *γt* (s). Please refer to article “Maximum Likelihood from Incomplete Data via the EM Algorithm” by author (Dempster, Laird, & Rubin, 1977, p. 5) for more details about maximum likelihood method and EM algorithm.

For example, we have normal PDF with mean *μ* and variance *σ*2. Its specification includes:

According to formula [IV.1](#_Formula_IV.1._Canonical), the normal PDF is:

We have:

The derivative of *A*(*η*) of normal PDF is a vector as follows:

We have the same result if taking directly derivative on *A*(*η*):

The derivative of log-likelihood of normal PDF is:

Now we solve formula [IV.6](#_Formula_IV.6._Equation) given normal observation PDF *p*(*ot|θ*) with mean *μ* and variance *σ*2. The natural parameter of such PDF is:

We have:

The estimate is the same to one in formula [II.6](#_Formula_IV.5.6._Continuously) but the estimate is different. The previous estimate specified by formula [II.6](#_Formula_IV.5.6._Continuously) is:

However, our method is still correct because the expectation of is kept intact. In fact, we have:

In general, it is easy for us to solve estimation equation [IV.6](#_Formula_IV.6._Equation) for finding parameter estimates or to know that the equation [IV.6](#_Formula_IV.6._Equation) is impossible if the continuous PDF belongs to exponential family because the derivative of log-likelihood specified by formula [IV.5](#_Formula_IV.5._Derivative) always exists and it is totally possible to calculate the expectation of function *T*(*x*).

Now there is a question “how to estimate parameters of a PDF which does not conform to exponential family, for instance, Student’s t-distribution, F-distribution, Cauchy distribution (Wikipedia, Exponential family, 2016). In that case, we will solve formula [II.5](#_Formula_IV.5.5._Equation) with subject to each concrete distribution. The research gives an example of estimating t-distribution parameters. Similarly, our main point is to calculate the derivative *DLnL*(*θ*) of log-likelihood function of t-distribution. Formula [IV.7](#_Formula_IV.7._Student’s) specifies the PDF of t-distribution (Wikipedia, Student's t-distribution, 2016).

Where *v* is the number of degrees of freedom and Γ is gamma function.

**Formula IV.7.** Student’s t-distribution

Following are gamma function Γ(*x*) and digamma function *ψ*(*x*) which is derivative of natural logarithm of gamma function. Trigamma function *ψ*1(*x*) is derivative of digamma function *ψ*(*x*). Formula [IV.8](#_Formula_IV.8._Functions) specifies Γ(*x*), *ψ*(*x*), and *ψ*1(*x*) (Nguyen, Beta Likelihood Estimation and Its Application to Specify Prior Probabilities in Bayesian Network, 2016).

**Formula IV.8.** Functions gamma, digamma, trigamma

The t-distribution replaces normal distribution for testing on means of normal population in case of small size samples. It is important distribution to hypothesis testing. When *v* is parameter of t-distribution, we need to estimate it according to formula [II.5](#_Formula_IV.5.5._Equation). The log-likelihood of t-distribution PDF is:

The derivative *DLnL*(*v*) of log-likelihood function is:

Given the t-distribution PDF of observation, the formula [II.5](#_Formula_IV.5.5._Equation) to find out the parameter estimate becomes:

Shortly, the estimate is solution of equation specified by formula [IV.9](#_Formula_IV.9._Equation).

**Formula IV.9.** Equation of t-distribution PDF parameter

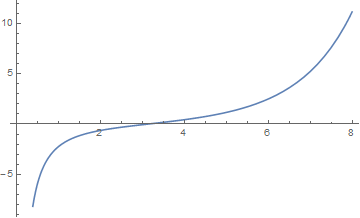
Where *ot* is continuous observation, *T* is the number of observations and *γt* is the joint probability specified by formula [I.2.1](#_Formula_IV.4.2.1._Probability).

The formula [IV.9](#_Formula_IV.9._Equation) is much more complicated than formula [IV.6](#_Formula_IV.6._Equation). As aforementioned suggestion, the Newton-Raphson method (Burden & Faires, 2011, pp. 67-69) is used to find solution of given equation *g*(*v*)=0 along with the tangent of *g*(*v*). It starts with an arbitrary value of *v*0 as a solution candidate. Suppose the current value is *vk*, the next value *vk*+1 is calculated based on following formula:

The value *vk* is solution of *g*(*v*)=0 if *g*(*vk*)=0 which means that *vk*+1=*vk*. The most important aspect of Newton-Raphson method is that derivative of *g*(*v*) must be determinate within domain of *g*(*v*). Fortunately, the derivative *g’*(*v*) is always determinate for all *v* > 0.

Suppose there is only one observation *o*1=1 and the joint probability *γ*1 is 1, we have:

Figure [IV.1](#_Figure_IV.1._Graph) shows function *g*(*v*) with *o*1=1 and *γ*1=1.



**Figure IV.1.** Graph of t-distribution parameter function

By applying the Newton-Raphson method, we receive the solution of *g*(*v*)=0 given starting value 3, after 6 times to run.

In conclusion, it is optimal if observation PDF belongs to exponential family where the derivative *DLnL*(*η*) of log-likelihood specified by formula [IV.5](#_Formula_IV.5._Derivative) is simple, which makes solving estimation equation easier. Otherwise, Newton-Raphson method is a work-around because it can find out solution if the target function is increasing (decreasing) within the interval containing solution. We cannot solve effectively an arbitrary equation. In future, I try my best to prove whether the solution of general estimation equation specified by formula [II.5](#_Formula_IV.5.5._Equation) exists. When asserting the existence of solution, we can apply artificial intelligence approaches such as neural network and genetic algorithm into solving formula [II.5](#_Formula_IV.5.5._Equation) instead of using only Newton-Raphson method.

## List of Formulas

[**Formula I.1.1.** Forward variable](#_Toc451437819)

[**Formula I.1.2.** Recurrence property of forward variable](#_Toc451437820)

[**Formula I.1.3.** Probability *P*(*O|*Δ) based on forward variable](#_Toc451437821)

[**Formula I.1.4.** Backward variable](#_Toc451437822)

[**Formula I.1.5.** Recurrence property of backward variable](#_Toc451437823)

[**Formula I.1.6.** Probability *P*(*O|*Δ) based on backward variable](#_Toc451437824)

[**Formula I.2.1.** Joint probability of being in state *si* at time point *t* with observation sequence *O*](#_Toc451437825)

[**Formula I.2.2.** Optimal state at time point *t*](#_Toc451437826)

[**Formula I.2.3.** Joint optimal criterion at time point *t*](#_Toc451437827)

[**Formula I.2.4.** Recurrence property of joint optimal criterion](#_Toc451437828)

[**Formula I.2.5.** Backtracking state](#_Toc451437829)

[**Formula I.3.1.1.** EM optimization criterion based on conditional expectation](#_Toc451437830)

[**Formula I.3.2.1.** General EM conditional expectation for HMM](#_Toc451437831)

[**Formula I.3.2.2.** EM conditional expectation for HMM](#_Toc451437832)

[**Formula I.3.2.3.** Lagrangian function for HMM](#_Toc451437833)

[**Formula I.3.2.4.** HMM parameter estimate](#_Toc451437834)

[**Formula I.3.2.5.** Joint probability *ξt*(*i*, *j*)](#_Toc451437835)

[**Formula I.3.2.6.** The *γt* is sum of *ξt* over all states](#_Toc451437836)

[**Formula I.3.2.7.** HMM parameter estimate in detailed](#_Toc451437837)

[**Formula II.1.** Probability density function (PDF) of observation](#_Toc451437838)

[**Formula II.2.** Joint probabilities *ξt*(*i*, *j*) and *γt*(*j*) based on single PDF](#_Toc451437839)

[**Formula II.3.** EM conditional expectation for continuous observation HMM with single PDF](#_Toc451437840)

[**Formula II.4.** Lagrangian function for continuous observation HMM with single PDF](#_Toc451437841)

[**Formula II.5.** Equation of single PDF parameter](#_Toc451437842)

[**Formula II.6.** Continuous observation HMM parameter estimate with single PDF](#_Toc451437843)

[**Formula III.1.** Mixture model probability density function (PDF) of observation](#_Toc451437844)

[**Formula III.2.** Partial joint probabilities *ξt*(*i*, *j*, *k*) and *γt*(*j*, *k*) based on mixture model PDF](#_Toc451437845)

[**Formula III.3.** Relationship between quantities *ξt*(*i*, *j*), *γt*(*j*) and partial quantities *ξt*(*i*, *j*, *k*), *γt*(*j*, *k*)](#_Toc451437846)

[**Formula III.4.** EM conditional expectation for continuous observation HMM given mixture model PDF](#_Toc451437847)

[**Formula III.5.** Lagrangian function for continuous observation HMM with single PDF](#_Toc451437848)

[**Formula III.6.** Weight estimate of partial PDF](#_Toc451437849)

[**Formula III.7.** Equation of partial PDF parameter](#_Toc451437850)

[**Formula III.8.** Continuous observation HMM parameter estimate with mixture PDF](#_Toc451437851)

[**Formula IV.1.** Exponential family PDF](#_Toc451437852)

[**Formula IV.2.** Log-partition function *A*(*η*)](#_Toc451437853)

[**Formula IV.3.** First derivative of log-partition function *A*(*η*) with regard to *η*](#_Toc451437854)

[**Formula IV.4.** Log-likelihood function of exponential family PDF](#_Toc451437855)

[**Formula IV.5.** Derivative of log-likelihood function of exponential family PDF](#_Toc451437856)

[**Formula IV.6.** Equation of exponential family PDF parameters](#_Toc451437857)

[**Formula IV.7.** Student’s t-distribution](#_Toc451437858)

[**Formula IV.8.** Functions gamma, digamma, trigamma](#_Toc451437859)

[**Formula IV.9.** Equation of t-distribution PDF parameter](#_Toc451437860)

## References

Boyd, S., & Vandenberghe, L. (2009). *Convex Optimization.* New York, NY, USA: Cambridge University Press.

Burden, R. L., & Faires, D. J. (2011). *Numerical Analysis* (9th Edition ed.). (M. Julet, Ed.) Brooks/Cole Cengage Learning.

Cheng, C.-C., Sha, F., & Saul, L. K. (2009). A Fast Online Algorithm for Large Margin Training of Continuous Density Hidden Markov Models. *Proceedings of the Tenth Annual Conference of the International Speech Communication Association (INTERSPEECH 2009)*, (pp. 668-671). Brighton. Retrieved May 28, 2016, from http://cseweb.ucsd.edu/~saul/papers/is09\_hmm.pdf

Couvreur, C. (1996). *Hidden Markov Models and Their Mixtures.* Université Catholique de Louvain, Department of Methematics. Louvain-la-Neuve: ResearchGate. Retrieved May 7, 2016, from https://www.researchgate.net/publication/2677995\_Hidden\_Markov\_Models\_and\_Their\_Mixtures

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. (M. Stone, Ed.) *Journal of the Royal Statistical Society, Series B (Methodological), 39*(1), 1-38.

Fosler-Lussier, E. (1998). *Markov Models and Hidden Markov Models: A Brief Tutorial.* Technical Report TR-98-041, International Computer Science Institute, USA.

Gallová, J., & Kučerka, N. (n.d.). *Measurement of air humidity.* Technical Report, Comenius University in Bratislava, Faculty of Pharmacy.

Huo, Q., & Lee, C.-H. (1997, March). On-Line Adaptive Learning of the Continuous Density Hidden Markov Model Based on Approximate Recursive Bayes Estimate. *IEEE Transactions on Speech and Audio Processing, 5*(2), 161-172. doi:10.1109/89.554778

Jebara, T. (2015). *The Exponential Family of Distributions.* Columbia University, Computer Science Department. New York: Columbia Machine Learning Lab. Retrieved April 27, 2016, from http://www.cs.columbia.edu/~jebara/4771/tutorials/lecture12.pdf

Jia, Y.-B. (2013). *Lagrange Multipliers.* Lecture notes on course “Problem Solving Techniques for Applied Computer Science”, Iowa State University of Science and Technology, USA.

Lee, C. H., Rabiner, L. R., Pieraccini, R., & Wilpon, J. G. (1990, April). Acoustic modeling for large vocabulary speech recognition. *Computer Speech & Language, 4*(2), 127-165. doi:10.1016/0885-2308(90)90002-N

Montgomery, D. C., & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers* (3rd Edition ed.). New York, NY, USA: John Wiley & Sons, Inc.

Nguyen, L. (2016, April 27). Beta Likelihood Estimation and Its Application to Specify Prior Probabilities in Bayesian Network. (T.-X. He, P. Bracken, J. C. Valverde, & J. Li, Eds.) *British Journal of Mathematics, 16*(3). doi:10.9734/BJMCS/2016/25731

Nguyen, L. (2016). Tutorial on Hidden Markov Model. (L. Nguyen, & M. A. MELLAL, Eds.) *Special Issue "Some Novel Algorithms for Global Optimization and Relevant Subjects", Applied and Computational Mathematics (ACM)*.

NIST. (2008, March). International System of Units (SI). (330 - 2008 Edition). (B. N. Taylor, & A. Thompson, Eds.) Gaithersburg, Maryland, USA: National Institute of Standards and Technology.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. (J. H. Trussell, Ed.) *Proceedings of the IEEE, 77*(2), 257-286.

Ramage, D. (2007). *Hidden Markov Models Fundamentals.* Lecture Notes, Stanford University, USA.

Schmolze, J. G. (2001). *An Introduction to Hidden Markov Models.* Lecture Notes on course "COMP 232-Knowledge Based Systems", Department of Computer Science, Tufts University.

Sean, B. (2009). *The Expectation Maximization Algorithm - A short tutorial.* University of Notre Dame, Indiana, Department of Electrical Engineering. Sean Borman's Homepage.

Sha, F., & Saul, L. K. (2006). Large Margin Hidden Markov Models for Automatic Speech Recognition. (B. Schölkopf, J. C. Platt, & T. Hoffman, Eds.) *Advances in Neural Information Processing Systems (NIPS 2006), 19*, 1249-1256. Retrieved May 29, 2016, from http://machinelearning.wustl.edu/mlpapers/paper\_files/NIPS2006\_143.pdf

Sha, F., & Saul, L. K. (2009). Large margin training of continuous-density hidden Markov models. In J. Keshet, S. Bengio, J. Keshet, & S. Bengio (Eds.), *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods* (pp. 101-114). Chichester, UK: Wiley & Sons. doi:10.1002/9780470742044.ch7

Wikipedia. (2014, August 4). *Karush–Kuhn–Tucker conditions*. (Wikimedia Foundation) Retrieved November 16, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Karush–Kuhn–Tucker\_conditions

Wikipedia. (2016, March September). *Exponential family*. (Wikimedia Foundation) Retrieved 2015, from Wikipedia website: https://en.wikipedia.org/wiki/Exponential\_family

Wikipedia. (2016, April 12). *Student's t-distribution*. (Wikimedia Foundation) Retrieved April 24, 2016, from Wikipedia website: https://en.wikipedia.org/wiki/Student%27s\_t-distribution