Channels' Matching Algorithm for Mixture Models

Chenguang Lu(✉)

College of Intelligence Engineering and Mathematics,
Liaoning Engineering and Technology University,
Fuxin 123000, Liaoning, China
lcguang@foxmail.com

Abstract. To solve the Maximum Mutual Information (MMI) and Maximum Likelihood (ML) for tests, estimations, and mixture models, it is found that we can obtain a new iterative algorithm by the Semantic Mutual Information (SMI) and $R(G)$ function proposed by Chenguang Lu (1993) (where $R(G)$ function is an extension of information rate distortion function $R(D)$, G is the lower limit of the SMI, and $R(G)$ represents the minimum R for given G). This paper focus on mixture models. The SMI is defined by the average log normalized likelihood. The likelihood function is produced from the truth function and the prior by the semantic Bayesian inference. A group of truth functions constitute a semantic channel. Letting the semantic channel and Shannon channel mutually match and iterate, we can obtain the Shannon channel that maximizes the MMI and the average log likelihood. Therefore, this iterative algorithm is called Channels' Matching algorithm or the CM algorithm. It is proved that the relative entropy between the sampling distribution and predicted distribution may be equal to $R - G$. Hence, solving the maximum likelihood mixture model only needs minimizing $R - G$, without needing Jensen's inequality. The convergence can be intuitively explained and proved by the $R(G)$ function. Two iterative examples of mixture models (which are demonstrated in an excel file) show that the computation for the CM algorithm is simple. In most cases, the number of iterations for convergence (as the relative entropy <0.001 bit) is about 5. The CM algorithm is similar to the EM algorithm; however, the CM algorithm has better convergence and more potential applications.

Keywords: Shannon channel · Semantic channel · Semantic information · Likelihood · Mixture models · EM algorithm · Machine learning · Statistical inference

1 Introduction

To obtain maximum likelihood mixture models, The EM algorithm [1] and the Newton method [2] are often used. There have been many papers on applying or improving the EM algorithm. Lu proposed the semantic information measure (SIM) and the $R(G)$ function in 1993 [3–5]. The $R(G)$ function is an extension of Shannon's information rate distortion function $R(D)$ [6, 7]. The $R(G)$ means the minimum R for given SIM G. It is found that using SIM and $R(G)$ function, we can obtain a new iterative

Z. Shi et al. (Eds.): ICIS 2017, IFIP AICT 510, pp. 321–332, 2017.
DOI: 10.1007/978-3-319-68121-4_35

algorithm, i.e, Channels' Matching algorithm (or the CM algorithm). Compared with the EM algorithm, the CM algorithm proposed by this paper is seemly similar yet essentially different[1].

In this study, we use the sampling distribution instead of the sampling sequence. Assume the sampling distribution is $P(X)$ and the predicted distribution by the mixture model is $Q(X)$. The goal is to minimize the relative entropy or Kullback-Leibler (KL) divergence $H(Q||P)$ [8, 9]. With the semantic information method, we may prove $H(Q||P) = R(G) - G$. Then, maximizing G and modifying R alternatively, we can minimize $H(Q||P)$.

We first introduce the semantic channel, semantic information measure, and $R(G)$ function in a way that is as compatible with the likelihood method as possible. Then we discuss how the CM algorithm is applied to mixture models. Finally, we compare the CM algorithm with the EM algorithm to show the advantages of the CM algorithm.

2 Semantic Channel, Semantic Information Measure, and the $R(G)$ Function

2.1 From the Shannon Channel to the Semantic Channel

First, we introduce the Shannon channel.

Let X be a discrete random variable representing a fact with alphabet $A = \{x_1, x_2, \ldots, x_m\}$, and let Y be a discrete random variable representing a message with alphabet $B = \{y_1, y_2, \ldots, y_n\}$. A Shannon channel is composed of a group of transition probability functions [6]: $P(y_j|X)$, $j = 1, 2, \ldots, n$.

In terms of hypothesis-testing, X is a sample point and Y is a hypothesis or a model label. We need a sample sequence or sampling distribution $P(X|.)$ to test a hypothesis to see how accurate it is.

Let Θ be a random variable for a predictive model, and let θ_j be a value taken by Θ when $Y = y_j$. The semantic meaning of a predicate $y_j(X)$ is defined by θ_j or its (fuzzy) truth function $T(\theta_j|X) \in [0,1]$. Because $T(\theta_j|X)$ is constructed with some parameters, we may also treat θ_j as a set of model parameters. We can also state that $T(\theta_j|X)$ is defined by a normalized likelihood, i.e., $T(\theta_j|X) = k\, P(\theta_j|X)/P(\theta_j) = k\, P(X|\theta_j)/P(X)$, where k is a coefficient that makes the maximum of $T(\theta_j|X)$ be 1. The θ_j can also be regarded as a fuzzy set, and $T(\theta_j|X)$ can be considered as a membership function of a fuzzy set proposed by Zadeh [10].

In contrast to the popular likelihood method, the above method uses sub-models $\theta_1, \theta_2, \ldots, \theta_n$ instead of one model θ or Θ. The $P(X|\theta_j)$ is equivalent to $P(X|y_j, \theta)$ in the popular likelihood method. A sample used to test y_j is also a sub-sample or a conditional sample. These changes will make the new method more flexible and more compatible with the Shannon information theory.

[1] Excel files demonstrating iterative process for tests, estimations, and mixture models can be download from http://survivor99.com/lcg/CM-iteration.zip.

A semantic channel is composed of a group of truth value functions or membership functions: $T(\theta_j|X)$, $j = 1, 2, \ldots, n$.

Similar to $P(y_j|X)$, $T(\theta_j|X)$ can also be used for Bayesian prediction to produce likelihood function [4]:

$$P(X|\theta_j) = P(X)T(\theta_j|X)/T(\theta_j), \quad T(\theta_j) = \sum_i P(x_i)T(\theta_j|x_i) \tag{1}$$

where $T(\theta_j)$ is called the logical probability of y_j. The author now know that this formula was proposed by Thomas as early as 1981 [11]. We call this prediction the semantic Bayesian prediction. If $T(\theta_j|X) \propto P(y_j|X)$, then the semantic Bayesian prediction is equivalent to the Bayesian prediction.

2.2 Semantic Information Measure and the Optimization of the Semantic Channel

The semantic information conveyed by y_j about x_i is defined by normalized likelihood as [3]:

$$I(x_i;\theta_j) = \log\frac{P(x_i|\theta_j)}{P(x_i)} = \log\frac{T(\theta_j|x_i)}{T(\theta_j)} \tag{2}$$

where the semantic Bayesian inference is used; it is assumed that prior likelihood function $P(X|\Theta)$ is equal to prior probability distribution $P(X)$.

After averaging $I(x_i;\theta_j)$, we obtain semantic (or generalized) KL information:

$$I(X;\theta_j) = \sum_i P(x_i|y_j)\log\frac{P(x_i|\theta_j)}{P(x_i)} = \sum_i P(x_i|y_j)\log\frac{T(\theta_j|x_i)}{T(\theta_j)} \tag{3}$$

The statistical probability $P(x_i|y_j)$, $i = 1, 2, \ldots$, on the left of "log" above, represents a sampling distribution to test the hypothesis y_j or model θ_j. Assume we choose y_j according to observed condition $Z \epsilon C$. If $y_j = f(Z|Z \epsilon C_j)$, where C_j is a cub-set of C, then $P(X|y_j) = P(X|C_j)$.

After averaging $I(X;\theta_j)$, we obtain semantic (or generalized) mutual information:

$$\begin{aligned} I(X;\Theta) &= \sum_j P(y_j)\sum_i P(x_i|y_j)\log\frac{P(x_i|\theta_j)}{P(x_i)} \\ &= \sum_j\sum_i P(x_i)P(y_j|x_i)\log\frac{T(\theta_j|x_i)}{T(\theta_j)} = H(X) - H(X|\Theta) \\ H(X|\Theta) &= -\sum_j\sum_i P(x_i,y_j)\log P(x_i|\theta_j) \end{aligned} \tag{4}$$

where $H(X)$ is the Shannon entropy of X, $H(X|\Theta)$ is the generalized posterior entropy of X. Each of them has coding meaning [4, 5].

Optimizing a semantic Channel is equivalent to optimizing a predictive model Θ. For given $y_j = f(Z|Z\epsilon C_j)$, optimizing θ_j is equivalent to optimizing $T(\theta_j|X)$ by

$$T^*(\theta_j|X) = \arg\max_{T(\theta_j|X)} I(X;\theta_j) \tag{5}$$

It is easy to prove that when $P(X|\theta_j) = P(X|y_j)$, or

$$\frac{T(\theta_j|X)}{T(\theta_j)} = \frac{P(y_j|X)}{P(y_j)}, \quad \text{or } T(h_j|E) \propto P(h_j|E) \tag{6}$$

$I(X;\theta_j)$ reaches the maximum. Set the maximum of $T(\theta_j|X)$ to 1. Then we can obtain

$$T^*(\theta_j|X) = P(y_j|X)/P(y_j|x_j^*) = [P(X|y_j)/P(X)]/[P\left(x_j^*|y_j\right)/P\left(x_j^*\right)] \tag{7}$$

In this equation, x_j^* makes $P(x_j^*|y_j)/P(x_j^*)$ be the maximum of $P(X|y_j)/P(X)$.

2.3 Relationship Between Semantic Mutual Information and Likelihood

Assume that the size of the sample used to test y_j is N_j; the sample points come from independent and identically distributed random variables. Among these points, the number of x_i is N_{ij}. Assume that N_j is infinite, $P(X|y_j) = N_{ij}/N_j$. Hence, there is log normalized likelihood:

$$\log \prod_i \left[\frac{P(x_i|\theta_j)}{P(x_i)}\right]^{N_{ji}} = N_j \sum_i P(x_i|y_j) \log \frac{P(x_i|\theta_j)}{P(x_i)} = N_j I(X;\theta_j) \tag{8}$$

After averaging the above likelihood for different y_j, $j = 1, 2, \ldots, n$, we have the average log normalized likelihood:

$$\begin{aligned} \sum_j \frac{N_j}{N} \log \prod_i \left[\frac{P(x_i|\theta_j)}{P(x_i)}\right]^{N_{ji}} &= \sum_j P(y_j) \sum_i P(x_i|y_j) \log \frac{P(x_i|\theta_j)}{P(x_i)} \\ &= I(X;\Theta) = H(X) - H(X|\Theta) \end{aligned} \tag{9}$$

where $N = N_1 + N_2 + \cdots + N_n$. It shows that the ML criterion is equivalent to the minimum generalized posterior entropy criterion and the Maximum Semantic Information (MSI) criterion. When $P(X|\theta_j) = P(X|y_j)$ (for all j), the semantic mutual information $I(X;\Theta)$ is equal to the Shannon mutual information $I(X;Y)$, which is the special case of $I(X;\Theta)$.

2.4 The Matching Function $R(G)$ of R and G

The $R(G)$ function is an extension of the rate distortion function $R(D)$ [7]. In the $R(D)$ function, R is the information rate, D is the upper limit of the distortion. The $R(D)$ function means that for given D, $R = R(D)$ is the minimum of the Shannon mutual information $I(X;Y)$.

Let distortion function d_{ij} be replaced with $I_{ij} = I(x_i, y_j) = \log[T(\theta_j|x_i)/T(\theta_j)] = \log[P(x_i|\theta_j)/P(x_i)]$, and let G be the lower limit of the semantic mutual information $I(X; \Theta)$. The information rate for given G and $P(X)$ is defined as

$$R(G) = \min_{P(Y|X):I(E;\Theta)\geq G} I(X;Y) \tag{10}$$

Following the derivation of $R(D)$ [12], we can obtain [3]

$$\begin{aligned} G(s) &= \sum_i \sum_j I_{ij} P(x_i) P(y_j) 2^{sI_{ij}})/\lambda_i = \sum_i \sum_j I_{ij} P(x_i) P(y_j) m_{ij}^s/\lambda_i \\ R(s) &= sG(s) - \sum_i P(x_i) \log \lambda_i \end{aligned} \tag{11}$$

where $m_{ij} = T(\theta_j|x_i)/T(\theta_j) = P(x_i|\theta_j)/P(x_i)$ is the normalized likelihood; $\lambda_i = \sum_j P(y_j) m_{ij}^s$. We may also use $m_{ij} = P(x_i|\ \theta_j)$, which results in the same m_{ij}^s/λ_i. The shape of an $R(G)$ function is a bowl-like curve as shown in Fig. 1.

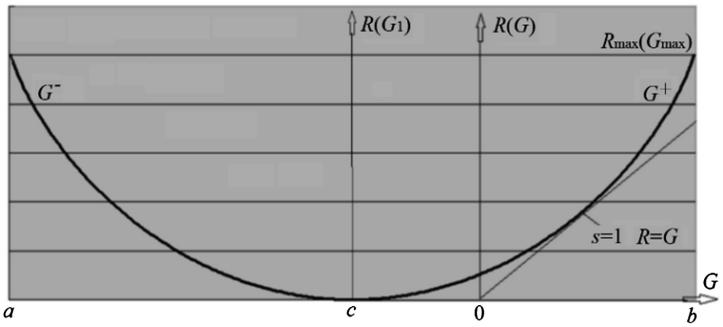

Fig. 1. The $R(G)$ function of a binary source. When the slope $s = 1$, $G = R$, and information efficiency G/R reaches its maximum 1.

The R(G) function is different from the $R(D)$ function. For a given R, we have the maximum value G^+ and the minimum value G^-, which is negative and means that to bring a certain information loss $|G|$ to enemies, we also need certain objective information R.

In the rate distortion theory, $\mathrm{d}R/\mathrm{d}D = s$ ($s \leq 0$). It is easy to prove that there is also $\mathrm{d}R/\mathrm{d}G = s$, where s may be less or greater than 0. The increase of s will raise the model's prediction precision. If s changes from positive s_1 to $-s_1$, then $R(-s_1) = R(s_1)$ and G changes from G^+ to G^- (see Fig. 1).

When $s = 1$, $\lambda_i = 1$, and $R = G$, which means that the semantic channel matches the Shannon channel and the semantic mutual information is equal to the Shannon mutual information. When $s = 0$, $R = 0$ and $G < 0$. In Fig. 1, $c = G(s = 0)$.

3 The CM Algorithm for Mixture Models

3.1 Explaining the Iterative Process by the $R(G)$ Function

Assume a sampling distribution $P(X)$ is produced by the conditional probability $P^*(X|Y)$ being some function such as Gaussian distribution. We only know that the number of the mixture components is n, without knowing $P(Y)$. We need to solve $P(Y)$ and model (or parameters) Θ, so that the predicted probability distribution of X, denoted by $Q(X)$, is as close to the sampling distribution $P(X)$ as possible, i.e. the relative entropy or Kullback-Leibler divergence $H(Q||P)$ is as small as possible. The Fig. 2 shows the convergent processes of two examples.

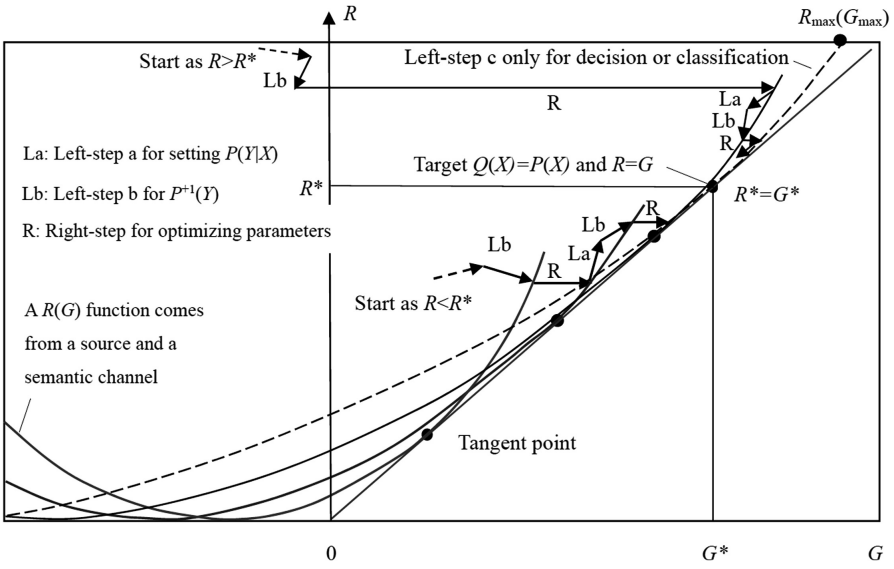

Fig. 2. Illustrating the CM algorithm for mixture models. There are two iterative examples. One is for $R > R^*$ and another is for $R < R^*$. The Left-step a and Left-step b make R close to R^*; whereas the Right-step increases G so that (G, R) approaches line $R = G$.

We use $P^*(Y)$ and $P^*(X|Y)$ to denote the $P(Y)$ and $P(X|Y)$ that are used to produce the sampling distribution $P(X)$, and use $P^*(Y|X)$ and $R^* = I^*(X;Y)$ to denote the corresponding Shannon channel and Shannon mutual information. When $Q(X) = P(X)$, there should be $P(X|\Theta) = P^*(X|Y)$, and $G^* = R^*$.

For mixture models, when we let the Shannon channel match the semantic channel (in Left-steps), we do not maximize $I(X;\Theta)$, but seek a $P(X|\Theta)$ that accords with $P^*(X|Y)$ as possible (Left-step a in Fig. 2 is for this purpose), and a $P(Y)$ that accords with $P^*(Y)$ as possible (Left-step b in Fig. 2 is for this purpose). That means we seek a R that is as close to R^* as possible. Meanwhile, $I(X;\Theta)$ may decrease. However, in popular

EM algorithms, the objective function, such as $P(X^N, Y|\Theta)$, is required to keep increasing without decreasing in both steps.

With CM algorithm, only after the optimal model is obtained, if we need to choose Y according to X (for decision or classification), we may seek the Shannon channel $P(Y|X)$ that conveys the MMI $R_{\max}(G_{\max})$ (see Left-step c in Fig. 2).

Assume that $P(X)$ is produced by $P^*(X|Y)$ with the Gaussian distribution. Then the likelihood functions are

$$P(X|\theta_j) = k_j \exp\left[-(X - c_j)^2/(2d_j)^2\right], j = 1, 2, \ldots, n$$

If $n = 2$, then parameters are c_1, c_2, d_1, d_2. In the beginning of the iteration, we may set $P(Y) = 1/n$. We begin iterating from Left-step a.

Left-step a: Construct Shannon channel by

$$\begin{aligned} &P(y_j|X) = P(y_j)P(X|\theta_j)/Q(X) \\ &Q(X) = \sum_j P(y_j)|(X|\theta_j) \qquad\qquad ,j = 1, 2, \ldots, n \end{aligned} \tag{12}$$

This formula has already been used in the EM algorithm [1]. It was also used in the derivation process of the $R(D)$ function [12]. Hence the semantic mutual information is

$$G = I(X; \Theta) = \sum_i \sum_j P(x_i) \frac{P(x_i|\theta_j)}{Q(x_i)} P(y_j) \log \frac{P(x_i|\theta_j)}{P(x_i)} \tag{13}$$

Left-step b: Use the following equation to obtain a new $P(Y)$ repeatedly until the iteration converges.

$$P(y_j) \Leftarrow \sum_i P(x_i)P(y_j|x_i) = \sum_i P(x_i) \frac{P(x_i|\theta_j)}{\sum_k P(y_k)P(x_i|\theta_k)} P(y_j), j = 1, 2, \ldots, n \tag{14}$$

The convergent $P(Y)$ is denoted by $P^{+1}(Y)$. This is because $P(Y|X)$ from Eq. (12) is an incompetent Shannon channel so that $\sum_i P(x_i)P(y_j|x_i) \neq P(y_j)$. The above iteration makes $P^{+1}(Y)$ match $P(X)$ and $P(X|\Theta)$ better. This iteration has been used by some authors, such as in [13].

When $n = 2$, we should avoid choosing c_1 and c_2 so that both are larger or less than the mean of X; otherwise $P(y_1)$ or $P(y_2)$ will be 0, and cannot be larger than 0 later.

If $H(Q||P)$ is less than a small number, such as 0.001 bit, then end the iteration; otherwise go to Right-step.

Right-step: Optimize the parameters in the likelihood function $P(X|\Theta)$ on the right of the log in Eq. (13) to maximize $I(X;\Theta)$. Then go to Left-step a.

3.2 Using Two Examples to Show the Iterative Processes

3.2.1 Example 1 for $R < R^*$

In Table 1, there are real parameters that produce the sample distribution $P(X)$ and guessed parameters that are used to produce $Q(X)$. The convergence process from the starting (G, R) to (G^*, R^*) is shown by the iterative locus as $R < R^*$ in Fig. 2. The convergence speed and changes of R and G are shown in Fig. 3. The iterative results are shown in Table 1.

Table 1. Real and guessed model parameters and iterative results of Example 1 ($R < R^*$)

Y	Real parameters			Start parameters $H(Q\|\|P) = 0.410$ bit			Parameters after 5 iterations $H(Q\|\|P) = 0.00088$ bit		
	c	d	$P^*(Y)$	c	d	$P(Y)$	c	d	$P(Y)$
y_1	35	8	0.7	30	15	0.5	35.4	8.3	0.72
y_2	65	12	0.3	70	15	0.5	65.2	11.4	0.28

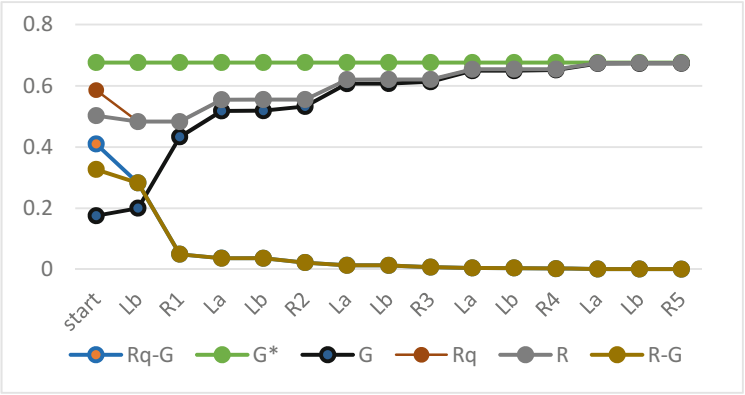

Fig. 3. The iterative process as $R < R^*$. Rq is R_Q in Eq. (15). $H(Q||P) = R_Q - G$ decreases in all steps. G is monotonically increasing. R is also monotonically increasing except in the first Left-step b. G and R gradually approach $G^* = R^*$ so that $H(Q||P) = R_Q - G$ is close to 0.

Analyses: In this iterative process, there are always $R < R^*$ and $G < G^*$. After each step, R and G increase a little bit so that G approaches G^* gradually. This process seams to tell us that each of Right-step, Left-step a, and Left-step b can increase G; and hence maximizing G can minimize $H(Q||P)$, which is our goal. Yet, it is wrong. The Left a and Left b do not necessarily increase G. There are many counterexamples. Fortunately, iterations for these counterexamples can still converge. Let us see Example 2 as a counterexample.

3.2.2 Example 2 for $R > R^*$

Table 2 shows the parameters and iterative results for $R > R^*$. The iterative process is shown in Fig. 4.

Table 2. Real and guessed model parameters and iterative results for Example 2 ($R > R^*$)

Y	Real parameters			Start parameters $H(Q\|\|P) = 0.680$ bit			Parameters after 5 iterations $H(Q\|\|P) = 0.00092$ bit		
	c	d	$P^*(Y)$	c	d	$P(Y)$	c	d	$P(Y)$
y_1	35	8	0.1	30	8	0.5	38	9.3	0.134
y_2	65	12	0.9	70	8	0.5	65.8	11.5	0.886

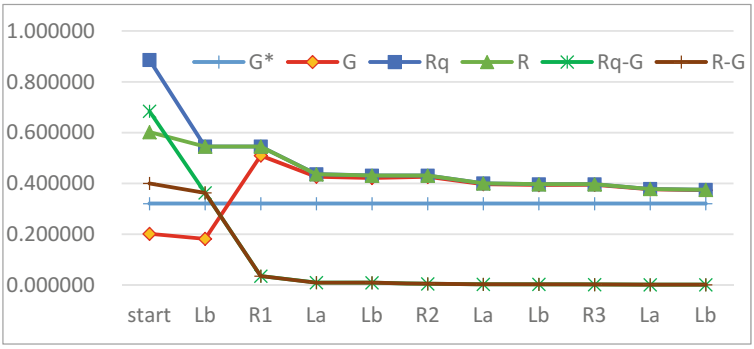

Fig. 4. The iterative process as $R > R^*$. Rq is R_Q in Eq. (15). $H(Q||P) = R_Q - G$ decreases in all steps. R is monotonically decreasing. G increases more or less in all Right-steps and decreases in all Left-steps. G and R gradually approach $G^* = R^*$ so that $H(Q||P) = R_Q - G$ is close to 0.

Analyses: G is not monotonically increasing nor monotonically decreasing. It increases in all Right steps and decreases in all Left steps. This example is a challenge to all authors who prove that the standard EM algorithm or a variant EM algorithm converges. If G is not monotonically increasing, it must be difficult or impossible to prove that $\log P(X^N, Y|\Theta)$ or other likelihood is monotonically increasing or no-decreasing in all steps. For example, in Example 2, $Q^* = -NH^*(X, Y) = -6.031\ N$. After the first optimization of parameters, $Q = -6.011\ N > Q^*$. If we continuously maximize Q, Q cannot approach less Q^*.

We also use some other true models $P^*(X|Y)$ and $P^*(Y)$ to test the CM algorithm. In most cases, the number of iterations is close to 5. In rare cases where R and G are much bigger than G^*, such as $R \approx G > 2G^*$, the iterative convergence is slow. In these cases where $\log P(X^N, Y|\Theta)$ is also much bigger than $\log P^*(X^N, Y)$, the EM algorithm confronts similar problem. Because of these cases, the convergence proof of the EM algorithm is challenged.

3.3 The Convergence Proof of the CM Algorithm

Proof. To prove the CM algorithm converges, we need to prove that $H(Q||P)$ is decreasing or no-increasing in every step.

Consider the Right-step. Assume that the Shannon mutual information conveyed by Y about $Q(X)$ is R_Q, and that about $P(X)$ is R. Then we have

$$R_Q = I_Q(X;Y) = \sum_i \sum_j P(x_i)\frac{P(x_i|\theta_j)}{Q(x_i)}P(y_j)\log\frac{P(x_i|\theta_j)}{Q(x_i)} \tag{15}$$

$$\begin{aligned} R = I(X;Y) &= \sum_i \sum_j P(x_i)\frac{P(x_i|\theta_j)}{Q(x_i)}P(y_j)\log\frac{P(y_j|x_i)}{P^{+1}(y_j)} \\ &= R_Q - H(Y||Y^{+1}) \\ H(Y||Y^{+1}) &= \sum_j P^{+1}(y_j)\log[P^{+1}(y_j)/P(y_j)] \end{aligned} \tag{16}$$

According to Eqs. (13) and (15), we have

$$H(Q||P) = R_Q - G = R + H(Y||Y^{+1}) - G \tag{17}$$

Because of this equation, we do not need Jensen's inequality that the EM algorithm needs.

In Right-steps, the Shannon channel and R_Q does not change, G is maximized. Therefore $H(Q||P)$ is decreasing and its decrement is equal to the increment of G.

Consider Left-step a. After this step, $Q(X)$ becomes $Q^{+1}(X) = \sum_j P(y_j)P^{+1}(X|\theta_j)$. Since $Q^{+1}(X)$ is produced by a better likelihood function and the same $P(Y)$, $Q^{+1}(X)$ should be closer to $P(X)$ than $Q(X)$, i.e. $H(Q^{+1}\|P) < H(Q||P)$ (More strict mathematical proof for this conclusion is needed).

Consider Left-step b. The iteration for $P^{+1}(Y)$ moves (G, R) to the $R(G)$ function cure ascertained by $P(X)$ and $P(X|\theta_j)$ (for all j) that form a semantic channel. This conclusion can be obtained from the derivation processes of $R(D)$ function [12] and $R(G)$ function [3]. A similar iteration is used for $P(Y|X)$ and $P(Y)$ in deriving the $R(D)$ function. Because $R(G)$ is the minimum R for given G, $H(Q||P) = R_Q - G = R - G$ becomes less.

Because $H(Q||P)$ becomes less after every step, the iteration converges. **Q.E.D.**

3.4 The Decision Function with the ML Criterion

After we obtain optimized $P(X|\Theta)$, we need to select Y (to make decision or classification) according to X. The parameter s in $R(G)$ function (see Eq. (11)) reminds us that we may use the following Shannon channel

$$\begin{aligned} P(y_j|X) &= P(y_j)[P(X|\theta_j)]^s/Q(X) \\ Q(X) &= \sum_j P(y_j)[P(X|\theta_j)]^s \end{aligned} \quad ,j = 1, 2, \ldots, n \tag{18}$$

which are fuzzy decision functions. When $s \to +\infty$, the fuzzy decision will become crisp decision. Different from Maximum A prior (MAP) estimation, the above decision function still persists in the ML criterion or MSI criterion. The Left-step c in Fig. 2 shows that (G, R) moves to ($G_{\max}$, $R_{\max}$) with s increasing.

3.5 Comparing the CM Algorithm and the EM Algorithm

In the EM algorithm [1, 14], the likelihood of a mixture model is expressed as $\log P(X^N|\Theta) > L = Q - H$. If we move $P(Y)$ or $P(Y|\Theta)$ from Q into H, then Q will become $-NH(X|\Theta)$ and H becomes $-NR_Q$. If we add $NH(X)$ to both sides of the inequality, we will have $H(Q||P) \le R_Q - G$, which is similar to Eq. (17). It is easy to prove

$$Q = NG - NP(X) - NH(Y) \tag{19}$$

where $H(Y) = -\sum_j P^{+1}(y_j)\log P(y_j)$ is a generalized entropy. We may think the M-step merges the Left-step b and the Right-step of the CM algorithm into one step. In brief,

$$\begin{aligned} &\text{The E-step of EM} = \text{the Left-step a of CM} \\ &\text{The M-step of EM} \approx \text{the Left-step b} + \text{the Right-step of CM} \end{aligned}$$

In the EM algorithm, if we first optimize $P(Y)$ (not for maximum Q) and then optimize $P(X|Y, \Theta)$, then the M-step will be equivalent to the CM algorithm.

There are also other improved EM algorithms [13, 15–17] with some advantages. However, no one of these algorithms facilitates that R converges to R^*, and $R - G$ converges to 0 as the CM algorithm.

The convergence reason of the CM algorithm is seemly clearer than the EM algorithm (see the analyses in Example 2 for $R > R^*$). According to [7, 15–17], the CM algorithm is faster at least in most cases than the various EM algorithms.

The CM algorithm can also be used to achieve maximum mutual information and maximum likelihood of tests and estimations. There are more detailed discussions about the CM algorithm[2].

4 Conclusions

Lu's semantic information measure can combine the Shannon information theory and likelihood method so that the semantic mutual information is the average log normalized likelihood. By letting the semantic channel and Shannon channel mutually match and iterate, we can achieve the mixture model with minimum relative entropy. The iterative convergence can be intuitively explained and proved by the $R(G)$ function.

[2] https://arxiv.org/abs/1706.07918.

Two iterative examples and mathematical analyses show that the CM algorithm has higher efficiency at least in most cases and clearer convergence reasons than the popular EM algorithm.

Acknowledgment. The author thanks Professor Peizhuang Wang for his long term supports and encouragements.

References

1. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B **39**, 1–38 (1977)
2. Kok, M., Dahlin, J.B., Schon, T.B., Wills, A: Newton-based maximum likelihood estimation in nonlinear state space models. In: IFAC-PapersOnLine 48, pp. 398–403 (2015)
3. Lu, C.: A Generalized Information Theory. China Science and Technology University Press, Hefei (1993). (in Chinese)
4. Lu, C.: Coding meanings of generalized entropy and generalized mutual information. J. China Inst. Commun. **15**, 37–44 (1994). (in Chinese)
5. Lu, C.: A generalization of Shannon's information theory. Int. J. Gen. Syst. **28**, 453–490 (1999)
6. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–429, 623–656 (1948)
7. Shannon, C.E.: Coding theorems for a discrete source with a fidelity criterion. IRE Nat. Conv. Rec. **4**, 142–163 (1959)
8. Kullback, S., Leibler, R.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
9. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**, 716–723 (1974)
10. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
11. Thomas, S.F.: Possibilistic uncertainty and statistical inference. ORSA/TIMS Meeting, Houston, Texas (1981)
12. Zhou, J.: Fundamentals of Information Theory. People's Posts & Telecom Press, Beijing (1983). (in Chinese)
13. Byrne, C.L.: The EM algorithm theory applications and related methods. https://www.researchgate.net/profile/Charles_Byrne
14. Wu, C.F.J.: On the convergence properties of the EM algorithm. Ann. Stat. **11**, 95–103 (1983)
15. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 355–368. MIT Press, Cambridge (1999)
16. Huang, W.H., Chen, Y.G.: The multiset EM algorithm. Stat. Probab. Lett. **126**, 41–48 (2017)
17. Springer, T., Urban, K.: Comparison of the EM algorithm and alternatives. Numer. Algorithms **67**, 335–364 (2014)