information

MDPI

Article

Semantic Information G Theory and Logical Bayesian Inference for Machine Learning

Chenguang Lu

Institute of Intelligence Engineering and Mathematics, Liaoning Technical University, Fuxin 123000, China; survival99@gmail.com

Received: 20 June 2019; Accepted: 13 August 2019; Published: 16 August 2019

Abstract: An important problem in machine learning is that, when using more than two labels, it is very difficult to construct and optimize a group of learning functions that are still useful when the prior distribution of instances is changed. To resolve this problem, semantic information G theory, Logical Bayesian Inference (LBI), and a group of Channel Matching (CM) algorithms are combined to form a systematic solution. A semantic channel in G theory consists of a group of truth functions or membership functions. In comparison with the likelihood functions, Bayesian posteriors, and Logistic functions that are typically used in popular methods, membership functions are more convenient to use, providing learning functions that do not suffer the above problem. In Logical Bayesian Inference (LBI), every label is independently learned. For multilabel learning, we can directly obtain a group of optimized membership functions from a large enough sample with labels, without preparing different samples for different labels. Furthermore, a group of Channel Matching (CM) algorithms are developed for machine learning. For the Maximum Mutual Information (MMI) classification of three classes with Gaussian distributions in a two-dimensional feature space, only 2–3 iterations are required for the mutual information between three classes and three labels to surpass 99% of the MMI for most initial partitions For mixture models, the Expectation-Maximization (EM) algorithm is improved to form the CM-EM algorithm, which can outperform the EM algorithm when the mixture ratios are imbalanced, or when local convergence exists. The CM iteration algorithm needs to combine with neural networks for MMI classification in high-dimensional feature spaces. LBI needs further investigation for the unification of statistics and logic.

Keywords: semantic information theory; Bayesian inference; machine learning; Multilabel learning; maximum mutual information classifications; mixture models; confirmation measure; truth function

1. Introduction

Machine learning is based on learning functions and classifiers. In 1922, Fisher [1] proposed the Likelihood Inference (LI), which uses likelihood functions as learning functions and it uses the Maximum Likelihood (ML) criterion to optimize the learning functions and classifiers (see Appendix A for all abbreviations in this paper). However, when the prior distribution, $P(x)$ (where x is an instance), is changed, the optimized likelihood function will be invalid. As LI cannot make use of prior knowledge, Bayesians proposed Bayesian Inference (BI) during the 1950s [2,3], which uses Bayesian posteriors as learning functions. However, in many cases, we only have prior knowledge of instances, instead of labels or model parameters and, hence, BI is still not good in such cases. A pair of Logistic (or Sigmoid) functions are often used as the learning functions for binary classifications. With a Logistic function and Bayes' Theorem, we can make use of a new prior $P(x)$ to make new

probability predictions for the ML classifier. However, when the number of labels is greater than two, we cannot find proper learning functions that are similar to Logistic functions for multilabel learning. We call the above problem the "Multilabel-Learning-for-New-$P(x)$ Problem".

Machine learning is used to acquire and convey information, and so the information criterion that is used should be a good criterion. In 1974, Akaike [4] proved that the ML criterion is equal to the minimum Kullback–Leibler (KL) divergence criterion, where the KL divergence [5] is also called "KL information". Since then, information criteria, especially information criteria that are compatible with the likelihood criterion, have attracted the attention of researchers [6]. However, KL divergence decreases as the likelihood increases and, hence, the Least KL divergence is not ideal as an information criterion. Can we use Shannon's mutual information or another information measure for the information criterion?

In 1948, Shannon [7] initiated classical information theory. In 1949, Weaver [8] proposed three levels of communication that are relevant to the technical problem that was resolved by Shannon, a semantic problem that relates to meaning and truth, and an effectiveness problem concerning information values. In 1952, Carnap and Bar-Hillel [9] proposed an outline of semantic information theory. Multiple different semantic information theories currently exist [10–13], as well as fuzzy information theories [14–16] and generalized information theories [17,18] that are related to semantic information theories. Recently, some researchers have used the Shannon mutual information measure with parameters to optimize neural networks [19,20].

However, Shannon's Mutual Information (SHMI) formula has not yet been used to optimize a learning functions with parameters by the use of a sampling distribution. Therefore, the author introduced a learning function into the SHMI formula and developed semantic information G theory, or G theory. The author mainly developed this theory over the past three decades [21–26]. The G theory uses the membership functions of fuzzy sets, as proposed by Zadeh [27], as learning functions and treats a membership function as the truth function of a hypothesis. The truth function can represent the semantic meaning of a hypothesis, according to Tarski's truth theory [28] and Davidson's truth-conditional semantics [29].

"G theory" is used because, in this theory, Semantic Mutual Information (SMI) is a natural generalization of SHMI ("G" denoting "generalization"), so that SHMI is the upper limit of SMI. G also denotes SMI as D denotes average distortion in Shannon's information rate distortion theory [30]. Replacing D with G, the author reformed the rate-distortion function $R(D)$ into the rate-verisimilitude function $R(G)$ [24,25], not only for data compression, but also for machine learning.

G theory has two headstreams: Shannon's information theory and Popper's hypothesis-testing theory (see [31], p. 96 and 269; and [32], p. 294), which emphasizes that a hypothesis with a smaller logical probability can convey more information if it can survive empirical tests and, hence, is more preferable.

Carnap and Bar-Hillel [9] used logical probability to define the semantic information measure, which contains Popper's partial thought. However, this measure does not deal with whether the hypothesis can survive empirical tests. Therefore, G theory introduces the membership function into the semantic information measure.

Cross-entropy has become a popular tool in machine learning [33]. G theory uses not only cross-entropy, but also mutual cross-entropy [22,25]. The SMI in G theory is a mutual cross-entropy.

To resolve the "Multilabel-Learning-for-New-$P(x)$" problem, the author investigated a new inference method: Logical Bayesian Inference (LBI). The Bayesians include subjective Bayesians and logical Bayesians. BI was developed by subjective Bayesians, who use subjective probability for statistical inference. Logical Bayesians, such as Keynes and Carnap [34], use logical probability, including the truth function, for inductive logic. BI uses the Bayesian posterior as the inferential tool. Logical Bayesian Inference uses the truth function (e.g., the fuzzy truth function) instead of the Bayesian posterior as the inferential tool. In LBI, both statistical and logical probabilities are simultaneously used. BI fits cases with a given prior distribution of a predictive model θ, whereas LBI fits cases with a given prior distribution of an instance X.

Besides Shannon's information theory and Poppers' hypothesis-testing theory, G theory and LBI should inherit, absorb, or be compatible with:

- Fisher's likelihood method for hypothesis-testing [1];
- Zadeh's fuzzy set theory [27,35] for semantic meanings and logical probabilities of hypotheses;
- Carnap and Bar-Hillel's semantic information formula with logical probability [9];
- Floridi's semantic concepts of information [11,36];
- Tarski's truth theory for the definition of truth and logical probability [28];
- Davidson's truth-conditional semantics [29];
- Kullback and Leibler's KL divergence [5];
- Akaike's proof [4] that the ML criterion is equal to the minimum KL divergence criterion;
- Theil's generalized KL formula [37];
- the Donsker–Varadhan representation as a generalized KL formula with Gibbs density [38];
- Wittgenstein's thought: meaning lies in uses (see [39], p.80);
- Bayes' Theorem [40], which can be extended to link likelihood functions and membership functions [41]; and,
- Logical Bayesian methods for inductive logic used by Carnap et al. [3,34]

Based on G theory and LBI, the author developed a group of algorithms, called Channel Matching (CM) algorithms [41–44], for machine learning. In the CM algorithms, the semantic channel and Shannon channel mutually match to achieve maximum information (for classification) or maximum information efficiency (*G*/*R*) (for mixture models).

These algorithms are used mainly for:

- making use of the prior knowledge of instances for probability predictions;
- multilabel learning, belonging to supervised learning;
- the Maximum Mutual Information (MMI) classifications of unseen instances, belonging to semi-supervised learning; and,
- mixture models, belonging to unsupervised learning.

Each of them is very difficult and not well resolved before.

This study aims to completely introduce G theory, LBI, and the CM algorithms, along with sufficient background knowledge and applications for readers to fully understand them, especially to understand how to use them to resolve the "Multilabel-Learning-for-New-$P(x)$" Problem.

Partial contents of this paper have been introduced in several short papers that were published in conference proceedings [41–44]. Some contents introduced before are improved in this paper, such as one-dimensional examples for MMI classification and mixture models, which are now two-dimensional examples, as well as the previous two formulae for the confirmation measure being consolidated into one formula.

According to the author's knowledge, nowhere in the literature has a semantic information measure been used to optimize the membership functions or truth functions with parameters by sampling distributions; no has the statistical probability and the logical probability of a hypothesis been distinguished and simultaneously used in the same formula; nor has the semantic channel with its mathematical representation been proposed.

2. Methods I: Background

2.1. From Shannon Information Theory to Semantic Information G Theory

2.1.1. From Shannon's Mutual Information to Semantic Mutual Information

Definition 1.

- *x: an instance or data point; X: a discrete random variable taking a value $x \in U = \{x_1, x_2, \ldots, x_m\}$.*
- *y: a hypothesis or label; Y: a discrete random variable taking a value $y \in V = \{y_1, y_2, \ldots, y_n\}$.*

- *$P(y_j|x)$ (with fixed y_j and variable x): a Transition Probability Function (TPF) (named as such by Shannon [7]).*

Shannon named $P(X)$ the source, $P(Y)$ the destination, and $P(Y|X)$ the channel. A Shannon channel is a transition probability matrix or a group of transition probability functions:

$$P(Y|X) \Leftrightarrow \begin{bmatrix} P(y_1|x_1) & P(y_1|x_2) & ... & P(y_1|x_m) \\ P(y_2|x_1) & P(y_2|x_2) & ... & P(y_2|x_m) \\ ... & ... & ... & ... \\ P(y_n|x_1) & P(y_n|x_2) & ... & P(y_n|x_m) \end{bmatrix} \Leftrightarrow \begin{bmatrix} P(y_j|x) \\ P(y_j|x) \\ ... \\ P(y_n|x) \end{bmatrix}, \quad (1)$$

where ⇔ indicates equivalence. Note that the TPF $P(y_j|x)$ is not normalized, unlike the conditional probability function, $P(y|x_i)$, in which y is variable and x_i is constant. We will discuss how the TPF can be used for the traditional Bayes prediction in Section 2.2.1.

The Shannon entropies of X and Y are

$$H(X) = -\sum_j P(x_i) \log P(x_i), \quad (2)$$

$$H(Y) = -\sum_j P(y_j) \log P(y_j). \quad (3)$$

The Shannon posterior entropies of X and Y are

$$H(X|Y) = -\sum_j \sum_i P(x_i, y_j) \log P(x_i|y_j), \quad (4)$$

$$H(Y|X) = -\sum_j \sum_i P(x_i, y_j) \log P(y_j|x_i). \quad (5)$$

The Shannon mutual information is

$$\begin{aligned} I(X;Y) &= -\sum_j \sum_i P(x_i, y_j) \log \frac{P(x_i|y_j)}{P(x_i)} = H(X) - H(X|Y) \\ &= -\sum_j \sum_i P(x_i, y_j) \log \frac{P(y_j|x_i)}{P(y_j)} = H(Y) - H(Y|X). \end{aligned} \quad (6)$$

If $Y = y_j$, the mutual information $I(X; Y)$ will become the Kullback–Leibler (KL) divergence:

$$I(X; y_j) = \sum_i P(x_i|y_j) \log \frac{P(x_i|y_j)}{P(x_i)} = \sum_i P(x_i|y_j) \log \frac{P(y_j|x_i)}{P(y_j)}. \quad (7)$$

Some researchers have used the following formula to measure the information between x_i and y_j:

$$I(x_i; y_j) = \log \frac{P(x_i|y_j)}{P(x_i)} = \log \frac{P(y_j|x_i)}{P(y_j)}. \quad (8)$$

As $I(x_i; y_j)$ may be negative, however Shannon did not use this formulation. Shannon explained that information is the reduced uncertainty or the saved average code word length. The author believes that the above formula is meaningful, because negative information indicates that a bad prediction may increase the uncertainty or the code word length.

As Shannon's information theory cannot measure semantic information, Carnap and Bar-Hillel proposed a semantic information formula

$$I(p) = \log[1/m_p]. \quad (9)$$

As $I(p)$ is not relative to whether the prediction is correct or not, this formula is not practical.

Zhong [12] made use of the fuzzy entropy of DeLuca and Termini [14] to define the semantic information measure

$$I(y_j) = \log 2 + [t_j \log t_j + (1 - t_j) \log(1 - t_j)], \quad (10)$$

where t_j is "the logical truth" of y_j. However, according to this formula, whenever t_j=1 or t_j=0, the information reaches its maximum of 1 bit. This result is not expected. Therefore, this formula is unreasonable. This problem is also found in other semantic or fuzzy information theories that use DeLuca and Termini's fuzzy entropy [14].

Floridi's semantic information formula [11,36] is a little complicated. It can ensure that the information that is conveyed by a tautology or a contradiction reaches its minimum 0. However, according to common sense, a wrong prediction or a lie is worse than a tautology. As to how the semantic information is related to the deviation and how the amount of semantic information of a correct prediction differs from that of a wrong prediction, we cannot obtain clear answers from his formula.

The author proposed an improved semantic information measure in 1990 [21] and developed G theory later.

According to Tarski's truth theory [28], $P(X \in \theta_j)$ is equivalent to $P(\text{"}X \in \theta\text{" is true}) = P(y_j \text{ is true})$. The truth function of y_j ascertains the semantic meaning of y_j, according to Davidson's truth condition semantics [29]. Following Tarski and Davidson, we define, as follows:

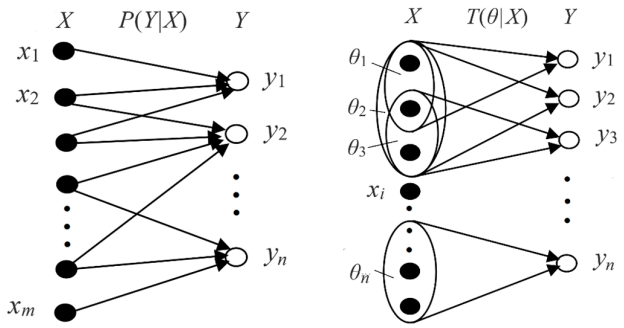
Definition 2.

- *θ_j is a fuzzy subset of U which is used to explain the semantic meaning of a predicate $y_j(X)$ = "$X \in \theta_j$". If θ_j is non-fuzzy, we may replace it with A_j. The θ_j is also treated as a model or a group of model parameters.*
- *A probability is defined with "=", such that $P(y_j) = P(Y = y_j)$, is a statistical probability; a probability is defined with "$\in$", such as $P(X \in \theta_j)$, is a logical probability. To distinguish $P(Y = y_j)$ and $P(X \in \theta_j)$, we define $T(\theta_j) = P(X \in \theta_j)$ as the logical probability of y_j.*
- *$T(\theta_j | x) = P(x \in \theta_j) = P(X \in \theta_j | X = x)$ is the conditional logical probability function of y_j; this is also called the (fuzzy) truth function of y_j or the membership function of θ_j.*

A group of TPFs $P(y_j | x)$, $j = 1,2,\ldots,n$, form a Shannon channel, whereas a group of membership functions $T(\theta_j | x)$, $j = 1,2\ldots n$, form a semantic channel:

$$T(\theta | X) \Leftrightarrow \begin{bmatrix} T(\theta_1 | x_1) & T(\theta_1 | x_2) & \ldots & T(\theta_1 | x_m) \\ T(\theta_2 | x_1) & T(\theta_2 | x_2) & \ldots & T(\theta_2 | x_m) \\ \ldots & \ldots & \ldots & \ldots \\ T(\theta_n | x_1) & T(\theta_n | x_2) & \ldots & T(\theta_n | x_m) \end{bmatrix} \Leftrightarrow \begin{bmatrix} T(\theta_1 | x) \\ T(\theta_2 | x) \\ \ldots \\ T(\theta_n | x) \end{bmatrix}. \quad (11)$$

Figure 1 illustrates the Shannon channel $P(Y | X)$ and the semantic channel $T(\theta | X)$.



(**a**) The Shannon channel (**b**) The semantic channel

Figure 1. The Shannon channel and the semantic channel. The semantic meaning of y_j is ascertained by the membership relation between x and θ_j. A fuzzy set θ_j may be overlapped or included by another.

The Shannon channel indicates the correlation between X and Y, whereas the semantic channel indicates the fuzzy denotations of a group of labels. The Shannon channel indicates the rule by which the observatory selects labels or forecasts for the weather forecasts between an observatory and its audience, whereas the semantic channel indicates the semantic meanings of these forecasts understood by the audience.

The expectation of the truth function is the logical probability:

$$T(\theta_j) = \sum_i P(x_i) T(\theta_j \mid x_i), \tag{12}$$

which was proposed earlier by Zadeh [35] as the probability of a fuzzy event. This logical probability is a little different from the m_p that was defined by Carnap and Bar-Hillel [9]. The latter only rests with the denotation of a hypothesis. For example, y_1 is a hypothesis (such as "X is infected by the Human Immunodeficiency Virus (HIV)") or a label (such as "HIV-infected"). Its logical probability $T(\theta_1)$ is very small for normal people, because HIV-infected people are rare. However, m_p is irrelative to $P(x)$; it may be 1/2.

Note that the statistical probability is normalized, whereas the logical probability is not, in general. When $\theta_0, \theta_1, \ldots, \theta_n$ form a cover of U, we have that $P(y_0) + P(y_1) + \ldots + P(y_n) = 1$ and $T(\theta_0) + T(\theta_1) + \ldots + T(\theta_n) \geq 1$.

For example, if U is a group of people of different ages with the subsets A_1 = {adults} = $\{x \mid x \geq 18\}$, A_0 = {juveniles} = $\{x \mid x < 18\}$, and A_2 = {young people} = $\{x \mid 15 \leq x \leq 35\}$. The three sets form a cover of U, and $T(A_0) + T(A_1) = 1$. If $T(A_2) = 0.3$; the sum of the three logical probabilities is 1.3 > 1. However, the sum of three statistical probabilities $P(y_0) + P(y_1) + P(y_2)$ must be less or equal to 1. If y_2 is correctly used, $P(y_2)$ will change from 0 to 0.3. If A_0, A_1, and A_2 become fuzzy sets, the conclusion is the same.

Consider the tautology "There will be rain or will not be rain tomorrow". Its logical probability is 1, whereas its statistical probability is close to 0 because it is rarely selected.

We can put $T(\theta_j \mid x)$ and $P(x)$ into Bayes' formula to obtain a likelihood function [21]:

$$P(x \mid \theta_j) = \frac{T(\theta_j \mid x) P(x)}{T(\theta_j)}, \quad T(\theta_j) = \sum_i T(\theta_j \mid x_i) P(x_i). \tag{13}$$

$P(x \mid \theta_j)$ can be called the semantic Bayes prediction or the semantic likelihood function. According to Dubois and Prade [45], Thomas [46] and others have proposed similar formulae.

Assume that the maximum of $T(\theta_j \mid x)$ is 1. From $P(x)$ and $P(x \mid \theta_j)$, we can obtain

$$T(\theta_j \mid x) = \frac{T(\theta_j) P(x \mid \theta_j)}{P(x)}, \; T(\theta_j) = 1 / \max[P(x \mid \theta_j) / P(x)]. \tag{14}$$

The author [41] proposed the third type of Bayes' theorem, which consists of the above two formulae. This theorem can convert the likelihood function and the membership function or the truth function from one to another when $P(x)$ is given. Equation (14) is compatible with Wang's fuzzy set falling shadow theory [41,47].

Figure 2 illustrates the relationship between $P(x|\theta_j)$ and $T(\theta_j|x)$ for a given $P(x)$, where x is an age, the label y_j = "Youth", and θ_j is a non-fuzzy set and, hence, becomes A_j.

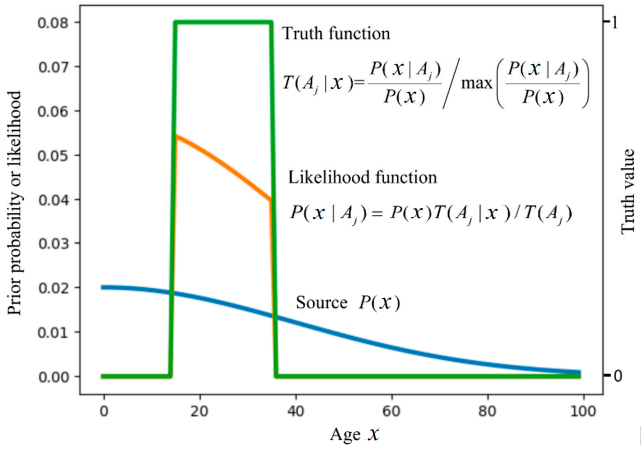


Figure 2. Relationship between $T(A_j|x)$ and $P(x|A_j)$ for given $P(x)$.

We use Global Positioning System (GPS) data as an example to demonstrate a semantic Bayes prediction.

Example 1. *A GPS device is used in a train, and hence P(x) is uniformly distributed on a line (see Figure 3). The GPS pointer has a deviation. Try to find the most probable position of the GPS device.*

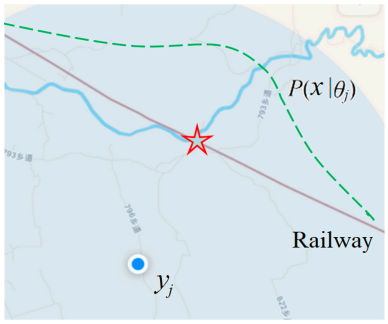


Figure 3. Illustrating the positioning of a GPS device with deviation. The round point is the pointed position with a deviation, and the position with the star is the most probable position.

The semantic meaning of the GPS pointer can be expressed by

$$T(\theta_j|x) = \exp[-(x - x_j)^2/(2\sigma^2)], \quad (15)$$

where x_j is the pointed position by y_j and σ is the Root Mean Square (RMS). For simplicity, we assume that x is one-dimensional.

According to Equation (13), we can predict that the position indicated by the star in Figure 3 is the most probable position. Most people would make the same prediction without using any mathematical formula. It seems that human brains must automatically use a similar method: making predictions according to the fuzzy denotation of y_j.

In semantic communication, we often see hypotheses or predictions, such as "the temperature is about 10 ˚C", "the time is about seven o'clock", or "the stock index will go up about 10% next

month". Each one of these may be represented by y_j = "X is about x_j". We can also express their truth functions by Equation (15).

The author defines the (amount of) semantic information that is conveyed by y_j about x_i with the log-normalized-likelihood:

$$I(x_i;\theta_j)=\log\frac{P(x_i\mid\theta_j)}{P(x_i)}=\log\frac{T(\theta_j\mid x_i)}{T(\theta_j)}. \tag{16}$$

Introducing Equation (15) into this formula, we have

$$I(x_i;\theta_j)=\log[1/T(\theta_j)]-(x_i-x_j)^2/(2\sigma^2), \tag{17}$$

by which we can explain that this information is equal to the Carnap–Bar-Hillel information minus the squared relative deviation. Figure 4 illustrates this formula.

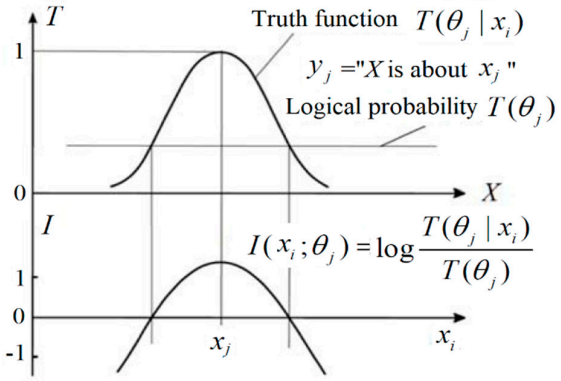


Figure 4. The semantic information conveyed by y_j about x_i.

Figure 4 indicates that, the smaller the logical probability, the more information there is; and, the larger the deviation is, the less information there is. Thus, a wrong hypothesis will convey negative information. These conclusions accord with Popper's thought (see [32], p. 294).

To average $I(x_i;\theta_j)$, we have generalized KL information

$$I(X;\theta_j)=\sum_i P(x_i\mid y_j)\log\frac{P(x_i\mid\theta_j)}{P(x_i)}=\sum_i P(x_i\mid y_j)\log\frac{T(\theta_j\mid x_i)}{T(\theta_j)}. \tag{18}$$

In Equation (18), $P(x_i\mid y_j)$ (i = 1,2…) is the sampling distribution, which may be unsmooth or discontinuous.

Theil proposed a generalized KL formula with three probability distributions [37]. However, in Equation (18), $T(\theta_j)$ is constant. If $T(\theta_j\mid x)$ is an exponential function with e as the base, and then Equation (18) will become the Donsker–Varadhan representation [19,38].

Akaike [4] proved that the Least KL divergence criterion is equivalent to the Maximum likelihood (ML) criterion. Following Akaike, we can prove that the Maximum Semantic Information (MSI) criterion (e.g., the maximum generalized KL information criterion) is also equivalent to the ML criterion.

Definition 3. ***D*** *is a sample with labels {(x(t), y(t))|t = 1 to N; x(t)∈U; y(t)∈V}, which includes n different sub-samples or conditional samples* $\mathbf{X}_j$, *j = 1,2,…,n. Every sub-sample includes data points x(1), x(2), …, x(*N_j*)∈U with label* y_j. *If* $\mathbf{X}_j$ *is large enough, we can obtain the distribution* $P(x\mid y_j)$ *from* $\mathbf{X}_j$. *If* y_j *in* $\mathbf{X}_j$ *is unknown, we replace* $\mathbf{X}_j$ *with* $\mathbf{X}$ *and* $P(x\mid y_j)$ *with* $P(x\mid.)$.

Assume that there are N_j data points in $\mathbf{X}_j$, where the N_{ji} data points are x_i. When N_j data points in $\mathbf{X}_j$ come from Independent and Identically Distributed (IID) random variables, we have the likelihood

$$\begin{aligned}\log P(\mathbf{X}_j \mid \theta_j) &= \log P(x(1), x(2), ..., x(N) \mid \theta_j) = \log \prod_i P(x_i \mid \theta_j)^{N_{ji}} \\ &= N_j \sum_i P(x_i \mid y_j) \log P(x_i \mid \theta_j) = -N_j H(X \mid \theta_j).\end{aligned} \tag{19}$$

As

$$I(X;\theta_j) = \sum_i P(x_i \mid y_j) \log \frac{P(x_i \mid \theta_j)}{P(x_i)} = -H(X \mid \theta_j) + \sum_i P(x_i \mid y_j) \log P(x_i), \tag{20}$$

$I(X; \theta_j)$ and $\log P(\mathbf{X}_j \mid \theta_j)$ reach their maxima at the same time that θ_j changes and, hence, the two criteria are equivalent. It is easy to prove that, when $P(x \mid \theta_j) = P(x \mid y_j)$, $I(X; \theta_j)$, and $\log P(\mathbf{X}_j \mid \theta_j)$ reach their maxima.

When the sample $\mathbf{X}_j$ is very large, letting $P(x \mid \theta_j) = P(x \mid y_j)$, we can obtain the optimized truth function:

$$T^*(\theta_j \mid x) = [P^*(x \mid \theta_j)/P(x)]/\max[P^*(x \mid \theta_j)/P(x)] = [P(x \mid y_j)/P(x)]/\max[P(x \mid y_j)/P(x)]. \tag{21}$$

We can also obtain

$$T^*(\theta_j \mid x) = P^*(\theta_j \mid x)/\max[P^*(\theta_j \mid x)] = P(y_j \mid x)/\max[P(y_j \mid x)]. \tag{22}$$

This formula clearly indicates how the semantic channel matches the Shannon channel, which indicates the use rule of Y. It is also compatible with Wittgenstein's thought: meaning lies in uses (see [39], p. 80).

To average $I(X; \theta_j)$ for different y, we use the Semantic Mutual Information (SMI) formula

$$I(X;\theta) = \sum_j P(y_j) \sum_i P(x_i \mid y_j) \log \frac{P(x_i \mid \theta_j)}{P(x_i)} = \sum_i \sum_j P(x_i) P(y_j \mid x_i) \log \frac{T(\theta_j \mid x_i)}{T(\theta_j)}. \tag{23}$$

If $P(x \mid \theta_j) = P(x \mid y_j)$ or $T(\theta_j \mid x) \propto P(y_j \mid x)$ for different y_j, the SMI will be equal to the Shannon Mutual Information (SHMI).

Introducing Equation (15) into the above formula, we have

$$\begin{aligned}I(X;\theta) &= H(\theta) - H(\theta \mid X) \\ &= -\sum_j P(y_j) \log T(\theta_j) - \sum_j \sum_i P(x_i, y_j)(x_i - x_j)^2 / (2\sigma_j^2).\end{aligned} \tag{24}$$

It is clear that the maximum SMI criterion is a special Regularized Least Squares (RLS) criterion [33]. $H(\theta)$ is the regularization term and $H(\theta \mid X)$ is the relative error term. However, $H(\theta)$ only penalizes the deviations without penalizing the means. The importance of this is that the maximum SMI criterion is also compatible with the ML criterion.

2.1.2. From the Rate-Distortion Function $R(D)$ to the Rate-Verisimilitude Function $R(G)$

The function $R(G)$ will be used to explain the convergence of the CM algorithms for the MMI classification and mixture models.

Shannon proposed the rate-distortion function $R(D)$ [30]. $R(G)$ [25] is a new version of $R(D)$. In $R(D)$, R is the information rate and D is the upper limit of average distortion. $R(D)$ means that, for given D, $R(D)$ is the minimum of SHMI $I(X; Y)$.

The rate distortion function with parameter s (see [48], p. 32) includes two formulae:

$$\begin{aligned}D(s) &= \sum_i \sum_j d_{ij} P(x_i) P(y_j) \exp(s d_{ij}) / \lambda_i, \\ R(s) &= sD(s) - \sum_i P(x_i) \ln \lambda_i,\end{aligned} \tag{25}$$

where $\lambda_i = \sum_j P(y_j)\exp(sd_{ij})$ is the partition function.

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

$$R(G) = \min_{P(Y|X):I(X;\theta)\geq G} I(X;Y) \tag{26}$$

Popper [32] proposed using verisimilitude, instead of correctness, to evaluate a hypothesis. Verisimilitude includes both correctness and precision. Hence, $I(x_i, \theta_j)$ can be a good measure for the verisimilitude of y_j reflecting x_i; therefore, we call $R(G)$ the rate-verisimilitude function.

Following the derivation of $R(D)$ ([48], p. 31), we obtain

$$\begin{aligned} G(s) &= \sum_i \sum_j I_{ij} P(x_i)P(y_j)2^{sI_{ij}}) / \lambda_i = \sum_i \sum_j I_{ij} P(x_i)P(y_j)m_{ij}^s / \lambda_i, \\ R(s) &= sG(s) - \sum_i P(x_i)\log \lambda_i, \\ m_{ij} &= T(\theta_j|x_i)/T(\theta_j),\ \lambda_i = \sum_j P(y_j)m_{ij}^s, \end{aligned} \tag{27}$$

where $m_{ij} = T(\theta_j|x_i)/T(\theta_j) = P(x_i|\theta_j)/P(x_i)$ is the normalized likelihood and $\lambda_i = \sum_j P(y_j)m_{ij}^s$. The shape of any $R(G)$ function is a bowl-like curve with second derivative > 0, as shown in Figure 5.

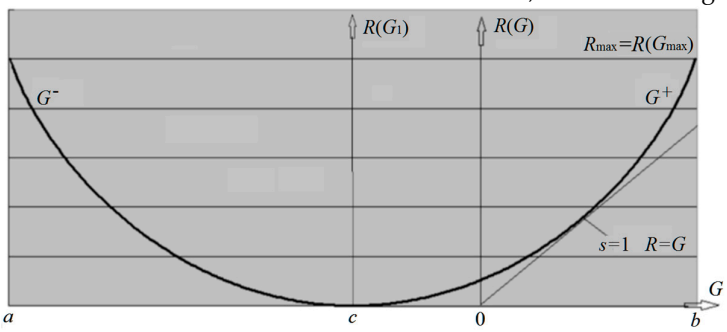


Figure 5. The rate-verisimilitude function $R(G)$ for binary communication. For any $R(G)$ function, there is a point where $R(G)=G$.

In Figure 5, $s = dR/dD$. When $s = 1$, R is equal to G, which means that the semantic channel matches the Shannon channel. G/R indicates the efficiency of semantic communication. In Section 3.4, we will see that solving a mixture model is equivalent to finding a parameter set θ that maximizes G/R, such that G/R is close to 1 or $G \approx R$.

When $s \to \infty$, R and G both reach their maxima R_{max} and G_{max}. As s increases, the TPFs $P(y_j|x)$, $j = 1, 2, \ldots, n$, will become steeper and the Shannon channel will have less noise. Hence, R and G will increase. This property of $R(G)$ can be used to prove the convergence of the CM iteration algorithm for the MMI classification of unseen instances.

The function $R(G)$ is different from $R(D)$. For a given R, there exists a maximum value G^+ and a minimum value G^-; G^- is negative, which means that we also need certain objective information R to bring a certain information loss $|G|$ to enemies. When $R = 0$, G is negative, which means that if we listen to someone who randomly predicts, the information that we already have will be reduced.

The function $R(G)$ was mainly developed for image compression, according to visual discrimination [25]. However, it can also be used for convergence proofs of MMI classification and mixture models.

2.2. From Traditional Bayes Prediction to Logical Bayesian Inference

2.2.1. Traditional Bayes Prediction, Likelihood Inference (LI), and Bayesian Inference (BI)

To understand LBI better, we will first review Traditional Bayes Prediction (TBP), LI, and BI. Note that "Bayes prediction" means the prediction according to Bayes' theorem, which is different from "Bayesian prediction" [3] that was made by Bayesians.

We call probability prediction with the TPF $P(y_j|x)$ TBP. For given $P(x)$ and $P(y_j|x)$, we can make a probability prediction

$$P(x|y_j) = P(x)\,P(y_j|x)/P(y_j). \tag{28}$$

When $P(y_j|x)$ is replaced with $kP(y_j|x)$, where k is a constant, $P(x|y_j)$ is the same, because

$$\frac{P(x)kP(y_j|x)}{\sum_i P(x_i)kP(y_j|x_i)} = \frac{P(x)P(y_j|x)}{\sum_i P(x_i)P(y_j|x_i)} = P(x|y_j). \tag{29}$$

Using this formula, we can easily explain that a truth function that is proportional to a TPF can be used for the same probability prediction.

For given $P(y_j)$, $P(x|y_j)$, and $P(x)$, we can obtain the predictive model

$$P(y_j|x) = P(y_j)\,P(x|y_j)/P(x). \tag{30}$$

After $P(x)$ is changed, we can still use $P(y_j|x)$ to make a new probability prediction, in most cases where the Shannon channels are stable.

We use the medical test (or signal detection) as an example to explain how a TPF or a Shannon channel can be used as a predictive model.

Definition 4. *Let z be an observed feature for an unseen instance (see Figure 1) and Z be a random variable, taking a value $z \in C = \{z_1, z_2, \ldots\}$. For unseen instance classification, x denotes a true class or true label.*

Assume that we classify every unseen instance with an unseen true label x, according to its observed feature $z \in C$. That is, we provide a classifier $y = f(z)$ to obtain a label y for z (see Figure 6).

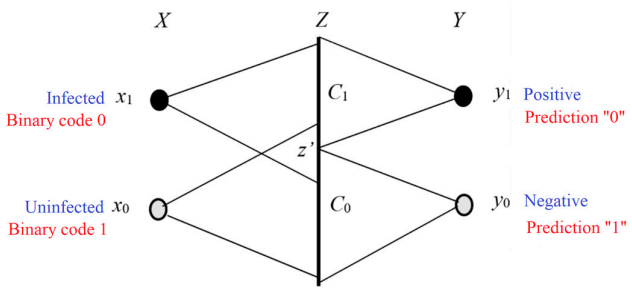


Figure 6. Illustrating the medical test and signal detection. We choose y_j according to $z \in C_j$. $\{C_0, C_1\}$ is a partition of C.

We use the HIV test to explain that the TPF can be used for probability prediction, with different $P(x)$. For an infected subject x_1, the conditional probability $P(y_1|x_1)$ of y_1 = positive is called sensitivity, which means the true positive rate. For an uninfected subject x_0, the conditional probability $P(y_0|x_0)$ of y_0 = negative is called specificity, which means the true negative rate [49]. The sensitivity and specificity ascertain a Shannon channel, as shown in Table 1.

Table 1. The sensitivity and Specificity of a Medical Test ascertain a Shannon Channel $P(Y|X)$.

	Infected Subject x_1	**Uninfected Subject x_0**
Positive y_1	$P(y_1|x_1)$ = sensitivity = 0.917	$P(y_1|x_0)$ = 1 − specificity = 0.001
Negative y_0	$P(y_0|x_1)$ = 1 − sensitivity = 0.083	$P(y_0|x_0)$ = specificity = 0.999

* Data are obtained from OREQuick HIV tests [50].

Example 2. *Calculate $P(x_1|y_1)$ using $P(y_1|x)$ in Table 1 for $P(x_1)$ = 0.0001, 0.002 (for normal people), and 0.1 (for high-risk crowd).*

Solution. *Using Equation (28), for $P(x_1)$ = 0.0001, 0.002, and 0.1, we have $P(x_1|y_1)$ = 0.084, 0.65, and 0.99, respectively.*

While using LI, it is not easy to solve Example 2. Nevertheless, when x is one of many different values and the sample size is not large enough, the TPFs cannot be smooth and, hence, we cannot use a TPF to obtain a smooth $P(x|y_j)$. This is why we use LI, which uses parameters to construct smooth likelihood functions. Using Maximum Likelihood Estimation (MLE), we can use a sample $\mathbf{X}$ to train a likelihood function to obtain the best θ_j:

$$\theta_j^* = \arg\max_{\theta_j} P(\mathbf{X}|\theta_j) = \arg\max_{\theta_j}[\sum_i P(x_i|.)\log P(x_i|\theta_j)], \tag{31}$$

where $P(x_i|.)$ indicates that y_j is unknown. The main defect of LI is that LI cannot make use of prior knowledge and that the optimized likelihood function will be invalid when $P(x)$ is changed.

Subjective Bayesians developed Bayesian Inference (BI) to make use of prior knowledge [2,3]. They brought the prior distribution $P(\theta)$ of θ into Bayes' Theorem to obtain the Bayesian posterior

$$P(\theta|\mathbf{X}) = \frac{P(\theta)P(\mathbf{X}|\theta)}{P_\theta(\mathbf{X})}\ ,\ P_\theta(\mathbf{X}) = \sum_j P(\theta_j)P(\mathbf{X}|\theta_j), \tag{32}$$

where $P_\theta(\mathbf{X})$ is the normalized constant related to θ and $P(\theta|\mathbf{X})$ is the posterior distribution of θ or the Bayesian posterior. Using $P(\theta|\mathbf{X})$, we can derive the Maximum A Posterior estimation:

$$\begin{aligned}\theta_j^* &= \arg\max_{\theta_j} P(\theta_j|\mathbf{X}_j) = \arg\max_{\theta} P(\theta_j)P(\mathbf{X}_j|\theta_j) \\ &= \arg\max_{\theta_j}[\sum_i P(x_i|.)\log P(x_i|\theta_j) + \log P(\theta_j)],\end{aligned} \tag{33}$$

where $P_\theta(\mathbf{X})$ is neglected.

BI has some advantages, such as

- it is especially suitable to cases where Y is a random variable for a frequency generator, such as a dice;
- as the sample size increases, the distribution $P(\theta|\mathbf{X})$ will gradually shrink to some θ_j^* coming from the MLE; and,
- BI can make use of prior knowledge better than LI.

However, BI also has some disadvantages:

- the probability prediction from BI [3] is not compatible with traditional Bayes prediction;
- $P(\theta)$ is subjectively selected; and,
- BI cannot make use of the prior of X.

If we try to use BI to solve Examples 1 and 2, we will find that the Bayesian posterior is not as good as TPF $P(y_j|x)$. Therefore, to make use of the prior of X, we still want a parameterized TPF $P(\theta_j|x)$.

2.2.2. From Fisher's Inverse Probability Function P($\theta_j|x$) to Logical Bayesian Inference (LBI)

De Morgan first called TPF $P(y_j|x)$ the "inverse probability", with respect to Laplace's method of probability [2]. The corresponding direct probability is $P(x|y_j)$. Later, Fisher called the likelihood function $P(x|\theta_j)$ the direct probability and the parameterized TPF $P(\theta_j|x)$ the inverse probability [2]. We use θ_j (instead of θ) and x (instead of x_i) to emphasize that θ_j is a constant and x is a variable, and hence $P(\theta_j|x)$ should be a function. In the following, we call $P(\theta_j|x)$ the Inverse Probability Function (IPF). According to Bayes' theorem,

$$P(\theta_j|x\ ) = P(\theta_j)\ P(x|\theta_j)/P(x), \tag{34}$$

$$P(x|\theta_j) = P(x_i)\ P(\theta_j|x)/P(\theta_j). \tag{35}$$

The IPF $P(\theta_j|x)$ can make use of the prior knowledge $P(x)$ well. When $P(x)$ is changed into $P'(x)$, we can still obtain $P'(x|\theta_j)$ from $P'(x)$ and $P(\theta_j|x)$.

When n = 2, we can easily construct $P(\theta_j|x)$, j=1,2, with parameters. For instance, we can use a pair of Logistic (or Sigmoid) functions as the IPFs. Unfortunately, when n > 2, it is hard to construct $P(\theta_j|x)$, j=1,2,…,n, because there is a normalization limitation $\sum_j P(\theta_j|x) = 1$ for every x. This is why a multiclass or multilabel classification is often converted into several binary classifications [51,52].

It seems that we may use the Softmax function as the IPF $P(\theta_j|x)$ for n > 2. However, this function is not compatible with $P(y_j|x)$, especially when two or more classes are not exclusive, the Softmax function does not work.

Using $P(\theta_j|x)$ and $P(y_j|x)$ as predictive models also has another disadvantage: In many cases, we can only know $P(x)$ and $P(x|y_j)$ without knowing $P(\theta_j)$ or $P(y_j)$, such that we cannot obtain $P(y_j|x)$ or $P(\theta_j|x)$. Nevertheless, we can obtain a truth function $T(\theta_j|x)$ in these cases. In LBI, there is no normalization limitation and, hence, it is easy to construct a group of truth functions and train them with $P(x)$ and $P(x|y_j)$, j=1,2,…,n, without $P(y_j)$ or $P(\theta_j)$. This is an important reason why we use LBI.

When a sample $\mathbf{X}_j$ is very large, we can directly obtain $T^*(\theta_j|x)$ from Equation (21). For a size-limited sample, we can use the generalized KL information formula to obtain

$$\begin{aligned} T^*(\theta_j|x) &= \arg\max_{T(\theta_j|x)} I(X;\theta_j) = \arg\max_{T(\theta_j|x)} \sum_i P(x_i|y_j)\log\frac{T(\theta_j|x_i)}{T(\theta_j)} \\ &= \arg\max_{T(\theta_j|x)} \sum_i P(x_i|y_j)\log\frac{P(x_i|\theta_j)}{P(x_i)}. \end{aligned} \tag{36}$$

This formula is the main formula that is used in LBI. LBI provides the Maximum Semantic Information Estimation (MSIE):

$$\begin{aligned} \theta_j^* &= \arg\max_{\theta_j} I(X;\theta_j) = \arg\max_{\theta_j}[\sum_i P(x_i|.)\log[P(x_i|\theta_j)/P(x_i)] \\ &= \arg\max_{\theta_j}[\sum_i P(x_i|.)\log T(\theta_j|x_i) - \log T(\theta_j)], \end{aligned} \tag{37}$$

which is compatible with MLE. If the samples are large enough, the MSIE, MLE, and MAP are equivalent.

We suggest using the truth function as the predictive model or the inferential tool for LBI in some cases, as it has the following advantages:

- we can use an optimized truth function $T^*(\theta_j|x)$ to make probability predictions for different $P(x)$ just as we would use $P(y_j|x)$ or $P(\theta_j|x)$;
- we can train a truth function with parameters by a sample with small size, as we would train a likelihood function;
- the truth function indicates the semantic meaning of a hypothesis and, hence, is easy for us to understand;

- it is also the membership function, which indicates the denotation of a label or the range of a class and, hence, is suitable for classification;
- to train a truth function, we only need $P(x)$ and $P(x|y_j)$, without needing $P(y_j)$ or $P(\theta_j)$; and,
- letting $T(\theta_j|x) \propto P(y_j|x)$, we construct a bridge between statistics and logic.

The CM algorithms further reveal these advantages.

3. Methods II: The Channel Matching (CM) Algorithms

3.1. CM1: To Resolve the Multilabel-Learning-for-New-P(x) Problem

3.1.1. Optimizing Truth Functions or Membership Functions

We use CM1 to denote the basic matching algorithm, in which the semantic channel matches the Shannon channel, and membership functions or truth functions are used as learning functions.

Assume that x is an age, y_j is a label "Youth", and θ_j is a fuzzy set $\{x|x$ is a youth$\}$. From population statistics, we can obtain a population age distribution $P(x)$ and a posterior distribution $P(x|y_j)$.

We can directly use Equation (21) to obtain the optimized membership function $T^*(\theta_j|x)$ without parameters if the sample is very large and, hence, the distributions $P(x)$ and $P(x|y_j)$ are smooth. If $P(x)$ and $P(x|y_j)$ are not smooth, we can use Equation (36) to obtain $T^*(\theta_j|x)$ with parameters. Without needing $P(y_j)$, in CM1, every label's learning for $T^*(\theta_j|x)$ is independent.

If the given sampling distribution is a TPF $P(y_j|x)$, we may assume that $P(x)$ is flat. Subsequently, Equation (36) becomes

$$T^*(\theta_j \mid x) = \arg\max_{T(\theta_j|x)} \sum_i \frac{P(y_j|x_i)}{\sum_k P(y_j|x_k)} \log \frac{T(\theta_j \mid x_i)}{\sum_k T(\theta_j|x_k)}. \quad (38)$$

If $P(y_j|x)$ is smooth, we can use Equation (22) to obtain $T^*(\theta_j|x)$ without parameters. For multilabel learning, we can directly obtain a group of truth functions from a Shannon channel $P(Y|X)$ or a sample with distribution $P(x,y)$. However, while using popular multilabel learning methods, such as Binary Relevance, we have to prepare several samples for several Logistic functions.

When $P(x)$ is changed, $T^*(\theta_j|x)$ is still useful for making semantic Bayes predictions.

3.1.2. For the Confirmation Measure of Major Premises

We use "degree of confirmation" to denote "degree of belief" supported by evidence or samples.

Bayesians use "degree of belief" to explain the subjective probability of a hypothesis. This degree of belief is between 0 and 1. However, researchers of induction use "degree of belief" to evaluate if–then statements or major premises. This degree of belief should be between −1 and 1. In this paper, we take "degree of belief" between −1 and 1 for the subjective evaluation of if–then statements. We know that the correlation coefficient between the two random variables is also between −1 and 1. The difference is that if–then statements are asymmetric; there is more than one major premise and degree of belief between the instance X and the hypothesis Y.

Now, we take the medical test as an example to explain how to use truth functions to replace TPFs or how to use the semantic channel to replace the Shannon channel for probability predictions.

From the Shannon channel in Table 1, we can derive the semantic channel, as shown in Table 2. Assume that $T(y_1|x_1) = T(y_0|x_0) = 1$ and $T(y_1|x_0) = T(y_0|x_1) = 0$ for non-fuzzy hypotheses. Two truth functions for corresponding fuzzy hypotheses are

$$T(\theta_1|x) = b_1' + b_1 T(y_1|x), \tag{39}$$

$$T(\theta_0|x) = b_0' + b_0 T(y_1|x), \tag{40}$$

where $b_1 = b(y_1 \to x_1)$, which is the degree of belief of major premise $MP_1 = y_1 \to x_1$ = "if $Y = y_1$ then $X = x_1$", and $b_1' = 1 - |b_1|$ means the degree of disbelief of MP_1 and the ratio of a tautology in y_1. Likewise, $b_0 = b(y_0 \to x_0)$ and $b_0' = 1 - b_0$.

Table 2. The two degrees of disbelief of the medical test form a semantic channel $T(\theta|X)$.

Y	**Infected x_1**	**Uninfected x_0**
Positive y_1	$T(\theta_1|x_1) = 1$	$T(\theta_1|x_0) = b_1'$
Negative y_0	$T(\theta_0|x_1) = b_0'$	$T(\theta_0|x_0) = 1$

According to Equations (21) and (22), the two optimized degrees of disbelief are

$$b_1'^* = P(y_1|x_0)/P(y_1|x_1) = [P(x_0|y_1)/P(x_0)]/[P(x_1|y_1)/P(x_1)], \tag{41}$$

$$b_0'^* = P(y_0|x_1)/P(y_0|x_0) = [P(x_1|y_0)/P(x_1)]/[P(x_0|y_0)/P(x_0)]. \tag{42}$$

For given y_1, we can use $b_1'^*$ and different $P(x)$ to make the semantic Bayes prediction:

$$P(x_1|\theta_1) = P(x_1)/[P(x_1) + b_1'^* P(x_0)], \tag{43}$$

$$P(x_0|\theta_1) = b_1'^* P(x_0)/[P(x_1) + b_1'^* P(x_0)]. \tag{44}$$

This prediction is equivalent to the traditional Bayes prediction with the TPF $P(y_j|x)$. We can still make the prediction, even if we only know $P(x|y_1)$ and $P(x)$ without knowing $P(y_1)$. It is easy to verify that, while using Equation (43) to solve Example 2, the results are the same as those that were obtained from the traditional Bayes prediction.

We will find it is not easy for the model to fit different $P(x)$ if we try to use LI or BI to obtain a predictive model for medical tests.

In comparison with the Shannon channel in Table 1, the semantic channel in Table 2 is easier to understand and remember. To remember $P(y_1|x)$, we need to remember two numbers; whereas, to remember $T^*(\theta_1|x)$, we only need to remember one number $b_1'^*$.

In [41], the author provided two formulae for positive and negative degrees of confirmation. These two formulae can be merged into a new formula:

$$\begin{aligned} b_1^* = b^*(y_1 \to x_1) &= \frac{P(y_1|x_1) - P(y_1|x_0)}{\max(P(y_1|x_1), P(y_1|x_0))} = \frac{\text{sensitivity-(1-specificity)}}{\max(\text{sensitivity}, \text{1-specificity})} \\ &= \frac{\text{True\_positive\_rate} - \text{False\_positive\_rate}}{\max(\text{True\_positive\_rate}, \text{False\_positive\_rate})} = \frac{CL_1 - CL_1'}{\max(CL_1, CL_1')}, \end{aligned} \tag{45}$$

where $CL_1 = P(y_1|x_1)/[P(y_1|x_0) + P(y_1|x_1)]$ is the confidence level of MP_1 and $CL_1' = 1 - CL_1$. As CL_1 changes from 0 to 1, b_1^* changes from −1 to 1, as shown in Figure 7.

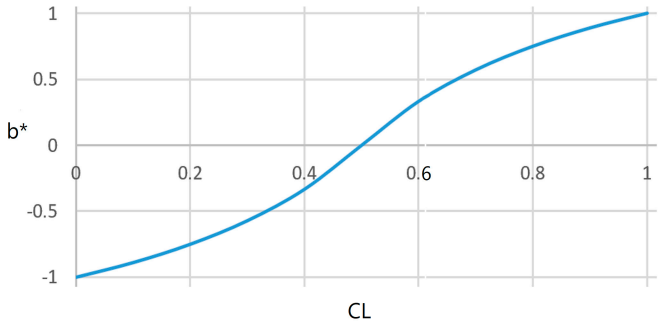


Figure 7. Relationship between degree of conformation b^* and confidence level CL. As CL changes from 0 to 1, b^* changes from −1 to 1.

3.1.3. Rectifying the Parameters of a GPS Device

If we do not know the real parameters of a GPS device or are suspicious of the parameters claimed by the producer, we can assume

$$T(\theta_j|x) = \exp[-|x-(x_j+\triangle x)|^2/(2\sigma^2)], \tag{46}$$

where x is a two-dimensional vector. Subsequently, we can use a sample to find the parameters $\triangle x$ (the systematic deviation) and σ. We may obtain the sample by driving a car with the GPS device around a big square and recording the relative positions $x' = x - x_j$. From many relative deviations, we can obtain a sampling distribution $P(x'|y_j)$. As we are driving on a big square, $P(x)$ should be flat. Afterwards, we can use the generalized KL information formula to obtain the optimized parameters $\triangle x^*$ and σ^*. Subsequently, we replace y_j with y_k = "X is about x_k", where $x_k = x_j + \triangle x^*$.

Assuming that the GPS device is often faulty, we can also use

$$T(\theta_j|x) = b\exp[-|x-(x_j+\triangle x)|^2/(2\sigma^2)] + 1 - b \tag{47}$$

as the learning function to obtain the degree of confirmation b^* of the GPS device.

If one tries to use the inverse likelihood function $P(\theta_j|x)$ or the Bayesian posterior $P(\theta|\mathbf{X})$ for the above task and probability prediction (see Figure 3), they will find that it is difficult to do, because they only have prior knowledge $P(x)$ from a GPS map, without prior knowledge $P(y)$ or $P(\theta)$.

3.2. CM2: The Semantic Channel and the Shannon Channel Mutually Match for Multilabel Classifications

CM2 includes two steps:

- **Matching I:** Let the semantic channel match the Shannon channel or use CM1 for multilabel learning; and,
- **Matching II:** Let the Shannon channel match the semantic channel by using the Maximum Semantic Information (MSI) classifier.

Both steps use the MMI or ML criterion.

For multilabel learning, we may train every label by Equation (36) or Equation (38). We may also train a label y_j with membership function $T(\theta_j|x)$ and its negative label y_j' with membership function $1 - T(\theta_j|x)$, at the same time, as in the popular method of [51,52], by

$$\begin{aligned} T^*(\theta_j|x) &= \arg\max_{T(\theta_j|X)}[I(X;\theta_j)+I(X;\theta_j^c)] \\ &= \arg\max_{T(\theta_j|x)}\sum_i[P(x_i|y_j)\log\frac{T(\theta_j|x_i)}{T(\theta_j)}+P(x_i|y_j')\log\frac{1-T(\theta_j|x_i)}{1-T(\theta_j)}], \end{aligned} \tag{48}$$

where θ_j^c is the complementary set of θ_j. The obtained $T^*(\theta_j|x)$ may be a Logistic function, which will cover a larger area of U, in comparison with $T^*(\theta_j|x)$ from Equation (36) or Equation (38).

If there are examples with one instance and several labels, or with several instances and one label, we may split such an example into several single-instance and single-label examples, in the manner of the popular method in [51]. Subsequently, we can obtain the Shannon channel $P(Y|X)$ for multilabel learning.

For classifications where instances are visible, x is given. In Matching II, the MSI classifier is

$$y_j^* = h(x) = \arg\max_{y_j}\log I(x;\theta_j) = \arg\max_{y_j}\log[T(\theta_j|x)/T(\theta_j)] \tag{49}$$

While using $T(\theta_j)$, we can overcome the class-imbalance problem [50] and reduce the rate of failure to report smaller probability events. If $T(\theta_j|x) \in \{0,1\}$, the semantic information measure

becomes Carnap and Bar-Hillel's semantic information measure, and the classifier becomes the minimum logical probability classifier:

$$y_j^*=h(x)=\underset{y_j \text{ with } T(A_j|x)=1}{\arg\max} \log[1/T(A_j)]=\underset{y_j \text{ with } T(A_j|x)=1}{\arg\min} \log T(\theta_j) \tag{50}$$

This criterion encourages us to select a compound label with smaller denotation.

For unseen instance classifications or uncertain x, we only have knowledge of $P(x|z)$. Afterwards, the MSI classifier becomes

$$y_j^*=f(z)=\underset{y_j}{\arg\max}\sum_i P(x_i|z)\log\frac{T(\theta_j|x_i)}{T(\theta_j)}. \tag{51}$$

To simplify multilabel learning, we may train fewer atomic labels and use them and the fuzzy logic, which is compatible with Boolean algebra [22] to produce the membership function of a compound label for multilabel classifications [44].

In the popular method for multilabel classifications while using the Bayes classifier or the MPP criterion, for different x the classifier compares two IPFs $P(\theta_j|x)$ and $P(\theta_k|x)$, such as two Logistic functions, to select a label with greater IPF. This method is not compatible with the information criterion or the likelihood criterion.

3.3. CM3: the CM Iteration Algorithm for MMI classification of Unseen instances

We use CM3 to denote the CM iteration algorithm, which repeats the two matching steps (i.e., Matching I and Matching II). CM2 is not an iterative algorithm; nevertheless, CM3 is. This algorithm is used for MMI classification, for which the most popular method is Gradient Decent.

We use the medical test, as shown in Figure 6, as an example to explain the problem with the MMI classification of unseen instances.

We need to optimize z' for the MMI. The problem is that, without the classifier $f(z)$, we cannot express the mutual information $I(X; Y)$; whereas, without the expression of mutual information, we cannot optimize the classifier $f(z)$. This problem is also met by MLE for uncertain Shannon channels. To resolve this problem, researchers generally use parameters to construct partition boundaries and then use Gradient Descent or the Newton method to search for the best MMI parameters. The CM iteration algorithm for MMI classification is different. It uses numerical values to express boundaries and information gain functions (e.g., reward functions). It repeatedly updates information gain functions and boundaries to achieve MMI.

Let C_j be a subset of C and $y_j=f(z|z\in C_j)$; hence, $S=\{C_1, C_2, \ldots\}$ is a partition of C. Our aim is, for given $P(x, z)$ from **D**, to find the optimized S, as given by

$$S^*=\underset{S}{\arg\max}\, I(X;\theta|S)=\underset{S}{\arg\max}\sum_j\sum_i P(C_j)P(x_i|C_j)\log\frac{T(\theta_j|x_i)}{T(\theta_j)} \tag{52}$$

Matching I: Let the semantic channel match the Shannon channel.

First, we obtain the Shannon channel for a given S:

$$P(y_j|x)=\sum_{z_k\in C_j} P(z_k|x),\ j=1,2,\ldots,n \tag{53}$$

From this Shannon channel, we obtain the semantic channel $T(\theta|X)$ and the semantic information $I(x_i; \theta_j)$. Subsequently, for given z, we obtain the information gain functions:

$$I(X_i;\theta_j|z)=\sum_i P(x_i|z)I(x_i;\theta_j)\text{ , }j=0,1,\ldots,n, \tag{54}$$

which are some curved surfaces over a two-dimensional feature space, as U is two-dimensional. We may directly let $I(x_i; \theta_j) = I(x_i; y_j) = \log[P(y_j|x)/P(y_j)]$. However, with the notion of the semantic channel, we can understand this algorithm and better prove its convergence.

Matching II: Let the Shannon channel match the semantic channel by the classifier

$$y_j^* = f(z) = \arg\max_{y_j} I(X_i;\theta_j|z),\ j=0,1,\ldots,n. \tag{55}$$

Repeat Matching I and II until S does not change. The convergent S is the S^* we seek.

Using Matching II for the optimization of the Shannon channel can reduce noise. We can understand the two matching steps in this way: Matching I is for the reward function; Matching II is for the Bayes decision.

For a given source $P(X)$, a semantic channel ascertains an $R(G)$ function. An improved $R(G)$ function has a higher matching point; that is, where $R(G) = G$. CM3 finds this matching point, which is also the point that attains R_{max} and G_{max} (see Figure 8).

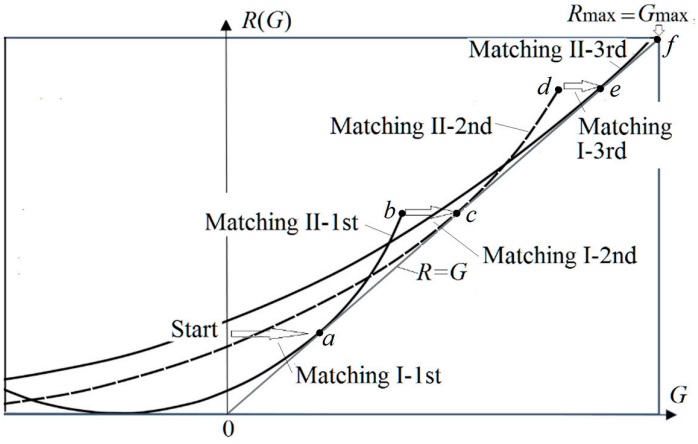


Figure 8. Illustrating the iterative convergence of the MMI classification of unseen instances. In the iterative process, (G, R) moves from the start point to $a,b,c,d,e,\ldots,f$ gradually.

We can prove that the iteration will converge. In the iterative process, the coordinate (G, R) changes, as follows:

$$\text{Start} \xrightarrow{\text{Matching I-1st}} a \xrightarrow{\text{Matching II-1st}} b \xrightarrow{\text{Matching I-2nd}} c \xrightarrow{\text{Matching II-2nd}} d \xrightarrow{\text{Matching I-3rd}} e \xrightarrow{\text{Matching II-3rd}} \ldots f$$

This process continues until Matching II cannot improve R and G.

The coordinate (G, R) can converge to (G_{max}, R_{max}), as every Matching I procedure increases G and every Matching II procedure increases G and R, and the maxima of G and R are finite.

Matching II can always find any best partition for given $I(X; y_j|z)$, $j=1,2,\ldots$, because it checks every z to see which of the $I(X; y_j|z)$, $j=1,2\ldots$, is the maximum.

We can understand the CM iteration algorithm in the following way: The function $R(G)$ is like a ladder, and the coordinate (G, R) is like a climber. In Matching I, (G, R) creates a ladder and then moves on it. In Matching II, it climbs up to the top of the ladder. Afterwards, the process is repeated, creating new ladders until (G_{max}, R_{max}) is reached.

3.4. CM4: the CM-EM Algorithm for Mixture Models

We use CM4 to denote the CM-EM Algorithm: An Improved Expectation-Maximization (EM) Algorithm for mixture models.

In CM3, Matching II is used to find the maximum R, whereas, in CM4, Matching II is used to find the maximum information efficiency G/R or minimum $R - G$.

CM4 is based on a different convergence theory of the mixture models. The popular convergence theory of the EM algorithm explains that we can maximize the incomplete data log-likelihood $L_X(\theta)$ by maximizing the complete data log-likelihood Q, whereas the convergence

theory of the CM-EM algorithm explains that we can maximize $L_X(\theta)$ by maximizing the information efficiency G/R.

The EM algorithm [53] for mixture models has been shown to often result in slow or invalid convergence [54,55]. We can improve the EM algorithm by letting the semantic channel and the Shannon channel mutually match. The difference is that Matching II is used to find the minimum of the Shannon mutual information R.

If a probability distribution $P_\theta(x)$ comes from the mixture of n likelihood functions, such as

$$P_\theta(x) = \sum_{j=1}^{n} P(y_j)P(x \mid \theta_j) \tag{56}$$

Subsequently, we call $P_\theta(x)$ a mixture model. If every predictive model $P(x \mid \theta_j)$ is a Gaussian function, then $P_\theta(x)$ is a Gaussian mixture. In the following, we use $n = 2$ to discuss the algorithms for mixture models.

Assume that $P(x)$ comes from the mixture of two true model $P(x \mid \theta_1^*)$ and $P(x \mid \theta_2^*)$ with ratios of $P^*(y_1)$ and $P^*(y_2) = 1 - P^*(y_1)$; that is,

$$P(x)=P^*(y_1)P(x \mid \theta_1^*)+P^*(y_2)P(x \mid \theta_2^*). \tag{57}$$

We only know $P(x)$ and $n = 2$. We can use the guessed parameters and mixture ratios to obtain

$$P_\theta(x) = P(y_1)P(x \mid \theta_1) + P(y_2)P(x \mid \theta_2). \tag{58}$$

Subsequently, we have the observed data log-likelihood

$$L_X(\boldsymbol{\theta}) = N\sum_i P(x_i)\log P_\theta(x_i) = -NH_\theta(X), \tag{59}$$

and the relative entropy or KL divergence:

$$H(P \,||\, P_\theta) = \sum_i P(x_i)\log\frac{P(x_i)}{P_\theta(x_i)} = H_\theta(X) - H(X). \tag{60}$$

If the two distributions $P(x)$ and $P_\theta(x)$ are close to each other, such that the relative entropy is close to 0, for example, less than 0.001 bit for a huge sample or 0.01 bit for a sample with size = 1000, then, we may say that our guess is right. Therefore, our task is to change θ and $P(y)$ to maximize the likelihood $L_X(\theta) = \log P(\mathbf{X} \mid \theta)$ or to minimize the relative entropy $H(P \mid\mid P_\theta)$.

The main formula of the EM algorithm for mixture models can be described, as follows:

$$\begin{aligned} Q &= N\sum_i\sum_j P(x_i)P(y_i \mid x_i)\log P(x_i, y_j \mid \theta) \\ &= L_X(\boldsymbol{\theta}) + N\sum_i\sum_j P(x_i)P(y_j \mid x_i)\log P(y_j \mid x_i), \end{aligned} \tag{61}$$

where $Q = -NH(X, Y \mid \theta)$ is called the complete data log-likelihood and $P(y_j \mid x)$ is from Equation (63). There exists

$$L_X(\theta) = Q + H, \tag{62}$$

where $H = -NH(Y \mid X,\theta)$ is a Shannon conditional entropy. A popular convergence theory of the EM algorithm explains that we can increase $L_X(\theta)$ by increasing Q.

The steps in the EM algorithm are:

E-step: Write the conditional probability functions (e. g., the Shannon channel):

$$\begin{aligned} P(y_j \mid x) &= P(y_j)P(x \mid \theta_j) / P_\theta(x), \\ P_\theta(x) &= \sum_j P(y_j)P(x \mid \theta_j). \end{aligned} \tag{63}$$

M-step: Improve $P(y)$ and θ to maximize Q. If Q cannot be further improved, then end the iteration process; otherwise, go to the E-step.

Neal and Hinton [56] proposed an improved EM algorithm, the Maximization–Maximization (MM), in which Q is replaced with $F = Q + H(Y)$ and F is maximized in both steps.

Almost all of the EM algorithm researchers believe that Q and $\log Lx(\theta)$ are positively correlated and that the E-step does not decrease Q; nevertheless, this is not true. The author found that Q may decrease in some E-steps; and, Q and F should decrease in some cases [42].

Using the CM algorithm to improve the EM algorithm, we have developed an algorithm, the CM-EM algorithm, for better convergence.

The CM-EM algorithm includes three steps:

E1-step: Construct the Shannon channel. This step is the same as the E-step of the EM algorithm.

E2-step: Repeat the following three equations until $P^{+1}(y)$ converges to $P(y)$:

$$\begin{aligned} &P^{+1}(y_j) = \sum_i P(x_i)P(y_j \mid x_i) = \sum_i P(x_i)P(y_j \mid x_i),\ j = 1, 2, \ldots; \\ &P(y_j) = P^{+1}(y_j); \\ &P(y_j \mid x_i) = P(x_i \mid \theta_j)P(y_j) / \sum_k P(y_k)P(x_i \mid \theta_k),\ i = 1, 2, \ldots; j = 1, 2, \ldots \end{aligned} \tag{64}$$

If $H(P||P_\theta)$ is less than a small value, then end the iteration.

MG-step: Optimize the parameters θ_j^{+1} of the likelihood function in log(.) to maximize G:

$$G = I(X;\theta) = \sum_i \sum_j P(x_i) \frac{P(x_i \mid \theta_j)}{P_\theta(x_i)} P(y_j) \log \frac{P(x_i \mid \theta_j^{+1})}{P(x_i)} \tag{65}$$

Then, go to the E1-step.

As G reaches a maximum when $P(x \mid \theta_j^{+1})/P(x) = P(x \mid \theta_j)/P_\theta(x)$, the new likelihood function is

$$P(x \mid \theta_j^{+1}) = P(x)P(x \mid \theta_j)/P_\theta(x). \tag{66}$$

Without the E2-step, $P(x \mid \theta_j^{+1})$ above is, in general, not normalized [57]. For Gaussian mixtures, we can easily obtain new parameters:

$$\begin{aligned} &\mu_j^{+1} = \sum_i P(x_i)P(x_i \mid \theta_j) / P_\theta(x_i),\ j = 1,\ 2, \ldots, n; \\ &\sigma_j^{+1} = \{\sum_i P(x_i)[P(x_i \mid \theta_j^{+1}) - \mu_j^{+1}]^2\}^{0.5},\ j = 1,\ 2 \ldots, n. \end{aligned} \tag{67}$$

If the likelihood functions are not Gaussian distributions, we can find optimized parameters by searching the parameter space, using methods such as Gradient Descent.

To prove the convergence of the CM-EM algorithm, we may make use of the properties of the function $R(G)$:

- the function $R(G)$ is concave and $R(G) - G$ has the exclusive minimum 0 at $R(G) = G$ [25]; and,
- $R(G) - G$ is close to the relative entropy $H(P||P_\theta)$.

After the E1-step, the Shannon mutual information $I(X; Y)$ becomes

$$R = \sum_i \sum_j P(x_i) \frac{P(x_i \mid \theta_j)}{P_\theta(x_i)} P(y_j) \log \frac{P(y_j \mid x_i)}{P^{+1}(y_j)} \tag{68}$$

We define

$$R'' = \sum_i \sum_j P(x_i) \frac{P(x_i \mid \theta_j)}{P_\theta(x_i)} P(y_j) \log \frac{P(x_i \mid \theta_j)}{P_\theta(x_i)} \tag{69}$$

It is easy to prove that $R'' - G = H(P \| P_\theta)$. Hence,

$$H(P \| P_\theta) = R'' - G = R - G + H(Y \| Y^{+1}) \tag{70}$$

where

$$H(Y^{+1} \| Y) = \sum_j P^{+1}(y_j) \log[P^{+1}(y_j) / P(y_j)] \tag{71}$$

Proving that $P_\theta(X)$ converges to $P(X)$ is equivalent to proving that $H(P \| P_\theta)$ converges to 0. As the E2-step forces $R = R''$ and $H(Y^{+1} \| Y) = 0$, we only need to prove that every step minimizes $R - G$. It is evident that the MG-step minimizes $R - G$, because this step maximizes G without changing R. The remaining problem is how to prove that $R - G$ is minimized in the E1- and E2-steps. Learning from the variational and iterative methods that Shannon [30] and others [48] have used for analyzing the rate-distortion function $R(D)$, we can optimize $P(y \mid x)$ and $P(y)$, respectively, to minimize $R - G = I(X; Y) - I(X; \theta)$. As $P(Y \mid X)$ and $P(Y)$ are interdependent, we can only fix one to optimize the other; the E2-step is for exactly this purpose. For the detailed convergence proof, see [57].

4. Results

4.1. The Results of CM2 for Multilabel Learning and Classification

We used a prior distribution $P(x)$ and a posterior distribution $P(x \mid y_j)$ to optimize a truth function in order to obtain $T^*(\theta_j \mid x)$, as shown in Figure 9.

For $P(x)$ and $P(x \mid y_j)$, we first used a Gaussian random number generator to produce two samples, S_1 and S_2. Both sample sizes were 100,000. The data with distribution $P(x)$ was a part of S_1. We have

$$P(x) \approx \begin{cases} k \exp[-(x/40)^2], & 0 \le x \le 86; \\ 0, & \text{otherwise.} \end{cases}$$

where k is a normalizing constant. S_2 had distribution $P_2(x)$. $P(x \mid y_j)$ was produced from $P_2(x)$ and the following truth function:

$$T(\theta_2 \mid x) = \begin{cases} \exp[-(x-18)^2 / (2 \times 3^2)], & x < 18; \\ 1, & 18 \le 1 \le 25; \\ \exp[-(x-25)^2 / (2 \times 4^2)], & x > 25. \end{cases}$$

This meant that $P(x \mid y_j) = P_2(x \mid \theta_2) = P_2(x) T(\theta_2 \mid x) / T(\theta_2)$.

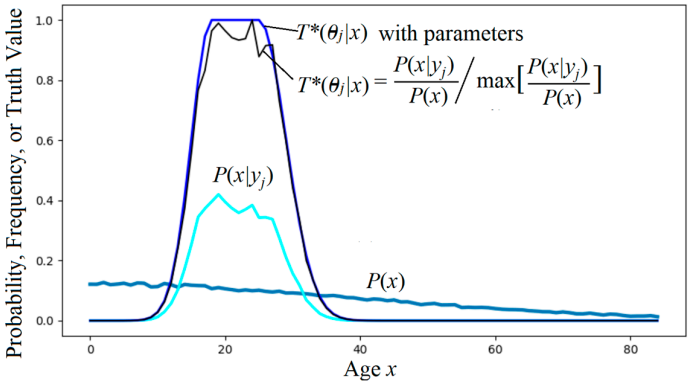


Figure 9. Using prior and posterior distributions $P(x)$ and $P(x|y_j)$ to obtain the optimized truth function $T^*(\theta_j|x)$.

Subsequently, we obtained $T^*(\theta_j|x)$ from $P(x)$ and $P(x|y_j)$. If we directly used the formula in Equation (21), $T^*(\theta_j|x)$ would not be smooth. We set a truth function with parameters

$$T(\theta_j|x)=\begin{cases}\exp[-(x-\mu_{j1})^2/(2\sigma_{j1}^2)], & x<\mu_{j1};\\ 1, & \mu_{j1}\le 1\le\mu_{j2};\\ \exp[-(x-\mu_{j2})^2/(2\sigma_{j2}^2)], & x>\mu_{j2}.\end{cases}$$

Afterwards, we used the Generalized KL information formula to optimize $T(\theta_j|x)$ to obtain smooth $T^*(\theta_j|x)$. If $S_2 = S_1$, then $T^*(\theta_j|x) = P(x|y_j)/P(x)/\max[P(x|y_j)/P(x)] = T(\theta_2|x)$.

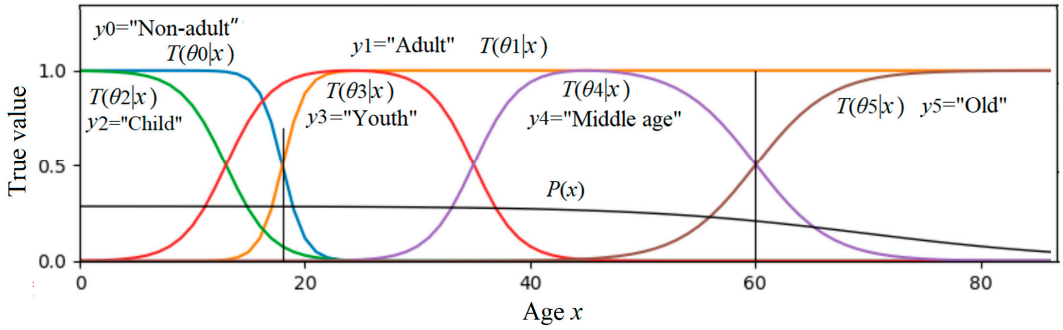
Figure 10 shows the MSI classification of ages for given prior distribution $P(x)$ and the truth functions of five labels. The five labels were (y_1, y_2, y_3, y_4, y_5)=("Adult", "Child", "Youth", "Middle age", "Old"). Figure 10a shows the truth functions of the five labels.

Among these truth functions, each of $T(\theta_3|x)$ and $T(\theta_4|x)$ were constructed by two logistic functions; each of others was a logistic function. The python 3.6 source file with parameters for Figures 9–15 can be found in Appendix B.

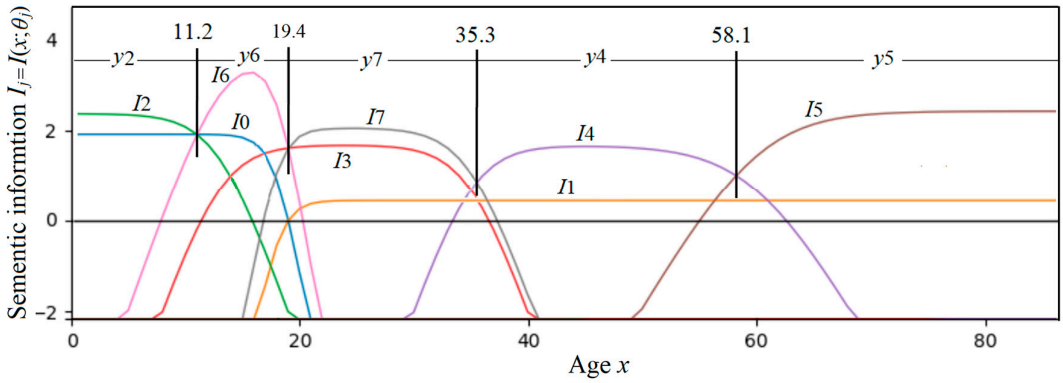
We could also treat these truth functions as the learning functions $P(\theta_j|x)$ that were obtained from the popular methods, and then use the Bayesian classifier or the Maximum Posterior Probability criterion to classify them.

In Figure 10a, $P(x)$ was assumed: $P(x) = k[1 - 1/\exp[-0.1(x - 70)]$ for $x > 0$, where k was a normalizing constant. From the five labels, we also had the compound labels $y_0 = y_1'$, $y_6 = y_3 \wedge y_1'$, and $y_7 = y_3 \wedge y_1$.

Figure 10b shows the effect of the MSI classifier.



(**a**) The truth functions of five labels for ages and the prior distribution $P(x)$ of the population



(**b**) Labeling x according to which of $I_j = I(x;\theta_j)$ ($j = 0,1,\ldots,7$) is maximum

Figure 10. The maximum semantic information classification of ages.

Figure 10 indicates that the Maximum Posterior Probability (MPP) criterion and the MSI criterion resulted in different classifications. Using the MPP criterion, we only selected y_0 = "Non-adult" or y_1 = "Adult", for most ages. However, while using the MSI criterion, we selected y_2, y_6, y_7, y_4, and y_5, in turn, as the age x increased. The MSI criterion encouraged us to use more labels with smaller logical probabilities. For example, if x was between 11.2–16.6, we should use the label $y_6 = y_3 \wedge y_1'$ = "Youth" and "Non-adult". However, for most x, CM2 did not use redundant labels, as Binary Relevance [52] does. For example, while using the MSI criterion, we did not add the label "Non-youth" to x = 60, with the label "Old" already.

4.2. The Results of CM3 for the MMI classifications of Unseen Instances

CM3 was tested by many examples.

Example 3. *The value of z changed from 0 to 100 with step length 1. Two Gaussian distributions $P(z|x_0)$ and $P(z|x_1)$ had parameters $\mu_0 = 30$, $\mu_1 = 70$, $\sigma_0 = 15$, and $\sigma_1 = 10$; $P(x_0) = 0.8$ and $P(x_1) = 0.2$. The initial partitioning point z' was 50.*

The iterative process: Matching II-1 obtained z' = 53; Matching II-2 obtained z' = 54; Matching II-3 obtained z^* = 54.

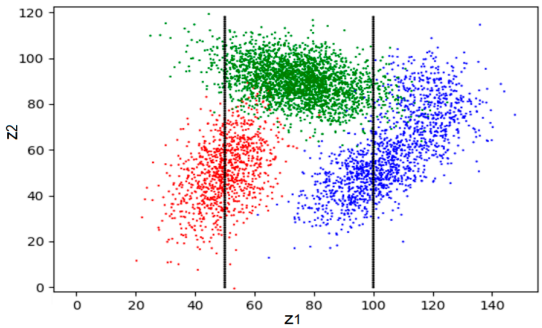
The following is a two-dimensional example.

Example 4. ***(****See Figure 11****)*** *There were three classes. The left two classes were two Gaussian distributions—$P(z|x_0)$ and $P(z|x_1)$—and the right one was a mixture of two Gaussian distributions—$P(z|x_{21})$ and $P(z|x_{22})$. The sample size was 1000. See Table 3 for the parameters of the four Gaussian distributions.*

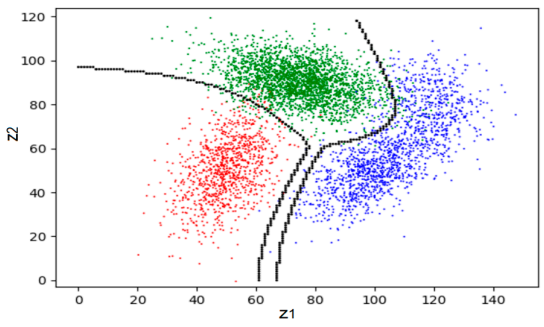
Table 3. The parameters of four Gaussian distributions.

	μ_{z1}	μ_{z2}	σ_{z1}	σ_{z2}	ρ	$P(x_i)$
$P(z\|x_0)$	50	50	75	200	50	0.2
$P(z\|x_1)$	75	90	200	75	-50	0.5
$P(z\|x_{21})$	100	50	125	125	75	0.2
$P(z\|x_{22})$	120	80	75	125	0	0.1

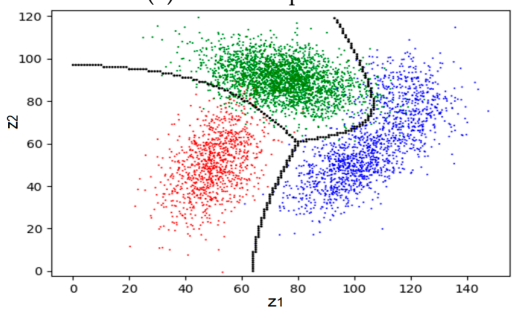
Two vertical lines made the initial partition. Figure 11 shows the iterative process.

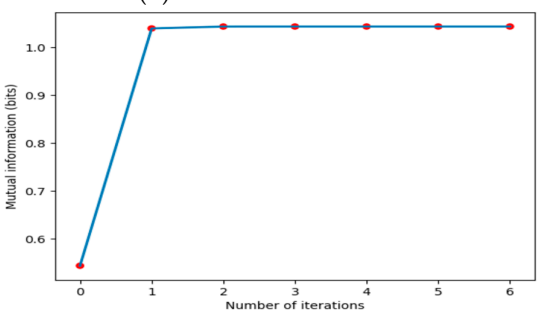


(**a**) The initial partition



(**b**) After the first iteration



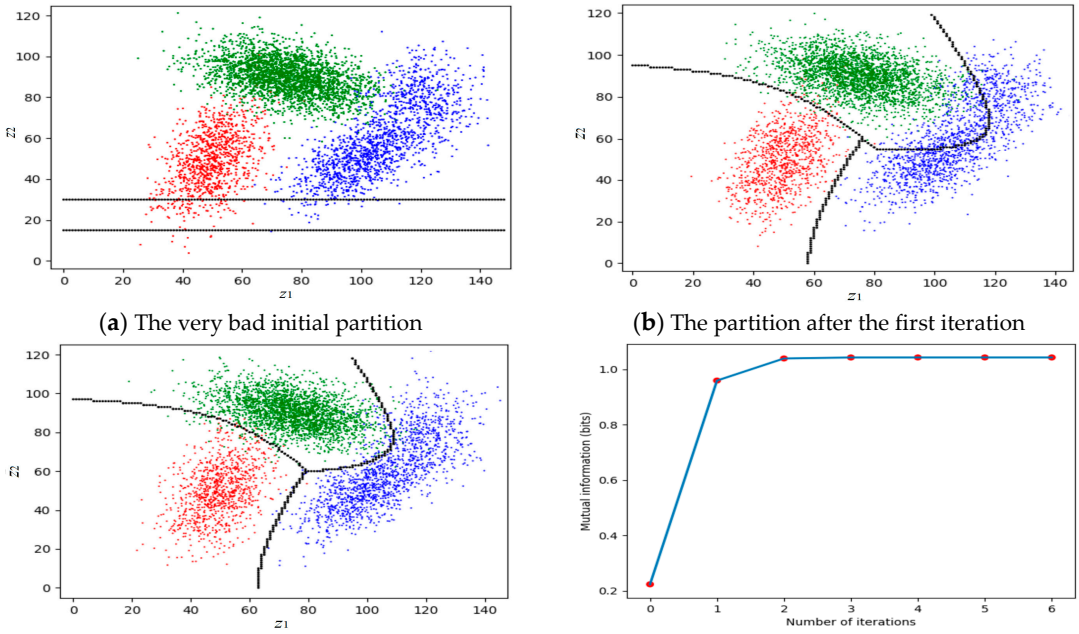


(**c**) After the second iteration (**d**) The mutual information changes with iterations

Figure 11. The Maximum Mutual Information (MMI) classification of unseen instances. The classifier is $y = f(z)$. The mutual information is $I(X; Y)$. X is a true class and Y is a selected label.

After two iterations, the mutual information $I(X; Y)$ was 1.0434 bits. The convergent MMI was 1.0435 bits. Only two iterations were required for the mutual information to reach 99.99% of the convergent MMI.

The author used a very bad initial partition to test the reliability of CM3. The convergence was also very fast in this case (see Figure 12).



(**a**) The very bad initial partition

(**b**) The partition after the first iteration

(**c**) The partition after the second iteration

(**d**) The mutual information changes with iterations

Figure 12. MMI classification with a very bad initial partition.

The author used the above example with different parameters and different initial partitions to test CM3. All of the iterative processes were fast and valid. In most cases, only 2–3 iterations were required for the mutual information to surpass 99% of the MMI.

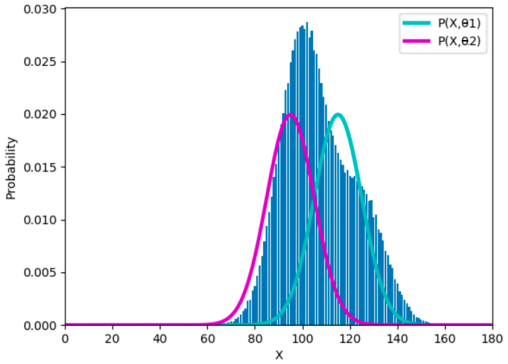
4.3. The Results of CM4 for Mixture Models

The following three examples show that the CM-EM algorithm can outperform both the EM and MM algorithms.

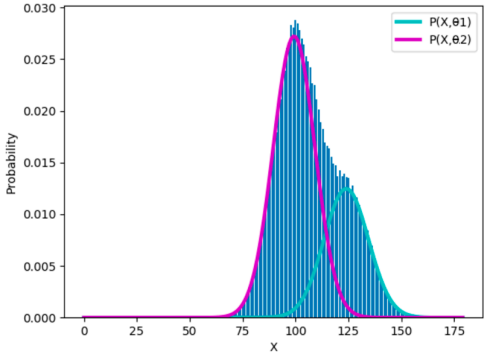
Ueda and Nakano [54] proposed an example to show that local or invalid convergence is inevitable in the EM algorithm because some initial parameters result in a local maximum of Q. This invalid convergence was also verified by Marin et al. [55]. Their example is as follows:

Example 5. *A mixture model has two Gaussian components. The true model parameters are* $(\mu_1^*, \mu_2^*, \sigma_1^*, \sigma_2^*, P^*(y_1)) = (100, 125, 10, 10, 0.7)$. *The invalid convergence is centered on the* μ_1–μ_2 *plane at* $(\mu_1, \mu_2) = (115, 95)$, *where Q reaches its local maximum.*

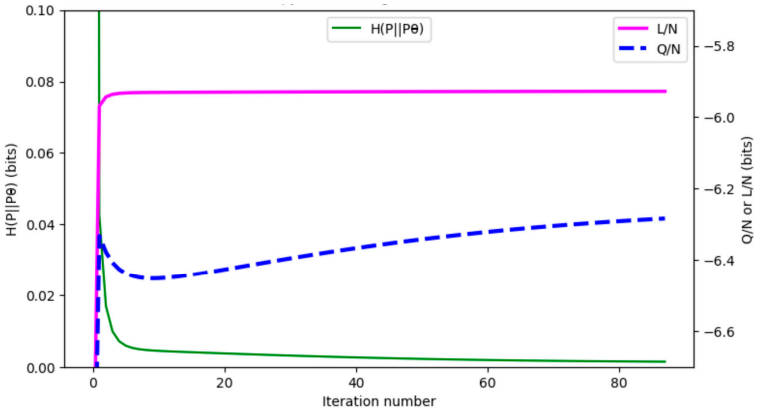
We used this example with initial parameters $(\mu_1, \mu_2, P(y_1), \sigma_1, \sigma_2) = (115, 95, 0.5, 10, 10)$ to test the CM-EM algorithm, in order to see whether $(\mu_1, \mu_2,)$ can converge to (μ_1^*, μ_2^*)=(100, 125). Figure 13 shows the result, which indicates that $L_X(\theta)$ converged to its global maximum under the CM-EM algorithm.



(**a**) Q is initially close to the local maximum



(**b**) $Lx(\theta)$ converges to the global maximum after 63 iterations



(**c**) Q decreases after the first E2-step and increases as $Lx(\theta)$ increases in the Channel Matching-Expectation-Maximization (CM-EM) algorithm.

Figure 13. The iterative process from the local maximum of Q to the global maximum of $Lx(\theta)$. The stopping condition is when the deviation of every parameter is smaller than 1%.

The following example is to compare the iteration numbers of different algorithms. Neal and Hinton [56] used this example to compare their Maximization–Maximization (MM) algorithm with the EM algorithm. Now, we use the same example to compare the CM-EM algorithm with the EM and MM algorithms.

Example 6. *Table 4 shows the true and initial parameters, including mixture ratios. The transforming formula was $x = 20(x' - 50)$, where x' is an original data point and x is a data point used in Table 4. It was assumed that $P(x)$ comes from two Gaussian functions with real parameters. Using the CM-EM algorithm, we obtained $H(P||P_\theta) = 0.00072$ bit after nine E1- and E2-steps and eight MG-steps.*

Table 4. True and guessed model parameters and iterative results of Example 6.

	Real Parameters			**Starting Parameters** $H(P\|\|P_\theta) = 0.68$ bit			**Parameters after 9 E2-steps** $H(P\|\|P_\theta) = 0.00072$ bit		
	μ^*	σ^*	$P^*(Y)$	μ	σ	$P(Y)$	μ	σ	$P(Y)$
y_1	46	2	0.7	30	20	0.5	46.001	2.032	0.6990
y_2	50	20	0.3	70	20	0.5	50.08	19.17	0.3010

The iterative process is shown in Figure 14.

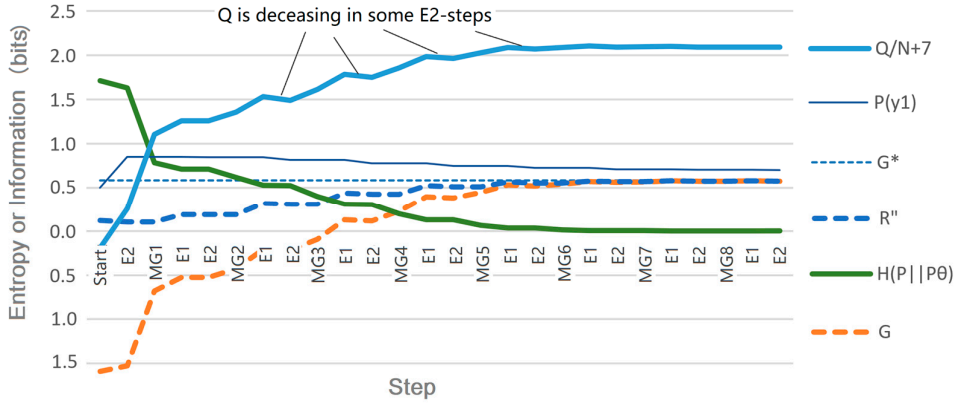


Figure 14. The iterative process of the CM-EM algorithm for Example 6. It can be seen that some E2-steps decrease Q. The relative entropy is smaller than 0.001 bit after nine iterations.

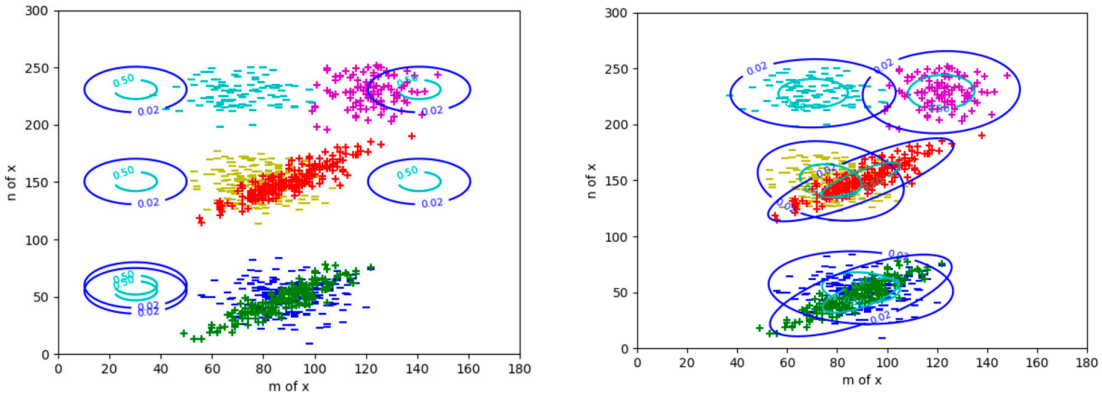
The author also used a sample whose size was 1000 to produce $P(x)$ to test the CM-EM algorithm. Table 5 shows the iteration numbers and the final parameters for the three different algorithms.

Table 5. The iteration numbers and final parameters for different algorithms.

Algorithm	sample Size	Iteration Number	Convergent Parameters				
			μ_1	μ_2	σ_1	σ_2	$P(y_1)$
EM	1000	about 36	46.14	49.68	1.90	19.18	0.731
MM	1000	about 18	46.14	49.68	1.90	19.18	0.731
CM-EM	1000	8	46.01	49.53	2.08	21.13	0.705
Real parameters			46	50	2	20	0.7

These data show that iterations that the CM-EMneeded was less than half of iterations that the EM or MM algorithm needed.

Example 7. *A mixture of six components in two-dimensional feature space, as shown in Figure 15, were tested. The sample size was 1000. The true and initial parameters can be found in Appendix B.*



(**a**) Iterative start

(**b**) The mixture model converges after 30 iterations

Figure 15. CM4 for a two-dimensional mixture model. There are six components with Gaussian distributions.

This example was to test whether CM4 could correctly converge for seriously overlapping components. The upper two pairs of components could quickly converge, whereas convergence was slow for the lower pair. The convergence condition was that the horizontal deviation was smaller than 1.

5. Discussion

*5.1. Discussing Confirmation Measure b**

In modern times, the induction problem has become the confirmation problem [58]. Many confirmation measures have been proposed [59].

Most confirmation measures emphasize that larger $P(y_1|x_1)$ (more positive examples) is important, whereas $b_1^* = b^*(y_1 \to x_1)$ emphasizes that smaller $P(y_1|x_0)$ (fewer negative examples) is important. For example, when the sensitivity $P(y_1|x_1)$ is 0.1 and the specificity $P(y_0|x_0)$ is 1, it follows that both b_1^* and CL_1 are 1, which is reasonable. However, while using the existing confirmation formulae [59], the degrees of confirmation of MP_1 are very small.

When the sensitivity is 1, if the specificity is as small as 0.1, the degree of confirmation of MP_1, b_1^*, is also 0.1. However, while using the existing confirmation formulae, the degrees of confirmation are much bigger than 0.1. A bigger degree is unreasonable, as the ratio of negative examples is $0.9/1.9 \approx 0.47 \approx 0.5$, which means that MP_1 is almost unbelievable.

From the above two examples, we can find that the confirmation measure b^* emphasizes that no negative examples (for non-fuzzy hypotheses) or fewer negative examples (for fuzzy hypotheses) are more important than more positive examples and, hence, it is compatible with Popper's falsification thought [31,32]. This measure b^* is compatible with the confidence level and, hence, is also supported by medical practices.

Eells and Fitelson [60] suggested that Hypothesis Symmetry can be used as a standard to evaluate various conformation measures. Hypothesis Symmetry means $b^*(y_1 \to x_1) = -b^*(y_1 \to \text{not } x_1) = -b^*(y_1 \to x_0)$. We can prove that the confirmation measure b^* has Hypothesis Symmetry:

$$\begin{aligned} b^*(y_1 \to x_1) &= \frac{P(y_1|x_1) - P(y_1|x_0)}{\max(P(y_1|x_1), P(y_1|x_0))} = -\frac{P(y_1|x_0) - P(y_1|x_1)}{\max(P(y_1|x_0), P(y_1|x_1))} \\ &= -b^*(y_1 \to x_0). \end{aligned} \tag{72}$$

5.2. Discussing CM2 for the Multilabel Classification

In comparision with popular methods (such as Binary Relevance [52]) for multilabel learning, CM3 does not need n samples for n pairs of labels. It can directly obtain the semantic channel that consists of a group of truth functions from the Shannon channel $P(Y|X)$ or the sampling distribution $P(x, y)$.

In comparison with the MPP criterion, the MSI criterion can reduce the rate of failure to report smaller probability events. The MSI criterion is better when information is more important than correctness.

Note that the boundary for "Old" in Figure 10b was not 60, but 58.1. This is because "Old" has smaller logical probability than "Middle age". If the average lifespan becomes longer, the boundary for "Old" will move to the right. We can imagine that the new partitioning boundary will result in a new sampling distribution $P(x|y_5)$ and a new truth function $T(\theta_5|x)$; the new truth function will cause the boundary to move further to the right. The truth function, or the semantic meaning of "Old", should evolve with the human average lifespan, in this way.

5.3. Discussing CM3 for the MMI Classification of Unseen Instances

Solving MMI is a difficult problem, not only in machine learning [19,20], but also in the classical information theory. Shannon and many researchers [6,7] have used the least average distortion

criterion, instead of the MMI criterion, to optimize the detection and estimation. If we use the MMI criterion, the residual error coding will need a smaller average code length. Why did not they use the MMI criterion? The reason for this is that it is hard to optimize partition boundaries for MMI. However, by using CM3, we can resolve this problem (at least for low-dimensional feature spaces).

The popular methods for MMI classification or estimation use parameters to construct transition probability functions or likelihood functions and, then, optimize these parameters by using the Gradient Descent or the Newton method. The optimized parameters ascertain partition boundaries. However, CM3 or the CM iteration algorithm separately construct n likelihood functions by parameters for n different classes and then optimize the labels for different z, providing numerical solutions for partition boundaries. We compare CM3 and the Gradient Descent in Table 6.

Table 6. Comparison of the CM algorithm and Gradient Descent for low-dimensional feature spaces.

About	Gradient Descent	CM3
Models for different classes	Optimized together	Optimized separately
Boundaries is expressed by	Functions with parameters	Numerical values
For complicated boundaries	Not easy	Easy
Consider gradient and search	Necessary	Unnecessary
Convergence	Not easy	Easy
Computation	Complicated	Simple
Iterations needed	Many	2–3
Samples required	Not necessarily big	Big enough

The CM iteration algorithm has two disadvantages: One is that it requires that every sub-sample for every class is big enough, so that we can construct n likelihood functions for n classes. The other is that, for high-dimensional feature spaces, it is not feasible to label every z. We need to combine the CM iteration algorithm with neural networks for the MMI classification of high-dimensional feature spaces.

A neural network is a classifier $y = f(z)$. For a given neural network, Matching I is used to let the semantic channel match the Shannon channel to obtain reward functions $I(X; \theta_j|z)$ ($j = 0, 1,\ldots$). For given reward functions, Matching II is used to let the Shannon channel match the semantic channel to obtain new neural network parameters. Repeating these two steps will cause $I(X; \theta)$ to converge to MMI. Matching I and Matching II are similar to the tasks of the generative and discriminative models in a Generative Adversarial Network. We should be able to improve the MMI classification in high-dimensional feature spaces by combining CM3 and popular deep learning methods [33].

5.4. Discussing CM4 for mixture models

The results of Section 4.3 indicate that the complete data log-likelihood Q and the incomplete data log-likelihood $Lx(\theta)$ are not always positively correlated, as most researchers believe. In some cases, Q may (and should) decrease, as Q may be greater than $Q^* = Q(\theta^*)$, which is the true model's Q. In Example 5, while assuming the true model's parameters $\sigma_1^* = \sigma_2^* = \sigma^*$ and $P^*(y_1) = P^*(y_2) = 0.5$, we could prove that $P(y_1|x)$ and $P(y_2|x)$ were a pair of logistic functions and they became steeper as σ decreased. Hence, H increases as σ increases. We can prove that the partial derivative $\partial H/\partial\sigma$ is greater than 0. Hence, when $\theta = \theta^*$,

$$\frac{\partial Q}{\partial \sigma} = \frac{\partial L_X(\theta)}{\partial \sigma} - \frac{\partial H}{\partial \sigma} = 0 - \frac{\partial H}{\partial \sigma} < 0.$$

Therefore, we can find a small positive number Δ and replace σ^* with $\sigma^* - \Delta$, such that $Q(\sigma^* - \Delta) > Q(\sigma^*)$.

The new convergence theory, which CM4 is based on, explains that CM4 can converge, because the iteration will maximize G/R or minimize $R - G$. We have used some different examples to test the CM-EM algorithm. The experiments show that the CM-EM algorithm can reduce the slow and invalid convergence behaviors that the EM algorithm exhibits when mixture ratios are imbalanced, or local maxima of Q exist. The proposed algorithm has convergence speed that is faster than or similar to other improved EM algorithms, such as the MM algorithm [55] and the multiset EM algorithm [61].

The CM-EM algorithm can be used not only for Gaussian mixtures, but also for other mixtures. For other mixtures, the MG step is a little more difficult, but the convergence proof should be the same.

CM4 and CM3 can be used together for unsupervised learning. From CM4, we can obtain a group of model parameters from a sample with distribution $P(x)$; while using CM3, we can find the MMI classification for the sample.

The CM-EM algorithm cannot avoid θ converging to the boundary of the parameter space, however. We need to incorporate some existing algorithms, such as the Split and Merge EM algorithm [62] or the Competitive EM algorithm [63], for better global convergence properties in the mixture models.

6. Conclusions

Semantic information G theory combines the thoughts of Shannon, Popper, Fisher, Zadeh, and Carnap et al. The semantic information measure—the G measure—increases as the logical probability decreases, as well as Carnap and Bar-Hillel's semantic information measure; however, the G measure also decreases as the relative deviation increases and, hence, it can be used for the hypothesis tests.

Logical Bayesian Inference (LBI) uses the truth function or the membership function, instead of the Bayesian posterior, as the inferential tool. While using the truth function $T(\theta_j|x)$, we can make probability predictions with a different prior $P(x)$, as we use the Transition Probability Function (TPF) $P(y_j|x)$ or the Inverse Probability Function (IPF) $P(\theta_j|x)$. However, it is much easier to obtain optimized truth functions from samples than to obtain the optimized IPF, as $P(y_j)$ or $P(\theta_j)$ are not necessary for optimizing the truth functions. Importantly, the truth function can represent the semantic meaning of a hypothesis or a label and connect statistics and logic better. A windfall is that the optimization of the truth function brings a seemly reasonable confirmation measure b^* for induction.

A group of Channel Matching (CM) algorithms, CM1, CM2, CM3, and CM4, were proposed to improve machine learning, especially to resolve the Multilabel-Learning-for-New-$P(x)$ problem. CM1 can be used to improve label learning and confirmation; CM2 can be used to improve multilabel classifications; CM3 can be used to improve maximum mutual information classification of unseen instances in low-dimensional feature spaces; and, CM4 can be used to improve the mixture models. G theory and LBI have be tested by their applications to machine learning.

For further applications of G theory and LBI to machine learning, we need to combine the CM algorithms with neural networks and other algorithms in future works. Logical Bayesian Inference may be further developed for the unification of logic and statistics.

Appendix A. Abbreviations

Abbreviation	Original text
BI	Bayesian Inference
CM	Channel Matching
CM-EM	Channel Matching Expectation-Maximization
EM	Expectation-Maximization
G theory	Semantic information G theory
GPS	Global Positioning System

HIV	Human Immunodeficiency Virus
IPF	Inverse Probability Function
KL	Kullback-Leibler
LBI	Logical Bayesian Inference
LI	Likelihood Inference
MLE	Maximum Likelihood Estimation
MM	Maximum Mutual Information
MMI	Maximization-Maximization
MPP	Maximum Posterior Probability
MSI	Maximum Semantic Information
MSIE	Maximum Semantic Information Estimation
SMI	Semantic Mutual Information
SHMI	Shannon's Mutual Information
TBP	Traditional Bayes Prediction
TPF	Transition Probability Function

Appendix B.

The supplemental materials with source codes can be downloaded from http://survivor99.com/lcg/cm/forGtheory.zip. Files *.py are python 3.6 files. Parameters used in Figures 9–15 can be found in source files as shown in Table B1.

Table B1. The list of files in Supplemental Materials.

Program Name	Task
Bayes Theorem III 2.py	For Figure 9. To show label learning.
Ages-MI-classification.py	For Figure 10. To show people classification on ages using maximum semantic information criterion for given membership functions and $P(x)$.
MMI-v.py	For Figure 11. To show the Channels Matching (CM) algorithm for the maximum mutual information classifications of unseen instances. One can modify parameters or the initial partition in the program for different result.
MMI-H.py	For Figure 12.
LocationTrap3lines.py	For Figure 13. To show how the CM-EM algorithm for mixture models avoids local convergence because of the local maximum of Q.
Folder ForEx6 (with Excel file and Word readme file)	For Figure 14. To show the effect of every step of the CM-EM algorithm for mixture models.
MixModels6-2valid.py	For Figure 15. To show the CM-EM algorithm of for a two-dimensional mixture models with seriously overlapped components.

Funding: This research received no external funding.

Acknowledgments: The author thanks Peizhuang Wang for his long-term support and encouragement. The author also thanks the anonymous reviewers for their comments and suggestions.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Fisher, R.A. On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc.* **1922**, *222*, 309–368.
2. Fienberg, S.E. When Did Bayesian Inference Become "Bayesian"? *Bayesian Anal.* **2006**, *1*, 1–40.
3. Bayesian Inference. In Wikipedia: The Free Encyclopedia. Available online: https://en.wikipedia.org/wiki/Bayesian_inference (accessed on 3 March 2019).
4. Akaike, H. A new look at the statistical model identification. *IEEE Trans. Autom. Control.* **1974**, *19*, 716–723.
5. Kullback, S.; Leibler, R. On information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
6. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: New York, NY, USA, 2006.
7. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–429, 623–656.
8. Weaver, W. Recent contributions to the mathematical theory of communication. In *The Mathematical Theory of Communication*, 1st ed.; Shannon, C.E., Weaver, W., Eds; The University of Illinois Press: Urbana, IL, USA, 1963; pp. 93–117.
9. Carnap, R.; Bar-Hillel, Y. *An Outline of a Theory of Semantic Information*; Tech. Rep. No. 247; Research Laboratory of Electronics, MIT: Cambridge, MA, USA, 1952.
10. Bonnevie, E. Dretske's semantic information theory and metatheories in library and information science. *J. Doc.* **2001**, *57*, 519–534.
11. Floridi, L. Outline of a theory of strongly semantic information. *Minds Mach.* **2004**, *14*, 197–221.
12. Zhong, Y.X. A theory of semantic information. *China Commun.* **2017**, *14*, 1–17.
13. D'Alfonso, S. On Quantifying Semantic Information. *Information* **2011**, *2*, 61–101.
14. De Luca, A.; Termini, S. A definition of a non-probabilistic entropy in setting of fuzzy sets. *Inf. Control* **1972**, *20*, 301–312.
15. Bhandari, D.; Pal, N.R. Some new information measures of fuzzy sets. *Inf. Sci.* **1993**, *67*, 209–228.
16. Kumar, T.; Bajaj, R.K.; Gupta, B. On some parametric generalized measures of fuzzy information, directed divergence and information Improvement. *Int. J. Comput. Appl.* **2011**, *30*, 5–10.
17. Klir, G. Generalized information theory. *Fuzzy Sets Syst.* **1991**, *40*, 127–142.
18. Wang, Y. Generalized Information Theory: A Review and Outlook. *Inf. Technol. J.* **2011**, *10*, 461–469.
19. Belghazi, I.; Rajeswar, S.; Baratin, A.; Hjelm, R.D.; Courville, A. Mine: Mutual information neural estimation. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2018. Available online: https://arxiv.org/abs/1801.04062 (accessed on 1 January 2019).
20. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Trischler, A.; Bengio, Y. Learning Deep Representations by Mutual Information Estimation and Maximization. Available online: https://arxiv.org/abs/1808.06670 (accessed on 22 February 2019).
21. Lu, C. Shannon equations reform and applications. *BUSEFAL* **1990**, *44*, 45–52. Available online: https://www.listic.univ-smb.fr/production-scientifique/revue-busefal/version-electronique/ebusefal-44/ (accessed on 5 March 2019).
22. Lu, C. B-fuzzy quasi-Boolean algebra and a generalize mutual entropy formula. *Fuzzy Syst. Math.* **1991**, *5*, 76–80. (in Chinese)
23. Lu, C. *A Generalized Information Theory*; China Science and Technology University Press: Hefei, China, 1993; ISBN 7-312-00501-2. (in Chinese)
24. Lu, C. Meanings of generalized entropy and generalized mutual information for coding. *J. China Inst. Commun.* **1994**, *15*, 37–44. (in Chinese)
25. Lu, C. A generalization of Shannon's information theory. *Int. J. Gen. Syst.* **1999**, *28*, 453–490.
26. Lu, C. GPS information and rate-tolerance and its relationships with rate distortion and complexity distortions. *J. Chengdu Univ. Inf. Technol.* **2012**, *6*, 27–32. In Chinese.
27. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353.
28. Tarski, A. The semantic conception of truth: and the foundations of semantics. *Philos. Phenomenol. Res.* **1994**, *4*, 341–376.
29. Davidson, D. Truth and meaning. *Synthese* **1967**, *17*, 304–323.
30. Shannon, C.E. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.* **1959**, *4*, 142–163.
31. Popper, K. *The Logic of Scientific Discovery*, 1st ed.; Routledge: London, UK, 1959.
32. Popper, K. *Conjectures and Refutations*, 1st ed.; Routledge: London, UK, 2002.

33. Goodfellow, I.; Bengio, Y. *Deep Learning*, 1st ed.; The MIP Press: Cambridge, MA, USA, 2016.
34. Carnap, R. *Logical Foundations of Probability*, 1st ed.; University of Chicago Press: Chicago, IL, USA, 1950.
35. Zadeh, L.A. Probability measures of fuzzy events. *J. Math. Anal. Appl.* **1986**, *23*, 421–427.
36. Floridi, L. Semantic conceptions of information. In *Stanford Encyclopedia of Philosophy*; Stanford University: Stanford, CA, USA, 2005. Available online: http://seop.illc.uva.nl/entries/information-semantic/ (accessed on 1 July 2019).
37. Theil, H. *Economics and Information Theory*; North-Holland Pub. Co.: Amsterdam, The Netherlands; Rand McNally: Chicago, IL, USA, 1967.
38. Donsker, M.; Varadhan, S. Asymptotic evaluation of certain Markov process expectations for large time IV. *Commun. Pure Appl. Math.* **1983**, *36*, 183–212.
39. Wittgenstein, L. 1958. *Philosophical Investigations*; Basil Blackwell Ltd: Oxford, UK, 1958.
40. Bayes, T.; Price, R. An essay towards solving a problem in the doctrine of chance. *Philos. Trans. R. Soc. Lond.* **1763**, *53*, 370–418.
41. Lu, C. From Bayesian inference to logical Bayesian inference: A new mathematical frame for semantic communication and machine learning. In *Intelligence Science II, Proceedings of the ICIS2018, Beijing, China, 2 October 2018*; Shi, Z.Z., Ed.; Springer International Publishing: Cham, Switzerland, 2018; pp. 11–23.
42. Lu, C. Channels' matching algorithm for mixture models. In *Intelligence Science I, Proceedings of ICIS 2017, Beijing, China, 27 September 2017*; Shi, Z.Z., Goertel, B., Feng, J.L., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 321–332.
43. Lu, C. Semantic channel and Shannon channel mutually match and iterate for tests and estimations with maximum mutual information and maximum likelihood. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing, Shanghai, China, 15 January 2018; IEEE Computer Society Press Room: Washington, DC, USA, 2018; pp. 15–18.
44. Lu, C. Semantic channel and Shannon channel mutually match for multi-label classification. In *Intelligence Science II, Proceedings of ICIS 2018, Beijing, China, 2 October 2018*; Shi, Z.Z., Ed.; Springer International Publishing: Cham, Switzerland, 2018; pp. 37–48.
45. Dubois, D.; Prade, H. Fuzzy sets and probability: Misunderstandings, bridges and gaps. In Proceedings 1993 Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, 28 March 1993.
46. Thomas, S.F. Possibilistic uncertainty and statistical inference. In Proceedings of ORSA/TIMS Meeting, Houston, TX, USA, 11–14 October 1981.
47. Wang, P.Z. From the fuzzy statistics to the falling fandom subsets. In *Advances in Fuzzy Sets, Possibility Theory and Applications*; Wang, P.P., Ed.; Plenum Press: New York, NY, 1983; pp. 81–96.
48. Berger, T. *Rate Distortion Theory*; Prentice-Hall: Enklewood Cliffs, NJ, USA, 1971.
49. Thornbury, J.R.; Fryback, D.G.; Edwards, W. Likelihood ratios as a measure of the diagnostic usefulness of excretory urogram information. *Radiology* **1975**, *114*, 561–565.
50. OraQuick. Available online: http://www.oraquick.com/Home (accessed on 31 December 2016).
51. Zhang, M.L.; Zhou, Z.H. A review on multi-label learning algorithm. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1819–1837.
52. Zhang, M.L.; Li, Y.K.; Liu, X.Y.; Geng, X. Binary relevance for multi-label learning: An overview. *Front. Comput. Sci.* **2018**, *12*, 191–202.
53. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B* **1997**, *39*, 1–38.
54. Ueda, N.; Nakano, R. Deterministic annealing EM algorithm. *Neural Netw.* **1998**, *11*, 271–282.
55. Marin, J. -M.; Mengersen, K.; Robert, C.P. Bayesian modelling and inference on mixtures of distributions. In *Handbook of Statistics: Bayesian Thinking, Modeling and Computation*; Dey, D., Rao, C.R., Eds.; Elsevier: Amsterdam, The Netherlands, 2011; pp. 459–507.
56. Neal, R.; Hinton, G. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*; Michael, I.J., Ed.; MIT Press: Cambridge, MA, USA, 1999; pp. 355–368.
57. Lu, C. From the EM Algorithm to the CM-EM Algorithm for Global Convergence of Mixture Models. Available online: https://arxiv.org/abs/18 (accessed on 26 October 2018).
58. James, H. Inductive logic. In *The Stanford Encyclopedia of Philosophy*; Spring 2018 Ed.; Edward, N.Z., Ed.; Stanford University Press: Palo Alto, CA, USA, 2018. https://plato.stanford.edu/archives/spr2018/entries/logic-inductive/ (uploaded on 19 March 2018).

59. Tentori, K.; Crupi, V.; Bonini, N.; Osherson, D. Comparison of confirmation measures. *Cognition* **2007**, *103*, 107–119.
60. Ellery, E.; Fitelson. B. Measuring confirmation and evidence. *J. Philos.* **2000**, *97*, 663–672.
61. Huang, W.H.; Chen, Y.G. The multiset EM algorithm. *Stat. Probab. Lett.* **2017**, *126*, 41–48.
62. Ueda, N.; Nakano, R.; Ghahramani, Z.; Hinton, G.E. SMEM algorithm for mixture models. *Neural Comput.* **2000**, *12*, 2109–2128, doi:10.1162/089976600300015088.
63. Zhang, B.; Zhang, C.; Yi, X. Competitive EM algorithm for finite mixture models. *Pattern Recognit.* **2004**, *37*, 131–144.