Inference of Abstraction for Human-like Probabilistic Reasoning

Hiroyuki Kido[0000−0002−7622−4428]

Cardiff University, Cardiff CF10 3AT, UK
KidoH@cardiff.ac.uk

Abstract. Inspired by Bayesian approaches to brain function in neuroscience, we give a simple theory of human-like probabilistic reasoning. We simply model how data cause symbolic knowledge in terms of its satisfiability in formal logic. The underlying idea is that reasoning is a process of deriving symbolic knowledge from data by abstraction, i.e., selective ignorance. The theory does not impose the assumption of independence or conditional independence of symbolic knowledge, an unrealistic but necessary assumption of Bayesian networks and their variants. The theory is empirically justified by its digit prediction and image generation performance on the MNIST dataset.

Keywords: Artificial intelligence · Cognition · Probability theory · Machine learning · Reasoning and learning · Generative reasoning

1 Introduction

Bayes' theorem plays an important role today in AI, neuroscience and cognitive science. It underlies most modern approaches to uncertain reasoning in AI systems [21]. Neuroscience often uses it as a metaphor for functions of the cerebral cortex, the outer portion of the brain in charge of higher-order cognitive functions such as perception, memory, emotion and thought [13, 10, 8, 3, 7]. It relates various brain theories such as Bayesian coding hypothesis [10], free-energy principle [6] and predictive coding [19]. Their common idea is that the biological brain can be seen as a probabilistic generative model by which the past experience of the brain is constantly, but unconsciously, used to predict what is likely to happen outside the brain [6, 22, 9].

The success of Bayesian approaches in AI and neuroscience makes us think that there is a common Bayesian account of reasoning and learning, especially entailment and prediction, the main concern of formal logic and machine learning, respectively. The idea is worth investigating as it may give a clue to think upon how reasoning and learning operate in the human brain. Additionally, finding a principle underlying reasoning and learning is an open problem in AI across different disciplines, e.g., neuro-symbolic AI [1]. Despite the scientific importance, few research in AI has focused on a Bayesian approach to a computational model of reasoning and learning. Indeed, most present research papers, e.g., [14–16, 5, 20, 24, 26, 23, 2], study how to combine existing reasoning and learning methods

with the assumption that they are intrinsically different. For example, maximum likelihood estimation is the method most often used to learn the probability or weight of symbolic knowledge. Logical semantics is then used to draw conclusions from the probabilistic or weighted symbolic knowledge. Various types of logical semantics exist such as the semantics of Bayesian networks [15], Markov logic networks [20] and distribution semantics [24]. However, the method used for learning cannot be used for reasoning, and vice versa. Moreover, in computational cognitive science, the theory-based Bayesian models of induction [25], the learned inference model [4] and the Bayesian program learning framework [11, 12] rest on the idea that observable data and their variants are generated from more abstract hypotheses such as background knowledge and principles about the world. Although the idea is prevalent in the machine learning community, the idea eventually struggles with intractable computation associated with an exponentially growing hypothesis space, especially when trying to incorporate symbolic knowledge.

In this paper, we argue that some important aspects of reasoning and learning can be unified by inference of abstraction as selective ignorance. The simple idea underlying the inference of abstraction is that intrinsically abstract symbolic knowledge should be derived from intrinsically concrete data via inference. The idea is simply formalised as a probabilistic model of the causality that data determine states of the world, and the states of the world determine the truth value of symbolic knowledge. The idea opposes existing work such as [25, 4, 11, 12], since we argue that abstract hypotheses and knowledge are generated from observable data by abstraction.

We discuss three important perspectives on reasoning and learning. First, knowledge is intrinsically abstract whereas data are intrinsically concrete. The inference of abstraction derives symbolic knowledge from data. The natural view and approach contrast rules of inference and the semantics of Bayesian networks deriving knowledge from another knowledge. Second, this paper looks at how symbolic knowledge can be derived from data. This contrasts the machine learning approach looking at how data can be derived from parameters characterising the data, e.g., the mean and variance of a normal distribution. Third, the inference of abstraction comprises an interpretation and inverse interpretation of formal logic. The inference can be seen as a realisation of top-down and bottom-up processing often used in neuroscience as a metaphor for the information processing of the brain.

The contributions of this paper are summarised as follows. First, this paper results in a new machine learning method that significantly generalises a sort of k-nearest neighbour method. The method empirically outperforms a k-nearest neighbour method in AUC on the MNIST dataset. Second, this paper bridges probability theory and machine learning in a novel way that a sort of k-nearest neighbour method can be seen as probabilistic reasoning.

This paper is organised as follows. In Section 2, we define a generative reasoning model for inference of abstraction. Section 3 discusses its probabilistic correctness. We summarise our results in Section 4.

2 Inference of Abstraction

Let $\{d_1, d_2, ..., d_K\}$ be a multiset of K data. D denotes a random variable of data whose values are all the elements of $\{d_1, d_2, ..., d_K\}$. For all data $d_k (1 \leq k \leq K)$, we define the probability of d_k, denoted by $p(D = d_k)$, as follows.

$$p(D = d_k) = \frac{1}{K}$$

L represents a propositional language for simplicity. Let $\{m_1, m_2, ..., m_N\}$ be the set of models of L. A model is an assignment of truth values to all the atomic formulas in L. Intuitively, each model represents a different state of the world. We assume that each data d_k supports a single model. We thus use a function m, $\{d_1, d_2, ..., d_K\} \rightarrow \{m_1, m_2, ..., m_N\}$, to map each data to the model supported by the data. M denotes a random variable of models whose realisations are all the elements of $\{m_1, m_2, ..., m_N\}$. For all models $m_n (1 \leq n \leq N)$, we define the probability of m_n given d_k, denoted by $p(M = m_n | D = d_k)$, as follows.

$$p(M = m_n | D = d_k) = \begin{cases} 1 & \text{if } m_n = m(d_k) \\ 0 & \text{otherwise} \end{cases}$$

The truth value of a propositional formula and first-order closed formula in classical logic is uniquely determined in a state of the world specified by a model of a language. Let α be a formula in L. We assume that α is a random variable whose realisations are 0 and 1 meaning false and true respectively. We use symbol $[\![\alpha]\!]$ to refer to the models of α. Namely, $[\![\alpha = 1]\!]$ and $[\![\alpha = 0]\!]$ represent the set of models in which α is true and false, respectively. Let $\mu \in [0, 1]$ be a variable, not a random variable. For all formulas $\alpha \in L$, we define the probability of each truth value of α given m_n, denoted by $p(\alpha | M = m_n)$, as follows.

$$p(\alpha = 1 | M = m_n) = \begin{cases} \mu & \text{if } m_n \in [\![\alpha = 1]\!] \\ 1 - \mu & \text{otherwise} \end{cases}$$

$$p(\alpha = 0 | M = m_n) = \begin{cases} \mu & \text{if } m_n \in [\![\alpha = 0]\!] \\ 1 - \mu & \text{otherwise} \end{cases}$$

Let $[\![\alpha]\!]_{m_n}$ be a function such that $[\![\alpha]\!]_{m_n} = 1$ if $m_n \in [\![\alpha]\!]$ and $[\![\alpha]\!]_{m_n} = 0$ otherwise. The above expressions can be simply written as a Bernoulli distribution with parameter $\mu \in [0, 1]$.

$$p(\alpha | M = m_n) = \mu^{[\![\alpha]\!]_{m_n}} (1 - \mu)^{1 - [\![\alpha]\!]_{m_n}}$$

Here, the variable $\mu \in [0, 1]$ plays an important role to relate formal logic to machine learning. We will see that $\mu = 1$ relates to Bayesian networks. We also see that $\mu \rightarrow 1$ relates to an all-nearest neighbour method, a generalisation of a sort of the k-nearest neighbour method in machine learning. Additionally, $\mu < 1$ relates to a smoothed or weighted version of the all-nearest neighbour method. They are all discussed in the next section.

In classical logic, given a model, the truth value of a formula does not change the truth value of another formula. Thus, in probability theory, the truth value of a formula α_1 is conditionally independent of the truth value of another formula α_2 given a model M, i.e., $p(\alpha_1|\alpha_2, M, D) = p(\alpha_1|M, D)$ or equivalently $p(\alpha_1, \alpha_2|M, D) = p(\alpha_1|M, D)p(\alpha_2|M, D)$. Let $\Gamma \subseteq L$ be a finite theory of L. We therefore have

$$p(L|M, D) = \prod_{\alpha \in L} p(\alpha|M, D). \tag{1}$$

Moreover, in classical logic, the truth value of a formula depends on models but not data. Thus, in probability theory, the truth value of a formula α is conditionally independent of data D given a model M, i.e., $p(\alpha|M, D) = p(\alpha|M)$. We thus have

$$\prod_{\alpha \in \Gamma} p(\alpha|M, D) = \prod_{\alpha \in \Gamma} p(\alpha|M). \tag{2}$$

Therefore, the full joint distribution, $p(\Gamma, M, D)$, can be written as follows.

$$p(\Gamma, M, D) = p(\Gamma|M, D)p(M|D)p(D) = \prod_{\alpha \in \Gamma} p(\alpha|M)p(M|D)p(D) \tag{3}$$

Here, the product rule (or chain rule) of probability theory is applied in the first equation, and Equations (1) and (2) in the second equation. As will be seen later, the joint distribution $p(\Gamma, M, D)$ is a probabilistic model of symbolic reasoning from data. We call the joint distribution a generative reasoning model for short. We often represent $p(\Gamma, M, D)$ as $p(\Gamma, M, D; \mu)$ if our discussion is relevant to μ. We use symbol ';' to represent that μ is a variable, but not a random variable. In this paper, we assume a finite number of realisations of each random variable.

The full joint distribution implies that we can no longer discuss only the probabilities of individual formulas, but they are derived from data. For example, the probability of $\alpha \in \Gamma$ is calculated as follows.

$$p(\alpha) = \sum_m \sum_d p(\alpha, m, d) = \sum_m p(\alpha|m) \sum_d p(m|d)p(d) \tag{4}$$

Here, the sum rule of probability theory is applied in the first equation, and Equation (3) in the second equation.

Proposition 1. *Let $p(\Gamma, M, D; \mu)$ be a generative reasoning model. For all $\alpha \in \Gamma$, $p(\alpha = 0) = p(\neg\alpha = 1)$ holds.*

Proof. For all models m, α is false in m if and only if $\neg\alpha$ is true in m. Thus, $[\![\alpha = 0]\!] = [\![\neg\alpha = 1]\!]$ is the case. Therefore,

$$p(\alpha = 0) = \textstyle\sum_m p(\alpha = 0|m)p(m) = \sum_m \mu^{[\![\alpha=0]\!]_m}(1-\mu)^{1-[\![\alpha=0]\!]_m}p(m)$$
$$= \textstyle\sum_m \mu^{[\![\neg\alpha=1]\!]_m}(1-\mu)^{1-[\![\neg\alpha=1]\!]_m}p(m) = \sum_m p(\neg\alpha = 1|m)p(m) = p(\neg\alpha = 1).$$

This holds regardless of the value of μ. □

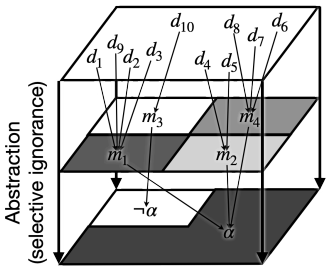


Fig. 1. A schematic of how the probability distribution over data determines the probability distribution over logical formulas. For simplicity, an arrow is omitted if the formula at the end of the arrow is false in the model at the start of the arrow and if the model at the end of the arrow is not supported by the data at the start of the arrow.

Hence, we replace $\alpha = 0$ by $\neg\alpha = 1$ and abbreviate $\neg\alpha = 1$ to $\neg\alpha$. We also abbreviate $M = m_n$ to m_n and $D = d_k$ to d_k.

The hierarchy shown in Figure 1 illustrates Equation (4). The top layer of the hierarchy is a probability distribution over data, the middle layer is a probability distribution over states of the world, often referred to as models in formal logic, and the bottom layer is a probability distribution over a logical formula α. A darker colour indicates a higher probability. Each element of a lower layer is an abstraction, i.e., selective ignorance, of the linked element of the upper layer.

Example 1. Let L be a propositional language built with two symbols, *rain* and *wet*, meaning 'rain falls' and 'the road gets wet,' respectively. Let $m_n(1 \leq n \leq 4)$ be the models of L and $d_k(1 \leq k \leq 10)$ be data about rain and road conditions. Table 1 shows which data support which models and which models specify which states of the world. The probability of $rain \rightarrow wet$ can be calculated using Equation (4) as follows.

$$
\begin{aligned}
p(rain \rightarrow wet) &= \sum_{n=1}^{4} p(rain \rightarrow wet|m_n) \sum_{k=1}^{10} p(m_n|d_k)p(d_k) \\
&= \mu \sum_{k=1}^{10} p(m_1|d_k)\frac{1}{10} + \mu \sum_{k=1}^{10} p(m_2|d_k)\frac{1}{10} + (1-\mu) \sum_{k=1}^{10} p(m_3|d_k)\frac{1}{10} \\
&+ \mu \sum_{k=1}^{10} p(m_4|d_k)\frac{1}{10} = \frac{4}{10}\mu + \frac{2}{10}\mu + \frac{1}{10}(1-\mu) + \frac{3}{10}\mu = \frac{1}{10} + \frac{8}{10}\mu
\end{aligned}
$$

Therefore, $p(rain \rightarrow wet) = 9/10$ when $\mu = 1$ or $\mu \rightarrow 1$, i.e., μ approaching 1. Figure 1 illustrates the calculation and visualises how the probability of $rain \rightarrow wet$, denoted by α in the figure, is derived from data.

Proposition 2 (Maximum likelihood estimation). *Let $p(\Gamma, M, D)$ be a generative reasoning model. $p(M)$ is equivalent to maximum likelihood estimation.*

Table 1. An example of Figure 1. From the left, each column show data, models and the likelihood of the formula.

D	M	$rain$	wet	$p(rain \to wet\|M)$
d_1, d_2, d_3, d_9	m_1	0	0	μ
d_4, d_5	m_2	0	1	μ
d_{10}	m_3	1	0	$1-\mu$
d_6, d_7, d_8	m_4	1	1	μ

Proof. For all n, $p(m_n)$ can be simply derived from $p(\Gamma, M, D)$ as follows.

$$p(m_n) = \sum_d p(m_n|d)p(d) = \frac{1}{K}\sum_d p(m_n|d) = \frac{K_n}{K}$$

Here, K_n is the number of data supporting the nth model. We show that this is a maximum likelihood estimate. In statistics, data are assumed to be generated from a probability distribution. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_N)$ be the parameter of a categorical distribution generating our data $d_1, d_2, ..., d_K$. The maximum likelihood estimation is defined as follows.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(d_1, d_2, ..., d_K|\boldsymbol{\theta})$$

It is common to assume that data are generated independently from the same distribution, i.e., i.i.d. We thus have

$$p(d_1, d_2, ..., d_K|\boldsymbol{\theta}) = \prod_{k=1}^{K} p(d_k|\boldsymbol{\theta}) = \theta_1^{K_1}\theta_2^{K_2}\cdots\theta_{N-1}^{K_{N-1}}(1-\theta_1-\theta_2-\cdots-\theta_{N-1})^{K_N},$$

where K_n is the number of data in the nth category and $\theta_N = 1-\theta_1-\theta_2-\cdots-\theta_{N-1}$. $\boldsymbol{\theta}$ maximises the likelihoods if and only if it maximises their log likelihoods given as follows.

$$L(\boldsymbol{\theta}) = K_1\log\theta_1 + K_2\log\theta_2 + \cdots + K_{N-1}\log\theta_{N-1} + K_N\log(1-\theta_1-\cdots-\theta_{N-1})$$

To find $\boldsymbol{\theta}$ maximising L, we differentiate L with respect to θ_n and set the resulting expression to zero, for all $n(1 \le n \le N-1)$. We then have

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta n} = \frac{K_n}{\theta_n} - \frac{K_N}{1-\theta_1-\theta_2-\cdots-\theta_{N-1}} = 0.$$

It causes the simultaneous equations with the following matrix representation.

$$\begin{pmatrix} K_1+K_N & K_1 & \dots & K_1 \\ K_2 & K_2+K_N & \dots & K_2 \\ \vdots & \vdots & \ddots & \vdots \\ K_{N-1} & K_{N-1} & \dots & K_{N-1}+K_N \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{N-1} \end{pmatrix} = \begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_{N-1} \end{pmatrix}$$

The solution to the simultaneous equations can be given as follows.

$$\boldsymbol{\theta} = \left(\frac{K_1}{K}, \frac{K_2}{K}, ..., \frac{K_N}{K}\right)$$

□

3 Correctness

3.1 Bayesian networks

Let X_i (for $i = 1, 2, 3$) represent three binary random variables corresponding to the propositions 'it is raining outside', 'the grass is wet', and 'the outside temperature is high', respectively. The lower case of each random variable represents its realisation. What one needs in most cases is a posterior probability. For example, $p(x_1|x_3)$ can be represented as follows.

$$p(x_1|x_3) = \frac{p(x_1, x_3)}{p(x_3)} = \frac{\sum_{x_2} p(x_1, x_2, x_3)}{\sum_{x_1,x_2} p(x_1, x_2, x_3)}$$

This equation shows that the full joint distribution is required for the exact posterior probability. The space complexity of the full joint distribution is $O(2^N)$ where N is the number of propositions. Thus, the calculation of a posterior probability is generally intractable. [17] tackled this issue by incorporating the idea of independence. For example, if X_3 is assumed to be conditionally independent of X_2 given X_1, the above equation can be simplified as follows (see Figure 2).

$$p(x_1|x_3) = \frac{\sum_{x_2} p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)}{\sum_{x_1,x_2} p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)} = \frac{\sum_{x_2} p(x_3|x_1)p(x_2|x_1)p(x_1)}{\sum_{x_1,x_2} p(x_3|x_1)p(x_2|x_1)p(x_1)}$$

The assumption of independence reduces a space complexity. However, those who strictly adhere to data should not accept the assumption. This is because the assumption rarely holds in reality without a modification of original data or resort to expert knowledge.

Now, let $p(L, M, D; \mu)$ be a generative reasoning model where L is built with X_i (for $i = 1, 2, 3$). The posterior probability $p(x_1|x_3)$ can be naively represented as follows (see the leftmost graph in Figure 3).

$$\begin{aligned} p(x_1|x_3) &= \frac{p(x_1, x_3)}{p(x_3)} = \frac{\sum_{x_2,m,d} p(x_1, x_2, x_3, m, d)}{\sum_{x_1,x_2,m,d} p(x_1, x_2, x_3, m, d)} \\ &= \frac{\sum_{x_2,m,d} p(x_3|x_2, x_1, m, d)p(x_2|x_1, m, d)p(x_1|m, d)p(m|d)p(d)}{\sum_{x_1,x_2,m,d} p(x_3|x_2, x_1, m, d)p(x_2|x_1, m, d)p(x_1|m, d)p(m|d)p(d)} \end{aligned}$$

From Equations (1) and (2), the above equation can be simplified as follows (see the second and third graphs).

$$= \frac{\sum_{x_2,m,d} p(x_3|m)p(x_2|m)p(x_1|m)p(m|d)p(d)}{\sum_{x_1,x_2,m,d} p(x_3|m)p(x_2|m)p(x_1|m)p(m|d)p(d)}$$

Each datum has an equal probability, and it supports a single model, i.e., $p(m|d) = 1$ if $m = m(d)$ and $p(m|d) = 0$ otherwise. Thus, the above equation can be simplified as follows (see the rightmost graph).

$$= \frac{\sum_{x_2,d} p(x_3|m(d))p(x_2|m(d))p(x_1|m(d))}{\sum_{x_1,x_2,d} p(x_3|m(d))p(x_2|m(d))p(x_1|m(d))} = \frac{\sum_d p(x_3|m(d))p(x_1|m(d))}{\sum_d p(x_3|m(d))}$$

Fig. 2. The left can fit with any full joint distribution. The right can fit with any full joint distribution with the conditional independence, $p(X_3|X_2, X_1) = p(X_3|X_1)$, that rarely holds without data modification.

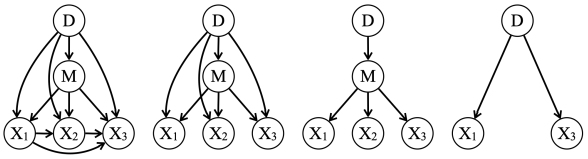


Fig. 3. The rightmost structure can be derived from the leftmost one using the properties of formal logic and the natural assumption that each data supports a single model.

The correctness of the outcome can be generally described as follows. Suppose $\alpha \in L$ and $\Delta \subseteq L$ for a generative reasoning model $p(L, M, D; \mu = 1)$. Since $1^1 0^0 = 1$ and $1^0 0^1 = 0$, we have

$$p(\alpha|\Delta) = \frac{\sum_m p(\alpha|m)p(\Delta|m)p(m)}{\sum_m p(\Delta|m)p(m)} = \frac{\sum_m 1^{[\![\alpha]\!]_m} 0^{1-[\![\alpha]\!]_m} 1^{[\![\Delta]\!]_m} 0^{1-[\![\Delta]\!]_m} p(m)}{\sum_m 1^{[\![\Delta]\!]_m} 0^{1-[\![\Delta]\!]_m} p(m)}$$
$$= \frac{\sum_m [\![\alpha]\!]_m [\![\Delta]\!]_m p(m)}{\sum_m [\![\Delta]\!]_m p(m)} = \frac{\sum_{m \in [\![\alpha]\!] \cap [\![\Delta]\!]} p(m)}{\sum_{m \in [\![\Delta]\!]} p(m)}$$

Here, $p(m)$ is a maximum likelihood estimate (MLE) as shown in Proposition 2. Therefore, the denominator (resp. numerator) is the sum of the MLEs of the probabilities of the models satisfying Δ (resp. α and Δ).

3.2 Nearest neighbour methods

The MNIST dataset contains 70,000 images (60,000 training and 10,000 test images) of handwritten digits from 0 to 9. Each image comprises $28 \times 28 (= 784)$ pixels in width×height. Each pixel has a greyscale from 0 to 255 representing pure black and white colours, respectively. We look at two machine learning tasks on MNIST: digit prediction and image generation.

Digit prediction Consider a generative reasoning model $p(\Gamma, M, D; \mu)$ where Γ is built with propositional symbols $digit_i (0 \leq i \leq 9)$ and $pixel_j (1 \leq j \leq 28 \times 28)$ (dig_i and pix_j for short), where $digit_i$ represents that an image is of digit i, and $pixel_j$ that the greyscale of the jth pixel of an image is above the threshold of 30. All the ten digit variables and 28×28 pixel variables can take the two states, true of false. L thus has $2^{10+28\times28}$ models in total, and each of the models is a value of the random variable M. Each training image is a value of the random variable D. We use the following fact in the machine learning context.

Proposition 3. *Let $p(\Gamma,\ M,\ D; \mu \in (0.5, 1))$ be a generative reasoning model, and $\alpha \in \Gamma$ and $\Delta \subseteq \Gamma$.*

$$p(\alpha|\Delta) = \frac{\sum_d p(\alpha|d) \prod_{\beta\in\Delta} p(\beta|d)}{\sum_d \prod_{\beta\in\Delta} p(\beta|d)}$$

Proof. For all $\gamma \in \Gamma$ and data d, we have

$$p(\gamma|d) = \frac{\sum_m p(\gamma, m, d)}{p(d)} = \frac{\sum_m p(\gamma|m)p(m|d)p(d)}{p(d)}$$
$$= \frac{p(\gamma|m(d))\cancel{p(d)}}{\cancel{p(d)}} = \mu^{[\![\gamma]\!]_{m(d)}}(1-\mu)^{[\![\gamma]\!]_{m(d)}}.$$

Since $\mu \notin \{0, 1\}$, $p(\gamma|d) \neq 0$. We also have

$$p(\alpha|\Delta) = \frac{\sum_d \sum_m p(\alpha|m) \prod_{\beta\in\Delta} p(\beta|m)p(m|d)p(d)}{\sum_d \sum_m \prod_{\beta\in\Delta} p(\beta|m)p(m|d)p(d)}$$
$$= \frac{\sum_d p(\alpha|m(d)) \prod_{\beta\in\Delta} p(\beta|m(d))}{\sum_d \prod_{\beta\in\Delta} p(\beta|m(d))} = \frac{\sum_d p(\alpha|d) \prod_{\beta\in\Delta} p(\beta|d)}{\sum_d \prod_{\beta\in\Delta} p(\beta|d)}.$$

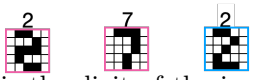
Since $\mu \notin \{0, 1\}$, this does not cause division by zero. □

For digit prediction, we first look at the generative reasoning model $p(\Gamma,\ M,\ D; \mu \to 1)$ where $\mu \to 1$ represents that μ approaches one, i.e., $\lim_{\mu\to 1}$. Given all the 60k training images, we use the following instance of Proposition 3.

$$p(Digit_i|Pixel_1, ..., Pixel_{28\times 28}) = \frac{\sum_{k=1}^{60k} p(Digit_i|d_k) \prod_{j=1}^{28\times 28} p(Pixel_j|d_k)}{\sum_{k=1}^{60k} \prod_{j=1}^{28\times 28} p(Pixel_j|d_k)} \quad (5)$$

Here, we capitalised the propositional symbols so that it is clear that they are not formulas being true, e.g., $digit_i = 1$, but random variables without observed values.

Example 2 (Digit prediction with $p(L, M, D; \mu \to 1)$). Let L be built with propositional symbols $digit_i (0 \leq i \leq 9)$ and $pixel_j (1 \leq j \leq 5 \times 5)$. Let the following two 5×5-pixel images with the purple borders be training images and the following one 5×5-pixel image with the blue border be a test image.



The label of each image is the digit of the image. Each 5×5-pixel training image with its digit instantiates the random variable D. Equation (5) can then be instantiated as follows, where $\boldsymbol{Pixel} = (Pixel_1, ..., Pixel_{5\times 5})$.

$$p(Dig_i|\boldsymbol{Pixel}) = \frac{\sum_{k=1}^{2} p(Dig_i|d_k) \prod_{j=1}^{5\times 5} p(Pix_j|d_k)}{\sum_{k=1}^{2} \prod_{j=1}^{5\times 5} p(Pix_j|d_k)}$$
$$= \frac{p(Dig_i|\boxed{2}) \prod_{j=1}^{5\times 5} p(Pix_j|\boxed{2}) + p(Dig_i|\boxed{7}) \prod_{j=1}^{5\times 5} p(Pix_j|\boxed{7})}{\prod_{j=1}^{5\times 5} p(Pix_j|\boxed{2}) + \prod_{j=1}^{5\times 5} p(Pix_j|\boxed{7})} \quad (6)$$

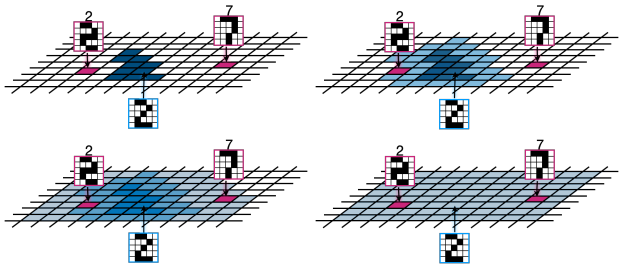


Fig. 4. Each cell of the grid is a model of L. The training and test images are shown above and below the grid, respectively. The blue cells on the top-left grid show that the prediction fails with $\mu = 1$, since no training image is found in the models of the test image. The light blue cells on the top-right grid show that the prediction succeeds with $\mu \to 1$, since the limit expands the models of the test image until its best matched training image is found. The bottom left and right grids illustrate $\mu \in (0.5, 1)$ and $\mu = 0.5$, respectively.

The map m from each training image to a model of L is obvious. We have the following likelihoods, where j indexes pixels from left to right and top to bottom.

$$p(Dig_i = 1|\text{[2]}) = \begin{cases} \mu & \text{if } i = 2 \\ 1-\mu & \text{otherwise} \end{cases} \quad p(Dig_i = 1|\text{[7]}) = \begin{cases} \mu & \text{if } i = 7 \\ 1-\mu & \text{otherwise} \end{cases}$$

$$p(Pix_j = 1|\text{[2]}) = \begin{cases} \mu & \text{if } j \in \{1, 4\text{–}8, 10, 11, 15, 16, 18\text{–}22, 25\} \\ 1-\mu & \text{otherwise} \end{cases}$$

$$p(Pix_j = 1|\text{[7]}) = \begin{cases} \mu & \text{if } j \in \{1, 5\text{–}8, 10\text{–}13, 15\text{–}17, 19\text{–}22, 24, 25\} \\ 1-\mu & \text{otherwise} \end{cases}$$

From the test image, we have

$$pixel_j = \begin{cases} 1 & \text{if } j \in \{1, 4\text{–}8, 10\text{–}12, 14\text{–}16, 18\text{–}21, 25\} \\ 0 & \text{otherwise, i.e., } j \in \{2, 3, 9, 13, 17, 22\text{–}24\}. \end{cases}$$

Let ***pixel***, abbreviated to ***pix***, denote ($Pixel_1 = pixel_1$, $Pixel_2 = pixel_2$, ..., $Pixel_{5\times5} = pixel_{5\times5}$). Equation (6) can then be instantiated as follows.

$$p(Dig_i = 1|\boldsymbol{pix}) = \frac{p(Dig_i = 1|\text{[2]})X_1 + p(Dig_i = 1|\text{[7]})X_2}{X_1 + X_2}$$

$$= \begin{cases} \dfrac{\mu^{23}(1-\mu)^3 + \mu^{18}(1-\mu)^8}{\mu^{22}(1-\mu)^3 + \mu^{18}(1-\mu)^7} & \text{if } i = 2 \quad (7) \\ \dfrac{\mu^{22}(1-\mu)^4 + \mu^{19}(1-\mu)^7}{\mu^{22}(1-\mu)^3 + \mu^{18}(1-\mu)^7} & \text{if } i = 7 \quad (8) \\ \dfrac{\mu^{22}(1-\mu)^4 + \mu^{18}(1-\mu)^8}{\mu^{22}(1-\mu)^3 + \mu^{18}(1-\mu)^7} & \text{otherwise} \quad (9) \end{cases}$$

Here, X_1 and X_2 were calculated as follows.

$$X_1 = \prod_{j=1}^{5\times5} p(Pixel_j = pixel_j|\text{  }) = \mu^{22}(1-\mu)^3$$

$$X_2 = \prod_{j=1}^{5\times5} p(Pixel_j = pixel_j|\text{  }) = \mu^{18}(1-\mu)^7$$

Given $\mu \to 1$, Equations (7), (8) and (9) thus turn out to be

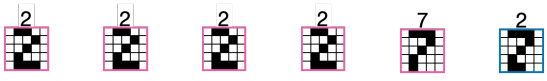
$$\begin{cases} \lim\limits_{\mu\to1} \dfrac{\mu^5 + (1-\mu)^5}{\mu^4 + (1-\mu)^4} = \dfrac{1}{1} = 1 & \text{if } i = 2 \quad (10) \\ \lim\limits_{\mu\to1} \dfrac{\mu^4(1-\mu) + \mu(1-\mu)^4}{\mu^4 + (1-\mu)^4} = \dfrac{0}{1} = 0 & \text{if } i = 7 \quad (11) \\ \lim\limits_{\mu\to1} \dfrac{\mu^4(1-\mu) + (1-\mu)^5}{\mu^4 + (1-\mu)^4} = \dfrac{0}{1} = 0 & \text{otherwise.} \quad (12) \end{cases}$$

Figure 4 illustrates the digit prediction with different μ values. It shows a reasonable role of the limit used in Equations (10), (11) and (12). The limit allows us to cancel out $(1-\mu)^3$ from the equations. Here, $(1-\mu)$ represents a mismatch between the test image and the training image, and thus, $(1-\mu)^3$ represents a mismatch between the test image and the training image with the best match for the test image. The limit thus subtracts the mismatch from all the training images. As a result, the digit of the given image turns out to be the digit of its best matched training image.

As shown in Equations (10), (11) and (12), the denominator turns out to be the number of training images whose pixel values are maximally the same as $Pixel_1, ..., Pixel_{28\times28}$, the pixel values of a test image. Amongst them, the numerator turns out to be the number of training images whose digit values are the same as $Digit_i$, the digit value of the test image. As a result, the above conditional probability can be seen as an all-nearest neighbours method, which generalises the k-nearest neighbours (kNN) method classifying test data by a majority vote from the k nearest training data. This is a reasonable solution to a well-known problem that it is often difficult to settle an appropriate value of k for kNN methods. Moreover, the search for the nearest neighbours and the use of them in prediction are given a unified computational account by Equation (5).

In the machine learning context, we until now saw generative reasoning models $p(L, M, D; \mu \to 1)$ as a sort of an all-nearest neighbours method. We will next see generative reasoning models $p(L, M, D; \mu \in (0.5, 1))$ as a smoothed or weighted version of the all-nearest neighbours method.

Example 3 (Digit prediction with $p(L, M, D; \mu \in (0.5, 1))$ (Continued)). Consider the following five 5×5-pixel training images and one 5×5-pixel test image with the labels of their digits.



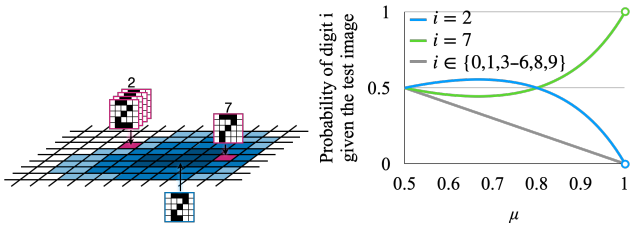


Fig. 5. The prediction fails with $\mu \to 1$, since the test image and its nearest training image have different digits (see the medium blue cells). It can succeed with $\mu \in (0.5, 1)$, since the models of the test image is expanded beyond its nearest training image for its second and further nearest training images (see the light blue cells). The curves on the right show the values of Expressions (13), (14) and (15).

Going through the same process we discussed in Example 2, we can now instantiate Equation (5) as follows.

$$p(Dig_i = 1|\boldsymbol{pix}) = \begin{cases} \dfrac{5\mu(1-\mu)}{4(1-\mu)+\mu} & \text{if i=2} \quad (13) \\ \dfrac{4(1-\mu)^2+\mu^2}{4(1-\mu)+\mu} & \text{if i=7} \quad (14) \\ \dfrac{4(1-\mu)^2+\mu(1-\mu)}{4(1-\mu)+\mu} & \text{otherwise} \quad (15) \end{cases}$$

Given $\mu \to 1$, each equation turns out to be 0, 1 and 0, respectively, which are all reasonable as the test image and its best matched training image have different digits. However, given $\mu \in (0.5, 0.8)$, the probability of the digit being two is equal or larger than the probability of the digit being seven (see the curves in Figure 5). This is also reasonable as the test image and all of the relatively large number of its second matched training images have the same digits. Here, the qualitative effect of the single best match for the test image is suppressed by the quantitative effect of the multiple second match. As shown in Figure 5, μ functions to balance the effects of matching quality and quantity.

Figure 6 shows the learning curves generated by Equation (5) using the real MNIST dataset. The baseline is given by the kNN method with different k values. We use AUC, the area under ROC (receiver operating characteristic) curve, for performance evaluation, since the generative reasoning model returns probabilistic outputs. $\mu \to 1$ experiences overfitting, since the number of the training images best matched for each test image is relatively too small to discard anomalies. This is similar to the 1NN method where only one nearest neighbour training image is used in prediction.

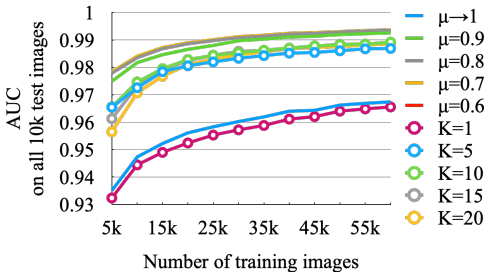


Fig. 6. The learning curves of the generative reasoning model with different μ values. The baseline is given by the kNN method built using the 'KNeighborsClassifier' function [18] with default setting, i.e., the 'uniform' weights and 'auto' algorithm. The training images were extracted from the beginning.

Image generation Figure 7 shows the images generated from each digit using the following equation, for all $j(1 \leq j \leq 28 \times 28)$.

$$p(Pixel_j|Digit_i) = \frac{\sum_{k=1}^{70k} p(Digit_i|d_k)p(Pixel_j|d_k)}{\sum_{k=1}^{70k} p(Digit_i|d_k)}$$

Each image can be seen as the average of all the images of the same digit. The pixel value of each pixel is the average of the all 70k images. Figure 8 shows the entire test images generated from their partial pixel values using the following equation, for all $j(I < j \leq 28 \times 28)$.

$$p(Pix_j|Pix_1, ..., Pix_I) = \frac{\sum_{k=1}^{60k} p(Pix_j|d_k) \prod_{i=1}^{I} p(Pix_i|d_k)}{\sum_{k=1}^{60k} \prod_{i=1}^{I} p(Pix_i|d_k)}$$

Figure 8 shows that, given the upper black-background area extracted from a test image, the lower white-background area was generated using all the 60k training images. The top row shows that the images of two and six appear to successfully generate the correct digits even with the 112 pixels having very few clues about the digits. All the other images in the first row appear to be an average training image, since no clue about the digits is included in the 112 pixel values. The fourth row shows that reasonable images are generated from the 448 pixel values, approximately worth 57%, for all the ten test images.

4 Conclusions

Inspired by Bayesian approaches to brain function in neuroscience, we asked how reasoning and learning can be given the same probabilistic account. We simply modelled how data cause symbolic knowledge in terms of its satisfiability in formal logic. The underlying idea is that reasoning is a process of deriving

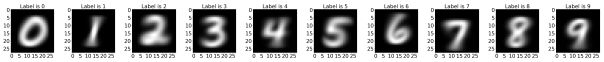


Fig. 7. The images of all the ten digits. We normalised $p(Pixel_j|Digit_i) \in [0, 1]$ to the grayscale between 0 (black) and 255 (white), for all pixels j ($1 \leq j \leq 28 \times 28$).

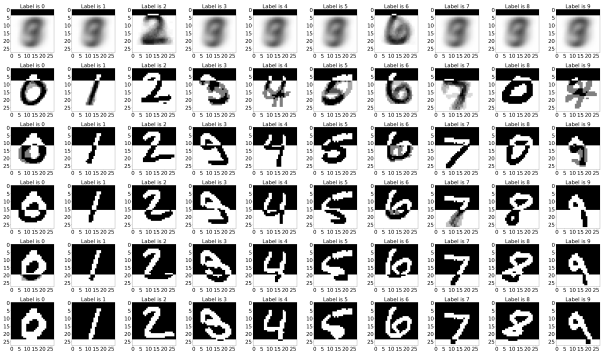


Fig. 8. The upper black-background area of each image in the ith row visualises the first $4i \times 28$ pixel values, approximately worth $14.3i\%$, of a test image. Given the partial information on the test image, the lower masked white-background area is generated using all the 60k training images. We inverted the colours of the generated pixel-values for visibility. Each test image is the first image of the digit in the test dataset. We again normalised the generated pixels.

symbolic knowledge from data by abstraction, i.e., selective ignorance. We empirically showed that it not only generalises a sort of k-nearest neighbour method but also can cancel the assumptions of independence and conditional independence imposed by Bayesian networks and their variants.

References

1. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 39–48 (2016)
2. Botha, L., Meyer, T., Peñaloza, R.: A Bayesian Extension of the Description Logic $\mathcal{ALC}$, vol. 11468, pp. 339–354. Springer, Cham, JELIA 2019 lncs edn. (2019)
3. Colombo, M., Seriès, P.: Bayes in the brain: On Bayesian modelling in neuroscience. The British Journal for the Philosophy of Science **63**, 697–723 (2012)
4. Dasgupta, I., Schulz, E., Tenenbaum, J.B., Gershman, S.J.: A theory of learning to infer. Psychol Rev. **127(3)**, 412–441 (2020)
5. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proc. 16th Int. Joint Conf. on Artif. Intell. pp. 1297–1304 (1996)
6. Friston, K.: The history of the future of the Bayesian brain. Neuroimage **62-248(2)**, 1230–1233 (2012)

7. Funamizu, A., Kuhn, B., Doya, K.: Neural substrate of dynamic Bayesian inference in the cerebral cortex. Nature Neuroscience **19**, 1682–1689 (2016)
8. George, D., Hawkins, J.: A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In: Proc. Int. Joint Conf. on Neural Networks. pp. 1812–1817 (2005)
9. Hohwy, J.: The Predictive Mind. Oxford University Press (2014)
10. Knill, D.C., Pouget, A.: The Bayesian brain: the role of uncertainty in neural coding and computation. Trends in Neurosciences **27**, 712–719 (2004)
11. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350(6266)**, 1332–1338 (2015)
12. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and Brain Sciences **40(e253)**, 1–72 (2017)
13. Lee, T.S., Mumford, D.: Hierarchical Bayesian inference in the visual cortex. Journal of Optical Society of America **20**, 1434–1448 (2003)
14. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence **28**, 71–87 (1986)
15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann; 1st edition, Burlington, Massachusetts (1988)
16. Pearl, J.: Probabilistic Semantics for Nonmonotonic Reasoning, pp. 157–188. Cambridge, MA: The MIT Press, philosophy and AI: essays at the interface edn. (1991)
17. Pearl, J., Russell, S.: Handbook of Brain Theory and Neural Networks, chap. Bayesian Networks, pp. 157–160. MIT Press, Cambridge, Massachusetts (2003)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Rao, R.P.N., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**, 79–87 (1999)
20. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning **62**, 107–136 (2006)
21. Russell, S., Norvig, P.: Artificial Intelligence : A Modern Approach, Third Edition. Pearson Education, Inc., London, England (2009)
22. Sanborn, A.N., Chater, N.: Bayesian brains without probabilities. Trends in Cognitive Sciences **20**, 883–893 (2016)
23. Sanfilippo, G., Pfeifer, N., Over, D.E., Gilio, A.: Probabilistic inferences from conjoined to iterated conditionals. Int. Journal of Approximate Reasoning **93**, 103–118 (2018)
24. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: Proc. 12th int. conf. on logic programming. pp. 715–729 (1995)
25. Tenenbaum, J.B., Griffiths, T.L., Kemp, C.: Theory-based Bayesian models of inductive learning and reasoning. Trends in Cognitive Sciences **10(7)**, 309–318 (2006)
26. Thimm, M.: Inconsistency measures for probabilistic logics. Artif. Intell. **197**, 1–24 (2013)