迈向贝叶斯深度学习：一个框架和一些现有方法

摘要

虽然视觉物体识别和文本理解等感知任务在人类智力中发挥着重要作用，但随后涉及的推理、规划的任务需要更高的智力水平。在过去的几年里，许多使用深度学习模型的感知任务取得了重大进展。然而，对于更高层次的推理，具有贝叶斯性质的概率图形模型仍然更强大和灵活。为了实现涉及感知和推理的集成智能，自然需要将深度学习和贝叶斯模型紧密集成在一个有原则的概率框架内，我们称之为贝叶斯深度学习。在这个统一的框架中，使用深度学习对文本或图像的感知可以提高更高级别的推理的性能，作为回报，推理过程的反馈能够增强对文本或图像的感知。本文提出了贝叶斯深度学习的一般框架，并回顾了其在推荐系统、主题模型和控制方面的最新应用。在本文中，我们还讨论了贝叶斯深度学习与其他相关主题之间的关系和差异，如神经网络的贝叶斯处理。

1. 引言

深度学习在许多感知任务中取得了显著的成功，包括视觉对象识别、阅读（文本理解）和听力（语音识别）。这些无疑是一个正常运作的综合人工智能（AI）或数据工程（DE）系统的基本任务。然而，为了

建立一个真正的AI/DE系统，仅仅能够看到、阅读和听到是远远不够的。最重要的是，它应该拥有思考的能力。

以医学诊断为例。除了看到可见的症状（或CT的医学图像）和听到病人的描述外，医生还必须寻找所有症状之间的关系，最好推断相应的病因。只有在那之后，医生才能为病人提供医疗建议。在这个例子中，虽然视觉和听觉的能力允许医生从病人那里获得信息，但定义医生的是思维部分。具体来说，这里的思考能力可能涉及因果推理、逻辑推导和处理不确定性，这显然超出了传统深度学习方法的能力。幸运的是，另一种类型的模型，**概率图形模型（PGM），擅长因果推理和处理不确定性**。问题是PGM在感知任务上不如深度学习模型好。因此，为了解决这个问题，将深度学习和PGM紧密集成在一个有原则的概率框架内是一个自然的选择，**我们在本文中称之为贝叶斯深度学习（BDL）**。

由于BDL中的紧密和有原则的集成，感知任务和推理任务被视为一个整体，可以相互受益。在上面的例子中，能够看到医学图像有助于医生的诊断和推断。另一方面，诊断和推理也有助于理解医学图像。假设医生不确定医学图像中的黑点是什么。然而，如果她能够推断症状和疾病的病因，就可以帮助她更好地决定黑斑是否是肿瘤。

作为另一个例子，为了在推荐系统(RS) [1]、[39]、[40]、[50]、【67】中实现高精度，我们需要充分了解项目的内容（例如，文档和电影）【46】，分析用户[70]、【73】的配置文件和首选项，并评估用户[3]、[11]、【29】之间的相似性。深度学习在第一个子任务上表现得很好，而PGM在其他两个子任务上表现得很好。除了更好地了解项目内容将有助于分析用户概况之外，用户之间的估计相似性也可以为理解项目内容提供宝贵的信息。为了充分利用这种双向效应来提高推荐准确性，我们可能希望将深度学习和PGM统一在一个单一的原则概率框架中，如【67】所示。

除了推荐系统，当我们处理以原始图像作为输入的非线性动态系统的控制时，也可能需要BDL。考虑根据从摄像机接收的实时视频流控制复杂的动态系统。这个问题可以转化为迭代执行两个任务，从原始图像中感知和基于动态模型的控制。感知任务可以使用多层简单非线性变换（深度学习）来处理，而控制任务通常需要更复杂的模型，如隐藏马尔可夫模型和卡尔曼滤波[22]。然后，反馈循环通过控制模型选择的动作可以影响接收到的视频流作为回报来完成。为了在感知任务和控制任务之间实现有效的迭代过程，我们需要在它们之间进行双向信息交换。感知组件将是控制组件估计其状态的基础，具有内置动态模型的控制组件将能够预测未来轨迹（图像）。在这种情况下，BDL是一个合适的选择【69】。

正如上面的示例中提到的，BDL对于涉及理解内容（例如文本、图像和视频）和变量之间的推理/推理的任务特别有用。在这种复杂的任务中，BDL的感知组件负责理解内容，任务特定组件（例如，动力学系统中的控制组件）对不同变量之间的概率关系进行建模。此外，这两个组件之间的相互作用产生了协同效应，并进一步提高了性能。

BDL除了提供了统一深度学习和PGM的原则性方法的主要优势外，另一个好处来自BDL中内置的隐式正则化。通过将先验强加于隐藏单元、定义神经网络的参数或指定因果推理的模型参数，BDL在某种程度上可以避免过度拟合，特别是当数据不足时。通常，**BDL模型由两个组件组成：（1）感知组件，是某种类型神经网络（NN）的贝叶斯公式；（2）描述不同隐藏或观察变量之间关系的任务特定组件使用PGM。**正规化对他们俩都至关重要。神经网络通常有大量的自由参数，需要正确正则化。权重衰减和丢弃等正则化技术【57】被证明在提高神经网络性能方面是有效的，而且它们都具有贝叶斯解释[15]。就特定任务的组成部分而言，专业知识或先验信息作为一种正则化，可以通过我们强加的先验，在数据稀缺时指导模型，将其纳入模型。

使用BDL处理复杂任务（需要感知和推理的任务）的另一个优势是，它提供了处理参数不确定性的原则贝叶斯方法。当BDL应用于复杂任务时，需要考虑三种参数不确定性：
1)神经网络参数的不确定性。
2)任务特定参数的不确定性。
3)感知部分和特定任务部分之间信息交换的不确定性。

通过使用分布而不是点估计来表示未知参数，BDL提供了一个有希望的框架来统一处理这三种不确定性。值得注意的是，第三个不确定性只能在BDL等统一框架下处理。如果我们分别训练感知组件和任务特定组件，就相当于在两个组件之间交换信息时假设没有不确定性。

当然，在将BDL应用于现实世界的任务时，也存在挑战。(1)首先，设计一个具有合理时间复杂度的神经网络有效贝叶斯公式并不简单。这一工作领域是由[25]、[41]、【44】开创的，但由于缺乏可扩展性，它没有得到广泛采用。幸运的是，最近在这一方向上的一些进展[2]、[9]、[23]、[34]、【66】似乎揭示了贝叶斯神经网络（BNN）的实际采用。1 (2)第二个挑战是确保感知部分和具体任务部分之间高效和有效的信息交流。理想情况下，一阶和二阶信息（例如，平均值和方差）都应该能够在两个分量之间来回流动。一种自然的方法是将感知组件表示为PGM，并将其无缝连接到特定于任务的PGM，如[17]、[64]、【67】中所做的那样。

在本文中，我们的目标是全面概述推荐系统、主题模型（和表示学习）和控制等应用程序的BDL模型。本文的其余部分组织如下：在第2节中，我们回顾了一些基本的深度学习模型。第3节涵盖了PGM的主要概念和技术。这两节是BDL的背景，下一节第4节提出了统一的BDL框架，并调查了应用于推荐系统和主题模型等领域的BDL模型。第5节讨论了未来的一些研究问题，并对本文进行了总结。

2. 深度学习

深度学习通常是指具有两层以上的神经网络。为了更好地理解深度学习，这里我们从最简单的神经网络——多层感知器（MLP）开始，作为一个例子，展示传统深度学习是如何工作的。之后，我们将回顾其

他几种基于MLP的深度学习模型。

2.1 多层感知器

从本质上讲，多层感知器是一系列参数非线性变换。假设我们想训练一个多层感知器来执行一个回归任务，该任务将 M 维向量映射到 D 维向量。我们将输入表示为矩阵 X_0 （0表示它是感知器的第0层）。 X_0 的第 j 行，表示为 $X_{0,j*}$ ，是表示一个数据点的 M 维向量。目标（我们要拟合的输出）表示为 Y 。类似地， Y_{j*} 表示 D 维行向量。学习 L 层多层感知器的问题可以表述为以下优化问题:

$$\min_{\{W_l\},\{b_l\}} ||X_L - Y||_F + \lambda \sum_l ||W_l||_F^2$$

$$s.t.\ X_l = \sigma(X_{l-1}W_l + b_l), l = 1, ..., L-1$$
$$X_L = X_{L-1}W_L + b_L$$

其中 $\sigma(\cdot)$ 是矩阵和 $\sigma(x) = \dfrac{1}{1+exp(-x)}$ 的元素sigmoid函数。 λ 是一个正则化参数， $||\cdot||_F$ 表示Frobeius范数。强加 $\sigma(\cdot)$ 的目的是允许非线性变换。通常，其他变换，如 $tanh(x)$ 和 $max(0, x)$ 可以用作sigmoid函数的替代。

这里 $X_l(l = 1, 2, ..., L-1)$ 是隐藏单位。正如我们所看到的，一旦给出 X_0, W_l 和 b_l ， X_L 就可以很容易地计算出来。由于 X_0 是由数据给出的，我们只需要在这里学习 W_l 和 b_l 。通常，这是使用反向分页和随机梯度下降（SGD）来完成的。关键是计算目标函数相对于 W_l 和 b_l 的梯度。如果我们将目标函数的值表示为 E ，我们可以使用链规则计算梯度:

$$\frac{\partial E}{\partial X_L} = 2(X_L - Y)$$

$$\frac{\partial E}{\partial X_l} = (\frac{\partial E}{\partial X_{l+1}} \cdot X_{l+1} \cdot (1 - X_{l+1}))W_{l+1}$$

$$\frac{\partial E}{\partial W_l} = X_{l-1}^T(\frac{\partial E}{\partial X_l} \cdot X_l \cdot (1 - X_l))$$

$$\frac{\partial E}{\partial b_l} = mean(\frac{\partial E}{\partial X_l} \cdot X_l \cdot (1 - X_l), 1)$$

其中 $l = 1, ..., L$ 和正则化术语被省略。元素乘积表示为 $\bullet$ ， $mean(\cdot, 1)$ 是矩阵上的matlab操作。在实践中，我们只使用一小部分数据（例如128个数据点）来计算每次更新的梯度。这被称为随机梯度

下降。

正如我们所看到的，在传统的深度学习模型中，只有 W_l 和 b_l 是自由参数，我们将在每次优化迭代中更新这些参数。 X_l 不是自由参数，因为如果给定 W_l 和 b_l ，它可以精确计算。

2.2 自动编码器

自动编码器（AE）是一个前馈神经网络，用于将输入编码为更紧凑的表示，并使用学习的表示重建输入。在最简单的形式中，自动编码器不过是一个多层感知器，中间有一个瓶颈层（一个有少量隐藏单元的层）。自动编码器的想法已经存在了几十年[10]、[20]、【35】，并且已经提出了大量的自动编码器变体来增强表示学习，包括稀疏AE [48]、收缩AE [51]、和去噪AE [59]。有关更多详细信息，请参考最近一本关于深度学习的好书[20]。在这里，我们介绍了一种多层去噪AE，称为**堆叠去噪自动编码器（SDAE）**，作为AE变体的示例，也作为其在第4节基于BDL的推荐系统上应用的背景。

SDAE [59]是一个前馈神经网络，用于通过学习预测输出中的干净输入本身来学习输入数据的表示（编码），如图1所示。中间的隐藏层，即，图中的 X_2 可以被限制为学习紧凑表示的瓶颈。传统AE和SDAE的区别在于，输入层 X_0 是干净输入数据的损坏版本。本质上，SDAE解决了以下优化问题：

$$\min_{\{W_l\},\{b_l\}} ||X_c - X_L||_F^2 + \lambda \sum_l ||W_l||_F^2$$

$$s.t.\ X_l = \sigma(X_{l-1}W_l + b_l), l = 1, ..., L-1$$
$$X_L = X_{L-1}W_L + b_L$$

在这里，SDAE可以被视为上一节中描述的回归任务的多层感知器。MLP的输入 X_0 是数据的损坏版本，目标 Y 是数据 X_c 的干净版本。例如， X_c 可以是原始数据矩阵，我们可以将 X_c 中 30 的条目随机设置为0，并得到 X_0 。简言之，SDAE学习了一个神经网络，该神经网络将噪声数据作为输入，并在最后一层恢复干净数据。这就是“去噪”的意思。通常，中间层的输出，即图1中的 X_2 ，将用于紧凑地表示数据。

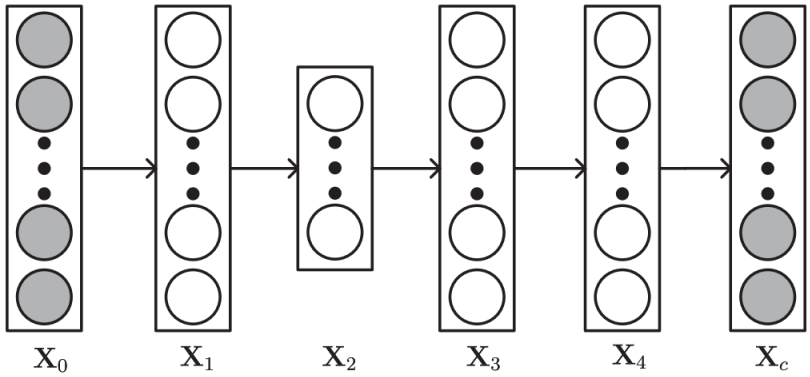

Fig. 1. A 2-layer SDAE with $L = 4$.

2.3 其他深度学习模型

其他常用的深度学习模型包括卷积神经网络(CNN) [31]、【36】，它应用卷积算子和池化算子来处理图像或视频数据，以及递归神经网络(RNN) [20]、【26】，它使用循环计算来模拟人类记忆，以及限制波尔兹曼机器(RBM) [24]，它们是具有二进制隐藏和可见层的无向概率神经网络。请注意，有大量关于深度学习和神经网络的文献。本节的介绍仅作为BDL的背景。读者可参考【20】了解全面调查和更多详细信息。

3. 概率图形模型

以深度学习和PGM为背景，我们现在准备介绍BDL的总体框架和一些具体示例。具体来说，在本节中，我们将列出一些最近的BDL模型，这些模型在推荐系统和主题模型上具有应用。这些模型的总结见表1。

3.1 模型

正如文献[5]所指出的，PGM有两种主要类型，有向PGM（也称为贝叶斯网络）和无向PGM（也称为马尔科夫随机场），尽管存在混合型PGM。在本文中，我们主要关注有向PGMs 关于无向PGMs的细节，读者可以参考[5]。

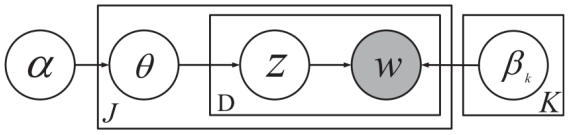

Fig. 2. The probabilistic graphical model for LDA, J is the number of documents and D is the number of words in a document.

PGM的一个经典例子是潜伏狄里切特分配（LDA），它被用作主题模型，分析文档中的词和主题的生成。通常PGM带有模型的图形表示和生成过程，描述随机变量是如何一步步生成的。图2显示了LDA的图形模型，相应的生成过程如下:

- 对于每个文档 $j(j = 1, 2, ..., J)$,
  - 绘制主题比例 $\theta_j \sim Dirichlet(\alpha)$
  - 对于一项（paper） w_j 的每一个词 w_{jn} :
    - 绘制主题任务 $z_{jn} \sim Mult(\theta_j)$
    - 绘制单词 $w_{jn} \sim Mult(\beta_{z_{jn}})$

上面的生成过程给出了随机变量如何生成的故事。随机变量是如何产生的。在图2的图形模型中，阴影节点表示观察到的变量，而其他节点是潜变量（ θ 和 z ）或参数（ α 和 β ）。我们可以看到，一旦定义了模型，就可以应用学习算法来自动学习潜变量和参数。

由于其贝叶斯的性质，像LDA这样的PGM很容易被扩展到其他信息或执行其他任务。延伸，以纳入其他信息或执行其他任务。例如，在LDA之后，人们提出了基于它的不同变种的主题模型。作者在[7]、[61]中提出要纳入时间信息，[6]通过假设话题之间的相关性来扩展LDA。为了使处理大型数据集成为可能，[27]将LDA从批处理模式扩展到在线设置。在推荐系统方面，[60]扩展了LDA，以纳入评级信息并进行推荐。这个模型随后被进一步扩展到纳入社会信息[49]，[62]，[63]。

3.2 推理和学习

严格来说，寻找参数（如图2中的 α 和 β ）的过程称为学习，而给定参数寻找潜在变量（如图2中的 θ 和 z ）的过程则称为推理。然而，只给定观察变量（如图2中的w），学习和推理往往是相互交织的。通常，LDA的学习和推理会在潜变量的更新（对应推理）和参数的更新（对应学习）之间交替进行。一旦LDA的学习和推理完成，我们就有了参数 α 和 β 。如果有新的文档到来，我们现在可以固定学习的 α 和 β ，然后单独进行推理，找到新文档的主题比例 θ_j 。

与LDA一样，每种PGM都有不同的学习和推理算法。可用于每个PGM。在这些算法中，最具成本效益的可能是最大后验（MAP），它相当于将潜变量的后验概率最大化。使用MAP，学习过程等同于最小化（或最大化）一个带有正则化的目标函数。一个著名的例子是概率矩阵分解法（PMF）[53]。在PMF中对图形模型的学习相当于将一个大矩阵分解成两个具有L2正则化的低秩矩阵。

MAP虽然高效，但它只给了我们潜变量（和参数）的点估计。为了将不确定性考虑在内并充分利用贝叶斯模型的力量，我们必须求助于贝叶斯的处理方法，如变分推理和马尔科夫链蒙特卡洛（MCMC）。例如，最初的LDA使用变异推理，用因子化的变异分布近似真实后验[8]。然后，潜变量和参数的学习可以归结为最小化变异分布和真实后验分布之间的KL–分歧。除了变分推理，贝叶斯处理的另一个选择是使用MCMC。例如，有人提出了诸如[47]等MCMC算法来学习LDA的后验分布。

4. 贝叶斯深度学习

有了深度学习和PGM的背景，我们现在准备介绍BDL的一般框架和一些具体例子。具体来说，在这一节中，我们将列出一些最近的BDL模型，并将其应用于推荐系统和主题模型。这些模型的摘要见表1。

4.1 一般框架

如第1节所述，BDL是一个有原则的概率框架，具有两个无缝集成的组件：感知组件和任务特定组件。

BDL的PGM。图3以简单BDL模型的PGM为例。左侧红色矩形内的部分表示感知组件，右侧蓝色矩形内的部分表示任务特定组件。通常，感知组件将是深度学习模型的概率公式，其中多个非线性处理层表示为PGM中的链结构。虽然感知组件中的节点和边相对简单，但任务特定组件中的节点和边通常描述变量之间更复杂的分布和关系（如LDA中）。

三组变量。BDL模型中有三组变量：**感知变量**、**铰链变量**和**任务变量**：（1）在本文中，我们用 Ω_p 表示感知变量的集合（例如图3中的A、B和C），它们是感知组件中的变量。通常， Ω_p 将包括深度学习模型的概率公式中的权重和神经元。（2）我们使用 Ω_h 来表示铰链变量集（例如，图3中的J）。这些变量直接与任务特定组件中的感知组件交互。表1显示了每个列出的BDL模型的铰链变量 Ω_h 集。（3）任务变量的集合（例如，图3中的G、I和H），即任务特定组件中与感知组件没有直接关系的变量，被表示为 Ω_t 。

$I.I.D$ 要求。请注意，铰链变量始终位于特定于任务的组件中。通常，铰链变量 Ω_h 和感知分量（例如，图3中的 $C \rightarrow J$ ）之间的连接应该是i.i.d.，以便于感知分量中的并行计算。例如，J中的每一行只与C中的一个相应行相关。虽然在BDL模型中并不是强制性的，但满足这一要求将显著提高模型训练中并行计算的效率。

联合分布分解。如果两个分量之间的边缘指向 Ω_h （如图3所示，其中 $\Omega_p = \{A, B, C, D, E, F\}, \Omega_h = \{J\}$ 和 $\Omega_t = \{I, G, H\}$ ，所有变量的联合分布可以写成:

$$p(\Omega_p, \Omega_h, \Omega_t) = p(\Omega_t)p(\Omega_h|\Omega_p)p(\Omega_t|\Omega_h)$$

如果两个分量之间的边来自 Ω_h （类似于图3，除了边指向J到C），所有变量的联合分布可以写成:

$$p(\Omega_p, \Omega_h, \Omega_t) = p(\Omega_t)p(\Omega_h|\Omega_t)p(\Omega_p|\Omega_h)$$

显然，BDL可能在指向 Ω_h 的两个分量之间有一些边缘，而一些边缘来自 Ω_h ，在这种情况下，关节分布的分解将更加复杂。

与 Ω_h 相关的方差。如第1节所述，BDL的动机之一是对感知组件和任务特定组件之间交换信息的不确定性进行建模，归根结底是对与 Ω_h 相关的不确定性进行建模。例如，这种不确定性反映在等式（5）中的条件密 $p(\Omega_h|\Omega_p)$ 的方差中。根据灵活性程度， Ω_h 有三种类型的方差（为了简单起见，我们假设BDL的联合可能性在我们的示例中是等式（5）、 $\Omega_p = \{p\}, \Omega_h = \{h\}$ 和 $p(\Omega_h|\Omega_p) = \mathcal{N}(h|p, s)$ ）:

- 零方差。零方差（ZV）在两个组件之间的信息交换过程中不假定不确定性。在本例中，零方差意味着直接将 s 设置为 0 。
- 超方差。超方差（HV）假设信息交换期间的不确定性是通过超参数定义的。在示例中，HV意味着 s 是手动调整的超参数。
- 可学习方差。可学习方差（LV）使用可学习参数来表示信息交换期间的不确定性。在示例中， s 是可学习参数。

如上所示，我们可以看到，在模型灵活性方面，LV > HV > ZV。通常情况下，如果模型被正确正则化，LV模型的性能将优于HV模型，后者优于ZV模型。在表1中，我们显示了不同BDL模型中 Ω_h 的方差类型。请注意，虽然表中的每个模型都有特定的类型，但始终可以调整模型以设计其他类型的对应模型。例如，虽然表中的CDL是一个HV模型，但我们可以很容易地调整CDL中的 $p(\Omega_h|\Omega_p)$ ，以设计其ZV和LV对应物。在【67】中，作者比较了HV CDL和ZV CDL的性能，发现前者的性能明显更好，这意味着复杂地建模两个组件之间的不确定性对于性能至关重要。

学习算法。由于BDL的性质，实际的学习算法需要满足以下标准：

1. 它们应该是在线算法，以便为大型数据集进行良好的扩展。
2. 它们应该足够有效，以与感知组件中的自由参数数量线性缩放。

标准（1）意味着传统的变分推理或MCMC方法不适用。通常需要它们的在线版本[28]。大多数基于SGD的方法也不起作用，除非只执行MAP推理（与贝叶斯处理相反）。需要标准（2），因为感知组件

中通常有大量的自由参数。这意味着基于拉普拉斯近似【41】的方法是不现实的，因为它们涉及计算一个随自由参数数量二次缩放的Hessian矩阵。

4.2 推荐系统的贝叶斯深度学习

尽管深度学习在自然语言处理和计算机视觉上的成功应用，但很少有人尝试开发CF的深度学习模型。[54]中的作者使用受限玻尔兹曼机器而不是传统的矩阵分解公式来执行CF,[19]通过合并用户–用户和项目–项目相关性来扩展这项工作。虽然这些方法涉及深度学习和CF，但它们实际上属于基于CF的方法，因为它们不像CTR [60]那样包含内容信息，这对于准确推荐至关重要。[52]中的作者在深度网络的最后一个权重层中使用低阶矩阵分解，以显著减少模型参数的数量并加快训练速度，但它是用于分类而不是推荐任务。在音乐推荐方面，[45]、【68】直接使用传统CNN或深度信仰网络（DBN）来帮助内容信息的表征学习，但它们模型的深度学习组件是确定性的，而不对噪声建模，因此它们的鲁棒性较低。这些模型主要通过松耦合的方法来实现性能提升，而不利用内容信息和评级之间的相互作用。此外，CNN直接链接到评级矩阵，这意味着当评级稀疏时，模型将由于严重的过拟合而表现不佳。

4.2.1 协同深度学习

协作深度学习为了解决上述挑战，【67】中引入了一种称为协作深度学习（CDL）的分层贝叶斯模型，作为RS的一种新的紧耦合方法。基于SDAE的贝叶斯公式，CDL将内容信息的深度表示学习和评级（反馈）矩阵的协作过滤紧密耦合，允许两者之间的双向交互。实验表明，CDL的性能明显优于现有技术。

在下面的文本中，我们将从介绍CDL演示过程中使用的符号开始。之后，我们将回顾CDL的设计和学习。

符号和问题公式。与【60】中的工作类似，CDL中考虑的推荐任务将隐式反馈【30】作为训练和测试数据。J项（文章或电影）的整个集合由J–by–B矩阵 X_c 表示，其中第 j 行是基于大小 B 的词汇表的项 j 的词袋向量 $X_{c.j*}$ 。对于 I 用户，我们定义了一个I–by–J二进制评级矩阵 $R = [R_{ij}]_{I \times J}$ 。例如，在数据集中cieulike–a[60]、[62]、【67】如果用户 i 在其个人库中有文章 j ，则 $R_{ij} = 1$ ，否则 $R_{ij} = 0$ 。给定 R 中的部分评级和内容信息 X_c ，问题是预测 R 中的其他评级。请注意，尽管CDL目前形式侧重于电影推荐（电影情节被视为内容信息）和文章推荐，如【60】在本节中，它足够通用，可以处理其他推荐任务（例如，标记推荐）。

矩阵 X_c 扮演SDAE的干净输入的角色，而噪声损坏的矩阵，也是J–by–B矩阵，用 X_0 表示。SDAE的第l层的输出用 X_l 表示， X_l 是 J 乘 K_l 矩阵，其中 K_l 是第l层中的单元数。与 X_c 类似， X_l 的第 j 行由 $X_{l.j*}$ 表示。 W_l 和 b_l 分别是层 l 的权重矩阵和偏置向量， $W_l, *n$ 表示 W_l 的

列 n ,L是层数。为了方便起见，我们使用 W^+ 来表示所有层权重矩阵和偏差的集合。请注意， $L/2$ 层SDAE对应 L 层网络。

广义贝叶斯的SDAE。根据第2.2节对SDAE的介绍，如果我们假设干净的输入 X_c 和损坏的输入 X_0 都被观察到，类似于[4]，[5]，[12]，[41]，我们可以定义以下广义贝叶斯SDAE的生成过程。

1. 对于SDAE网络的每一层 l ：
   a. 对于权重矩阵 W_l 的每一列 n ，绘制
   $$W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$$
   b. 绘制偏置向量 $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$
   c. 对于 X_l 的每一行 j ,绘制一个干净的输入:
   $$W_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1}, j*), \lambda_s^{-1} I_{K_l})$$
2. 对于每个item j ，绘制一个干净的输入：
   $$X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_{I_B})$$

请注意，如果 λ_s 到了无穷大，方程（7）中的高斯分布将变成以 $\sigma(X_{l-1,j*} W_l + b_l)$ 为中心的Dirac delta分布[58]，其中 $\sigma(\cdot)$ 是sigmoid函数。该模型将退化为SDAE的贝叶斯公式。这就是为什么我们称它为广义SDAE。

请注意，网络的前 $L/2$ 层作为一个编码器，最后 $L/2$ 层作为解码器。后验概率的最大化等同于考虑到权重衰减后重建误差的最小化。

协作式深度学习。以贝叶斯SDAE为组件，CDL的生成过程定义如下：

1. 生成广义贝叶斯SDAE的变量。
2. 对于每个item j ，
   a. 绘制潜在item偏移向量 $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$ ，然后设置潜在项目向量： $v_j = \epsilon_j + X^T_{\frac{L}{2},j*}$ 。
3. 为每个用户 i 绘制一个潜在的用户向量：
   $$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$$
4. 为每个用户–项目对 (i, j) 抽出一个评级 R_{ij} ，即 $R_{ij} \sim \mathcal{N}(u_i^T v_j, C_{ij}^{-1})$ 。

这里 $\lambda_w, \lambda_n, \lambda_u, \lambda_s$ 和 λ_v 是超参数， C_{ij} 是类似于CTR[60]的置信度参数（如果 $R_{ij} = 1$ ，则 $C_{ij} = a$ ，否则 $C_{ij} = b$ ）。请注意，中间层 $X_{L/2}$ 作为评级和内容信息之间的桥梁。这个中间层，连同潜在的偏移量 ϵ_j ，是使CDL同时学习有效的特征表示和捕捉项目（和用户）之间的相似性和（隐性）关系的关键。与广义的SDAE类似，为了提高计算效率，我们也可以将 λ_s 取为无穷大。

当 λ_s 接近正无穷大时，CDL的图形模型如图4所示，为了简化符号，我们分别用 $x_0, x_{L/2}$ 和 x_C 来替代 $X^T_{0,j*}, X^T_{\frac{L}{2},j*}$ 和 $X^T_{c,j*}$ 。

请注意，根据第4.1节的定义，这里的感知变量 $\Omega_p = \{\{W_l\}, \{b_l\}, \{X_l\}, X_c\}$ ，铰链变量 $\Omega_h = \{V\}$ ，以及任务变量 $\Omega_t = \{U, R\}$ ，其中 $V = (v_j)_{j=1}^J$ ， $U = (u_i)_{i=1}^I$ 。

学习。基于上述的CDL模型，所有的参数可以被视为随机变量，因此可以采用完全的贝叶斯方法，如马尔科夫链蒙特卡洛或变异的近似方法[32]可以被应用。然而，这种处理方法通常会产生很高的计算成本。因此，CDL使用EM风格的算法来获得MAP估计，如[60]。