基于信息瓶颈的表征学习方法

Information Bottleneck for Representation Learning: New Vision

(申请清华大学工学硕士学位论文)

培 养 单 位 ： 清华伯克利深圳学院

学　　科 ： 数据科学与信息技术

研 究 生 ： 王 子 丰

指 导 教 师 ： Khalid M. Mosalam教　授

二〇二一年六月

基于信息瓶颈的表征学习方法

王子丰

Information Bottleneck for Representation Learning: New Vision

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Data Science and Information Technology

by

Wang Zifeng

Thesis Supervisor: Professor Khalid M. Mosalam

June, 2021

学位论文指导小组、公开评阅人和答辩委员会名单

指导小组名单

Khalid M. Mosalam	教授	University of California, Berkeley
黄绍伦	副教授	清华大学

公开评阅人名单

丁文伯	助理教授	清华大学
董宇涵	副教授	清华大学

答辩委员会名单

主席	张林	教授	清华大学
委员	付红岩	副教授	清华大学
	袁坚	教授	清华大学
	黄绍伦	副教授	清华大学
	丁文伯	助理教授	清华大学
	叶旻	助理教授	清华大学
秘书	武智源	博士研究生	清华大学

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）按照上级教育主管部门督导、抽查等要求，报送相应的学位论文。

本人保证遵守上述规定。

作者签名：王子丰　　导师签名：Khalid Yosalam

日　　期：2021/5/28　　日　　期：2021/5/28

摘 要

近年来，深度学习的成功在很大程度上依赖于从数据中学习表征的能力。表征学习是通过端到端的方式来自动学习数据的信息表示。学界目前仍然对巨型的深度模型内部的学习和预测行为知之甚少。这种应用与理论之间的失衡推动了一系列尝试理解深层神经网络的工作的出现，包括基于信息瓶颈原理的深层表征学习模型。信息瓶颈理论发现了随机梯度下降法学习神经网络时的拟合和压缩相变。此外，研究还发现表征的信息遗忘是深度学习泛化能力的源泉。后续研究试图将信息瓶颈转化为表示学习的可以利用的目标函数，如用于变分信息瓶颈模型。此外，研究者也尝试通过多种方法估计神经网络中的信息流以期监测它们的训练行为。

然而，信息瓶颈在表征学习中的应用仍然面临着诸多挑战。目前的工作主要集中在高度结构化的数据，如图像、文本、图形等，而很少涉及非结构化数据，如推荐系统中的评级数据。此外，现实世界的评级数据往往是非随机缺失的。由于用户偏好分布的偏差，直接采用信息瓶颈原则用于推荐系统模型训练是不合理的。信息瓶颈理论在深度学习中的普适性也遭到质疑。当使用非双曲正切的激活函数时，网络的信息压缩阶段似乎消失了。这引发了人们对将表征的信息压缩作为泛化能力度量的质疑，并开始寻求其他的泛化度量方法。因此，本文主要从应用和理论两个方面对信息瓶颈理论进行探讨。本文的主要贡献有两个方面：

(1) 为了扩展信息瓶颈的应用，本文将信息瓶颈应用于推荐系统中的协同过滤。本文提出了反事实变分信息瓶颈方法用于在非随机确实的数据上协同过滤模型的纠偏。反事实变分信息瓶颈将任务相关的信息项分解为事实和反事实项，从而学习缺失非随机反馈下的平衡表示，并显著提高了模型的泛化能力。

(2) 为了修正信息瓶颈理论，本文借鉴了 PAC 贝叶斯学习理论的思想，提出了一种新的基于模型参数的信息瓶颈。它是一种在 PAC 贝叶斯泛化保证的信息瓶颈框架，即 PAC 贝叶斯信息瓶颈。此外，本文推导了一种有效的神经网络信息流估计算法和一种基于随机梯度郎之万动力学方法的贝叶斯推理算法，遂得以从 PAC 贝叶斯信息瓶颈的最优吉布斯后验分布中采样模型。实验发现，这种新的信息测度能够解释广泛的神经网络的学习行为。此外，本文发现这种信息测度作为正则化项用于训练时亦可强化神经网络的表示能力。

关键词：信息瓶颈；表征学习；贝叶斯推断；PAC-Bayes 理论；信息论

ABSTRACT

The success of recent deep learning highly depends on learning representations from data. Representation learning automatically learns informative data representations through an end-to-end paradigm thus useful for various downstream tasks. While the deep representation learning is widely rebuilt based on inductive bias, e.g., convolutional networks, recurrent networks, graph convolutional networks, etc., we still know little of what is going on when these giant models learn and predict. This mismatch encourages a surge of effort on understanding deep neural networks, including modeling the deep presentation learning on the basis of information bottleneck (IB) principle. The main finding of IB theory is the fitting and compression phase transition when neural networks are learned by stochastic gradient descent. Besides, it is claimed that the compression of representations after the initial memorizing stage is the source of generalization of deep learning. Encouraged by the success of IB in understanding deep learning, the follow-ups tried to transform IB to tractable objective for representation learning, e.g., variational information bottleneck for image classification. Besides, efforts were made to estimate information flow in neural networks thus allowing us to monitor their behaviour.

However, the practice of IB for representation learning involves several challenges. Current work mainly concentrated on highly structured data, e.g., image, text, graph, etc., but hardly involved unstructured data, e.g., rating data in recommender systems. Additionally, real-world ratings are missing-not-at-random (MNAR) hence prevent us from directly adopting IB principle due to the concern of distribution bias. Moreover, the interpretability of original IB was also intimidated. The original representation based information measure was recently denounced being incapable of explaining neural networks when non-linearities other than tanh are used because the compression phase seems to disappear. This ignites doubts on nominating information compression of representations as the measure of generalization capacity and encourages researchers to seek to other measures. In this paper, we aim to explore IB theory in two major aspects: application and theory. Our main contributions are thus two-fold:

(1) For extending the application of IB, we concentrated on leveraging IB for collaborative filtering (CF) in recommender systems. We proposed **C**ounterfactual **V**ariational

Information **B**ottleneck (CVIB) in order to debias CF with the emergence of counterfactual events. CVIB separates the task-aware sufficiency into factual/counterfactual terms thus learning balanced representations under missing-not-at-random feedback to improve its generalization ability significantly.

(2) For fixing the theory of IB, we delved deeper into IB theory and specifically highlight the pitfalls of representation based IB. We drew the idea from PAC-Bayes learning theory and propose a new weight based IB that is under the umbrella of PAC-Bayes generalization guarantee, namely **P**AC-Bayes **I**nformation **B**ottleneck (PIB). Then, we derived an efficient algorithm for estimating information flow in DNNs and an SGLD-based Bayesian inference algorithm so as to sample from the optimal Gibbs posterior of PAC-Bayes IB. We identified that this new information measure explains broader aspects of learning behaviour of DNNs. Also, this information measure enhances the learned representation capacity when engaged in our algorithm as a regularization.

Keywords: Information Bottleneck; Representation Learning; Bayesian Inference; PAC-Bayes; Information Theory

TABLE OF CONTENTS

摘　要……I

ABSTRACT……II

TABLE OF CONTENTS……IV

LIST OF FIGURES AND TABLES……VII

LIST OF SYMBOLS AND ACRONYMS……X

CHAPTER 1 INTRODUCTION……1

1.1 Background……1

1.2 Representation Learning……2

1.3 Information Bottleneck……3

1.4 Thesis Structure & Contributions……5

CHAPTER 2 LITERATURE REVIEW……7

2.1 Mutual Information Estimation & Learning……7

2.2 Bayesian Inference for Deep Learning……8

2.3 Information Bottleneck & Variational Auto-encoders……11

2.4 PAC-Bayes Learning Theory……14

2.5 Chapter Summary……18

CHAPTER 3 COUNTERFACTUAL INFORMATION BOTTLENECK……20

3.1 Emergence of Counterfactuals & Challenges……21

3.2 Problem Setup……24

3.3 Building Contrastive Information Regularizer……26

3.4 Minimal Embedding Insensitive to Policy Bias……27

3.5 Tractable Optimization Framework……27

3.5.1 Minimal Information Term……28

3.5.2 Contrastive Information Term……28

3.5.3 Task-aware Information Term……30

3.5.4 Algorithm Overview……30

3.6 Experiments 31
3.6.1 Datasets 31
3.6.2 Baselines 31
3.6.3 Experimental Protocol 32
3.6.4 Results & Analysis 32
3.7 Chapter Summary 34

CHAPTER 4 PAC-BAYES INFORMATION BOTTLENECK 36
4.1 On Caveats of Representation-based Information Bottleneck 36
4.2 A New Bottleneck with PAC-Bayes Guarantee 37
4.3 Estimating Information Stored in Weights 39
4.3.1 Closed-form Solution with Gaussian Assumption 39
4.3.2 Bootstrap Covariance of Oracle Prior 39
4.3.3 Efficient Information Estimation Algorithm 41
4.4 Bayesian Inference for the Optimal Posterior 43
4.5 Experiments 45
4.5.1 Information with Different Non-linearities 45
4.5.2 Information with Deeper and Wider Architecture 48
4.5.3 Random Labels v.s. True Labels 49
4.5.4 Information Compression w.r.t. Batch Size 50
4.5.5 Bayesian Inference with Varying Energy Functions 51
4.5.6 Summary of Experiments 52
4.6 Chapter Summary 53

CHAPTER 5 CONCLUSION AND FUTURE WORK 54
5.1 Conclusion of Current Achievements 54
5.2 Future Works 55

REFERENCES 56

APPENDIX A PROOF IN CHAPTER 3 64

APPENDIX B PROOF IN CHAPTER 4 65

ACKNOWLEDGEMENTS 69

声　明 70

RESUME 71

COMMENTS FROM THESIS SUPERVISOR GROUP 73

RESOLUTION OF THESIS DEFENSE COMMITTEE 74

LIST OF FIGURES AND TABLES

Figure 1.1 A hierarchy of representations in DNN is formulated as a Markov chain, reproduced from [13]. 4
Figure 1.2 An overview of the structure and topics covered by this thesis. 5
Figure 2.1 Graphical representations of generative and discriminative models. 9
Figure 2.2 DNN converge to the fixed point of IB, for each point on the bound of IB, there is an unique optimal β, reproduced from [14]. 12
Figure 2.3 The feasible and non-feasible region of IB problem. For a given β, the solution $p(T|X)$ results in a pair of $(I(X;T), I(Y;T))$ on the boundary of the feasible region, reproduced from [50]. 12
Figure 2.4 DNNs appears to be at odds with the classical bias-variance trade-off and demonstrate the ability to achieve double descent with more and more parameters, reproduced from [77]. 17
Figure 3.1 A comic of how the relationship between person is built based on their preferences over items, which is the intuition behind collaborative filtering. Blue line indicates ”like” and red line indicates ”dislike”. Dotted line indicates ”unknown”. Best viewed in color. 20
Figure 3.2 Analysis of simulation results of a motivating example built by Wang et al.[104]. Best viewed in color. 22
Figure 3.3 Analysis of the bias of estimated preference from the simulation results of a motivating example built by Wang et al.[104]. Best viewed in color. 22
Figure 3.4 The rating distributions of two public datasets Yahoo! R3[102] and Coat[105]. The rating distributions are significantly different between the training (MNAR) and test (MAR) sets for both datasets. Reproduced from [106]. Best viewed in color. 23
Figure 3.5 A hierarchy of the process how O decides the appearance of the event and depends on the previous recommendation policy. 24

Figure 3.6 The events and embeddings are separated by O and we introduce additional constrastive scheme between T^+ and T^-. Although the minimality term $I(T;X)$ is tractable for each event, the sufficiency term of the counterfactuals $I(T^-;Y)$ is not accessible. ……26

Figure 3.7 Test results of MF-CVIB with varying α and γ. Shaded regions show the 90% confidence intervals of the test AUC.……34

Figure 4.1 Information stored in weights (left), loss and accuracy (right) of DNNs trained with different non-linearities (linear, tanh, ReLU, and sigmoid). The y-axis of information is in logarithmic scale for better display. ……46

Figure 4.2 Nonlinear compression of a minimal model. (A) A three neuron MLP with Gaussian inputs x, weight w_1, and non-linearity $g(\cdot)$. (B) The representation t produced by tanh activation is binned into a discrete variable T_{binned} for computing information. (C) and (D) are information when tanh and ReLU are picked. This figure is reproduced from [64].……47

Figure 4.3 Information compression with varying layers (2,3,4,and 5) with tanh non-linearities. We show the 50 and 100 moving average of the 4-layer and 5-layer, respectively, for better display. ……48

Figure 4.4 **Left**: Train and test accuracy of model reaches with increasing number of hidden units; **Right**: Complexity measure (information in weights and ℓ_2-norm) with increasing unit number. ……49

Figure 4.5 **Left**: Information stored in weights with varying size of true-label and random-label data; **Right**: Information, test, and train accuracy when noise ratio in labels changes. ……50

Figure 4.6 Information compression with varying batch size and activation functions. The y-axis on the left indicates the information values and the right indicates the model accuracy. We track the average of minimum information at the end of training and the optimal train/test accuracy the model can reach in the course of whole training process. ……51

Figure 4.7 10 times repeated experiments of the test accuracy on SGD and SGLD with varying regularization terms.……52

Table 2.1 A list of commonly used loss functions.……15

Table 3.1 Missing ratings in a recommender system, where ✓, ✗ and ? mean positive, negative and unknown outcomes, respectively.……21
Table 3.2 MSE and AUC on the MAR test set of COAT[105] and YAHOO[102], where the best ones are in bold. ……33
Table 3.3 Average nDCG with 10 runs on the MAR test set of COAT and YAHOO where the best ones are in bold.……33

LIST OF SYMBOLS AND ACRONYMS

$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
a	Scalar
$\mathbf{W}$	Weight matrix
ω	A set of random variables $\{\mathbf{W}_1, \dots, \mathbf{W}_L\}$
$f^{\mathbf{w}}$	Function parametrized by the weight $\mathbf{w}$
$\mathbf{X}$	Dataset inputs (matrix with N rows, one for each sample)
$\mathbf{Y}$	Dataset outputs (matrix with N rows, one for each sample)
S	Dataset $S = (\mathbf{X}, \mathbf{Y})$
$\mathbf{x}_i$	Input sample for model
$\mathbf{y}_i$	Output label for model
$\mathbf{z}_i$	Data point combining both input and output $(\mathbf{x}_i, \mathbf{y}_i)$
$\hat{\mathbf{y}}_i$	Model prediction on input sample $\mathbf{x}_i$
ℓ	Loss function
$\mathcal{N}$	The Gaussian distribution
$\mathbb{R}$	The real numbers
IB	Information bottleneck
BNN	Bayesian neural network
VI	Variational inference
ELBO	Evidence lower bound
KL	Kullback–Leibler
DNN	Deep neural network
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
PAC	Probably approximately correct
e.g.	Exempli gratia (“for the sake of an example”)
i.e.	Id est (“it is”)
i.i.d.	Independent and identically distributed
s.t.	Subject to
w.r.t.	With respect to

RHS	Right hand side
LHS	Left hand side

CHAPTER 1 INTRODUCTION

Representation learning[1], especially deep representation learning (or deep learning, DL), has become one of the most popular techniques since Krizhevsky et al.[2] developed a powerful deep neural network (DNN) that can significantly outperform shallow methods on ImageNet[3]. After that, DNN based methods have thrived in various machine learning domains, including computer vision[4], natural language processing[5], recommendation[6], reinforcement learning[7], and so on. These works make a substantial progress in building better artifical intelligence (AI) applications by DNN. However, it remains elusive why DNN gains so much improvement with adding more layers of representations. Why representation learning with stochastic gradient descent (SGD) so powerful in so wide range of applications? Also, how can we design novel algorithms that can cover failure cases which are troublesome to classical methods? These question encourage us to think of opening the black box of DNN, e.g., for explanation and prescription.

1.1 Background

In statistical learning theory[8] and probably approximately correct (PAC) learning theory[9], a too complex model suffers from *over-fitting* problem thus failing in out-of-sample test. Recent work challenged them when the so-called benign over-fitting is observed in practice. That is, a hierarchical representation learning model with the number of parameters much more than the samples, i.e., *over-parametrized*, still generalizes well on numerous tasks. This phenomenon ignites a surge of research under the theme of rethinking the generalization of representation learning[10]. One pitfall of the traditional learning theory is the lacking consideration of the input data distribution.

Information theory[11] is a promising candidate for unveiling the black box in representation learning. A landmark in this line of research is using information bottleneck (IB) to model the learning process of DNN[12-13]. They utilized mutual information between the layers and the input and output variables to quantify DNN. It sheds light on characterizing generalization of representation learning by taking data distribution into account. After that, experiments showed that stochastic gradient descent (SGD) is capa-

ble of improving the generalization by compressing the redundant information contained in representations through the lens of IB[14].

In information theory, IB was originally proposed as a means of finding minimal sufficient statistics. The minimality term in IB originates from the principle of minimum description length. In the sense of representation learning, IB describes a trade-off between the representation minimality and sufficiency. The minimality term naturally works for a regularization term that urges the representation to generalize better. This property encourages a series of works in adopting IB for better representation learning. For example, by parameterizing the IB by variational inference, variational information bottleneck was proposed for yielding better generalization performance and robustness to adversarial attack of DNN in image classification[15]. For this reason, it is quite surprising to see how IB could be used for more applications beyond the simple image classification task. In fact, we shall see that we can utilize IB for debasing recommender system models with a novel VI based approach.

Moreover, in this paper, we explore a new perspective of information bottleneck. We look into a novel information-theoretic generalization measure that is built upon the mutual information between the learned weight and the selected finite-sample dataset. With this measure, we propose a new information bottleneck, namely PAC-Bayes information bottleneck (PIB), and give an MCMC based solution for approximate inference of the optimal posterior.

1.2 Representation Learning

The success of machine learning algorithms heavily relies on the representation of data. Prior to the deep learning era, manual feature engineering is a necessary step for preprocessing the raw data and then completing effective machine learning. Representation learning, on the other hand, advocates to automatic learning of informative data representations when building classifiers or other predictors. These representations are encouraged to be informative to the underlying explanatory factors from inputs. As a result, they are expected to be useful for downstream tasks or further fine-tuned under supervision. Since then, the core concern is that how to design good objective functions for learning data representations.

Deep learning (DL) has drawn a great revolution in AI research in past ten years. While the depth is a key factors of the sucess of DL, it should be noted DL is still a sub-

set of the conception of representation learning, in other words, we would prefer to call deep learning as *deep representation learning* for preciseness. For example, Word2Vec was proposed to reduce the number of parameters required by the the re-use of parameters[16]. Unlike the common deep learning setting, this distributed representation learning paradigm does not have multiple layers.

As the name indicates, one key characteristic of deep learning is its depth. Two high-level ideas are that the deep hierarchical architecture promotes the *re-use* of features and improve the feature *abstraction* at the tail near the final output[1]. When the number of layers grows, the number of possible paths grows exponentially. This feature allows computational efficiency of DNN in the universal approximation of arbitrary functions. That is, using deeper models can reduce the number of units required to represent the desired function and can reduce the amount of generalization error[17]. On the other hand, the hierarchy of features enables the composition of fundamental concepts towards abstract objects. Generally, more abstract features are more invariant to the local variation of inputs. The invariance thus allowing the excellent predictive power of learned representations.

However, beyond this perceptual understanding of DL, there are still questions waiting for answers. For example, why stochastic gradient descent (SGD) can encourage both efficient and effective training of DNNs? To what extent different batch size influences the generalization of the learned DNNs? On account of them, We follow the idea of casting our eyes on another toolkit, e.g., information theory, to help us understand and design representation learning algorithms throughout this paper.

1.3 Information Bottleneck

In the learning problem, information theory provides a quantitative notion of "relevant" information, defined by the mutual information $I(X;Y)$ as

$$I(X;Y) \triangleq \int\int p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy. \tag{1.1}$$

With the input feature X and the target signal Y, our object of interest is to extract the information contained in X that is relevant to the target Y by an intermediate representation T, i.e., $I(X;Y) = I(T;Y)$. It could be seen that a trivial way to obtain all the relevant information is just making T an identical mapping. Under the context of information theory, this is often formulated as a "rate distortion" problem that characterizes the

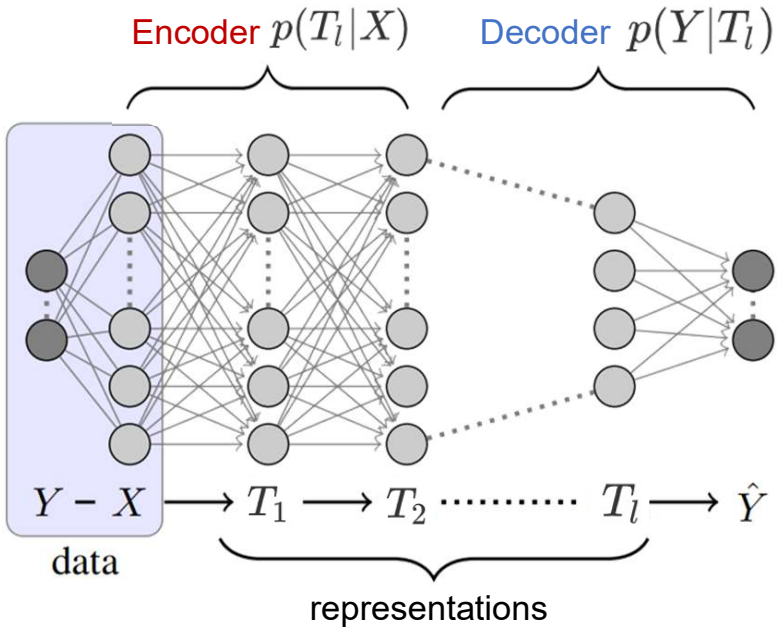


Figure 1.1 A hierarchy of representations in DNN is formulated as a Markov chain, reproduced from [13].

tradeoff between the size of representation, and the average distortion of the reconstructed signal[11]. Apart from the target, we design another target to restrict the irrelevant information by minimizing the mutual information $I(X;T)$. We thus build an optimization problem

$$\min_T I(T;X), \text{ s.t. } I(T;Y) = I(X;Y). \tag{1.2}$$

By solving this problem, we obtain the *simplest* representation of X while still maintain all relevant information of Y. In statistical term, the obtained T is thus called *minimal sufficient statistics* (MSS).

In this setting, we view the information processing of X as a Markov chain that

$$Y \to X \to T \to \hat{Y}. \tag{1.3}$$

For a general distribution $p(X,Y)$, however, the MSS might even not exist and the problem in Eq. (1.2) becomes insolvable. On consideration of this challenge, Tishby et al.[12] proposed a Lagrangian relaxation of the original problem, to build a bottleneck as

$$\min_T I(X;T) - \beta I(T;Y), \tag{1.4}$$

where β is Lagrangian multiplier operates a hyperparameter that controls the tradeoff of regularization and sufficiency. Please keep in mind that an equivalent and usually

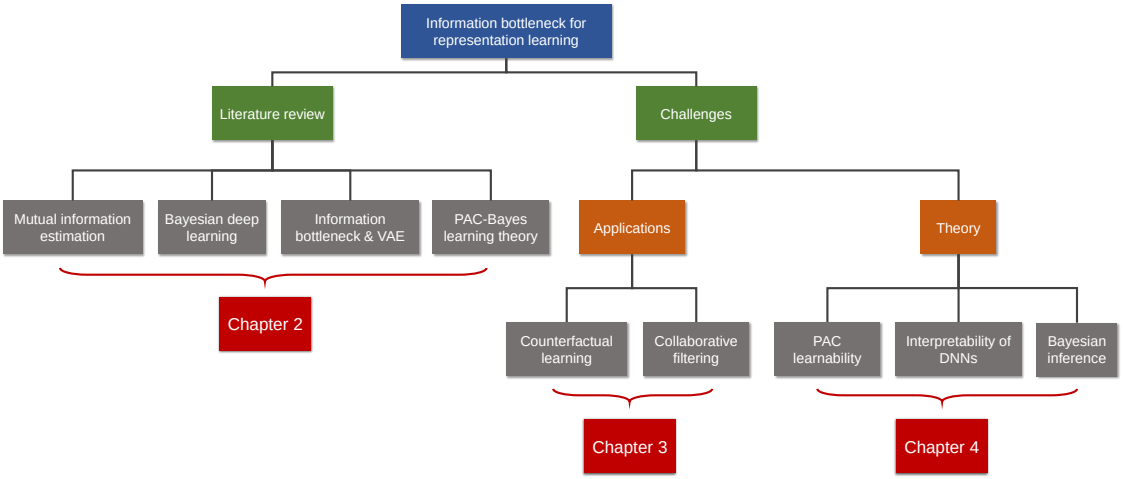


Figure 1.2 An overview of the structure and topics covered by this thesis.

appeared IB is written by

$$\max_{T} I(T;Y) - \beta I(X;T) \tag{1.5}$$

to use β as a hyperparameter to describe the degree of compression.

When there are multiple layers of representations, this Markov chain is extended to $X \to T_1 \to T_2 \dots T_l$ where l is the total number of layers, shown by Fig. 1.1. According to the data processing inequality (DPI), we know that $I(T_1;X) \geq I(T_2;X) \geq \dots I(T_l;X)$, i.e., the layer-wise compression of the input data X continues to happen with more layers. By simultaneously maximizing $I(T;Y)$, it is expected that the learned representations drop the irrelevant information after processing while still keep the predictive power.

In this work, our focus is designing principled information bottleneck methods for representation learning. This method can also be applied for DNN with minimal adaptions in consideration of computational efficiency.

1.4 Thesis Structure & Contributions

The overview framework of this paper is demonstrated by Fig. 1.2 where we follow a flow from the overview to technical details. Our main contributions are presented in §3 and §4 on the application and theory aspects, respectively.

In §1, we introduce the background of representation learning and the motivation of leveraging information theory to understand and enhance representation learning. We also briefly summarize the main contribution of our paper.

In §2, we review a series of works which are relevant to the main topic, including mutual information estimation & learning, Bayesian inference for deep learning, infor-

maiton bottleneck & variational auto-encoders, and PAC-Bayes theory. We discuss the shortcomings of current studies and point out the potential direction of improvement.

In §3, we focus on applying information bottleneck for collaborative filtering from missing-not-at-random feedback which is ubiquitous challenge for recommender systems. Our framework is an extension of variational inference method by considering counterfactuals and designing a contrastive learning regime.

In §4, we concentrate on fixing the theoretical limitations of current representation based information bottleneck by accounting for the information-theoretic generalization capacity of DNNs from the perspective of PAC-Bayes learning theorem. We demonstrate that our new information measure explains many behaviours of DNNs well and how the current deep learning algorithm benefits from the proposed PAC-Bayes information bottleneck.

In §5, we look back on the whole paper to summarize our main contributions. We also discuss the limitations of our current models and point out the promising directions that deserve future works, especially on abandoning the current Gaussian assumption and KL-divergence based information complexity measure as well as on generalization measure under non-identically distributed train/test data.

CHAPTER 2 LITERATURE REVIEW

2.1 Mutual Information Estimation & Learning

Using information theory to understand deep learning has become a vibrant research topic recently. Hjelm et al.[18] pioneered in adopting the information-maximization (InfoMax) principle for unsupervised representation learning. InfoMax means maximizing the mutual information between inputs and outputs to recover the input with reducing redundancies, e.g., InfoMax-based independent component analysis (ICA)[19] that recovers the independent source signals confronting a "cocktail problem". Similar in deep InfoMax, the mutual information between the data representations and raw inputs are maximized. In order to estimate mutual information for high dimensional and continuous representations, they adopted mutual information neural estimation (MINE)[20] by taking Donsker-Varadhan lower bound of mutual information as

$$\begin{aligned} I(X;T) &= \mathrm{KL}(p(X,T) \parallel p(X)p(T)) \\ &\geq \mathbb{E}_{p(X,T)}(f^{\omega}(x,t)) - \log \mathbb{E}_{p(X)p(T)}(\exp(f^{\omega}(x,t)), \end{aligned} \tag{2.1}$$

where $\mathrm{KL}(p \parallel q)$ indicates the Kullback–Leibler divergence between distribution p and q; $f^{\omega} : \mathcal{X} \times \mathcal{T} \mapsto \mathbb{R}$ is a discriminator that decides whether the representation is matched to the input. For instance, for a paired input and representation $(x,t) \sim p(X,T)$, we find another sample $\tilde{x} \sim p(X)$ from the marginal distribution. By setting the RHS as objective function, we are approaching and optimizing the mutual information through an unsupervised manner. In this line of research, deep graph InfoMax was proposed later for unsupervised representation learning from graphs[21].

To quantify mutual information in IB through a lighter way, several techniques were used, including binning and adding Gaussian noise. In [14], the neuron's tanh output activations are discretized into 30 equal intervals between -1 and 1. With these binned values, the joint distribution $p(T_l, X)$ and $p(T_l, Y) = \sum_x p(x,Y)p(T_i|x)$ for layers $l = 1, \ldots, L$, can be obtained with randomly sampled samples x, then used to calculate the mutual information $I(X;T_l)$ and $I(T_l;Y)$. To eliminate the bias of sampling, this process is repeated with 50 different randomized initialized weights and randomly sampled training samples. It is expected to approach the true value of mutual information when bin size is

approaching to zero by definition[11].

However, the binning method is criticized to be trivial in deterministic neural networks with strictly monotone nonlinearities, e.g., tanh or sigmoid, where the mapping is injective[22]. In this scenario, the true mutual information $I(X;T)$ should be infinite when X is continuous or just a constant when X is discrete because

$$I(X;T) = I(X;f(X)) = I(X;X) = H(X), \tag{2.2}$$

where $H(X)$ is a constant without any relation to the model parameters. That is the reason why Goldfeld et al.[23] proposed to inject Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_{d_l})$ in pre-activations as $T_l = f_l(T_{l-1}) + \varepsilon$. As a result, $f_l : \mathbb{R}^{d_{l-1}} \mapsto \mathbb{R}^{d_l}$ becomes a *stochastic* mapping then data processing inequality is satisfied. With intentionally set small noise ε, the performance of noisy DNN is not worse than the deterministic ones.

Adding noise is a compromise for the sake of tractability of mutual information in DNN. Another technique that has been far earlier used for modeling uncertainty in DNNs is variational inference (VI). In the light of it, recent research identified a close connection between IB and variational auto-encoders (VAE)[24] on which we will elaborate in §2.3.

2.2 Bayesian Inference for Deep Learning

Since mutual information based generalization measure can be trivial in deterministic cases, we are interested in generative models to render probabilisic modeling. For example, for a joint distribution $p(X, Y, Z)$, we would factorize it as

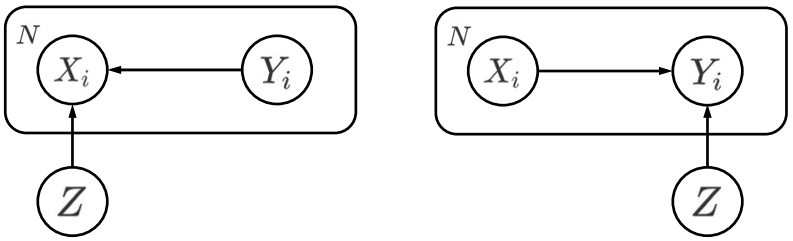
$$p(X, Y, Z) = p(Z) \prod_{i=1}^{N} p(Y_i) p(X_i | Y_i, Z) \tag{2.3}$$

where we should have an assumption to model $p(X|Y, Z)$, e.g., Gaussian mixture model. Difference between discriminative and generative modeling of $p(X, Y, Z)$ is shown by Fig. 2.1. The latent variable Z is used for modeling distribution of X instead of the target Y in generative modeling.

On Bayesian inference for machine learning, our target is to infer the posterior defined by Eq. (2.24) where the denominator is model evidence as $p(S) = \int p(S|\mathbf{w}) p(\mathbf{w}) d\mathbf{w}$. The central equation for inference of this model is

$$p(Y^*|X^*, S) = \int p(Y^*|X^*, \mathbf{w}) p(\mathbf{w}|S) d\mathbf{w}. \tag{2.4}$$

That is, how to make prediction for a new input sample X^* after observing the dataset S. However, in complex models like DNN, both the integration of $p(Y|X, \mathbf{w})$ and $p(\mathbf{w}|S)$ is



(a) Generative (b) Discriminative

Figure 2.1 Graphical representations of generative and discriminative models.

intractable. The first step of approximating this predictive distribution is usually adopting Monte Carlo sampling

$$p(Y^*|X^*, S) \simeq \frac{1}{K}\sum_{k=1}^{K} p(Y^*|X^*, \mathbf{w}_k),\ \mathbf{w}_k \sim p(\mathbf{w}|S). \tag{2.5}$$

K is the total number of weight samples drawn from the posterior $p(\mathbf{w}|S)$. Since the exact formula of $p(\mathbf{w}|S)$ is unknown, the execution of sampling is still troublesome.

Here comes the variational inference (VI). The core idea of VI is to pick a relatively simple parametric distribution $q_\phi(\mathbf{w})$ in replacement of $p(\mathbf{w}|S)$ during the Monte Carlo sampling. In order to ensure this approximate distribution being close to true posterior, we try to minimize the KL divergence $\mathrm{KL}[q_\phi(\mathbf{w}) \parallel p(\mathbf{w}|S)]$. When $p = q$, this term will diminish to zero. The objective of VI is obtained by transforming it to

$$\begin{aligned}\mathrm{KL}[q_\phi(\mathbf{w}) \parallel p(\mathbf{w}|S)] &= -\mathbb{E}_{q_\phi(\mathbf{w})}\left[\log\frac{p(\mathbf{w}, S)}{q_\phi(\mathbf{w})} - \log p(S)\right] \\ &= \log p(S) - \mathbb{E}_{q_\phi(\mathbf{w})}\left[\log\frac{p(\mathbf{w}, S)}{q_\phi(\mathbf{w})}\right].\end{aligned} \tag{2.6}$$

Since the model evidence $\log p(S)$ is a constant w.r.t. model parameters, the original objective function becomes $\max_\phi \mathbb{E}_{q_\phi(\mathbf{w})}\left[\log\frac{p(\mathbf{w},S)}{q_\phi(\mathbf{w})}\right]$. When we move this term in Eq. (2.6) to the left, under the condition that $\mathrm{KL}[q_\phi(\mathbf{w}) \parallel p(\mathbf{w}|S)] \geq 0$ all the time, we will know that $\log p(S) \geq \mathbb{E}_{q_\phi(\mathbf{w})}\left[\log\frac{p(\mathbf{w},S)}{q_\phi(\mathbf{w})}\right]$. That is the reason why we call the new objective the *evidence lower bound* (ELBO)[25-26], which is further transformed to yield the final objective

$$L_{\mathrm{ELBO}} = \log p(S|\mathbf{w}) - \mathrm{KL}[q_\phi(\mathbf{w}) \parallel p(\mathbf{w})]. \tag{2.7}$$

One possible choice of the variational distribution is mean-field Gaussian (fully factorized) $q_\phi(\mathbf{w}) = \prod_{i=1}^{d} q_{\phi_i}(\mathbf{w}_i)$, such that the optimal solution is given by $q^*(\mathbf{w}_j) \propto \exp(\mathbb{E}_{q(\mathbf{w}_{\setminus j})}[\log p(\mathbf{w}_j, S|\mathbf{w}_{\setminus j})])$[①][25], which allows for efficient Gibbs sampling[27] for

① We omit the subscript ϕ of $q(\mathbf{w})$ and $p(\mathbf{w})$ for notation conciseness.

inference. However, mean-field assumption would be too strong for real practices, we would prefer to more complex variational distributions and look for help from optimization techniques. This encourages the proposal of stochastic variational inference[28] for mini-batch training as $\log p(S|\mathbf{w}) \simeq \frac{n}{m} \log p(B|\mathbf{w})$ where $B \sim p^m(Z)$ is a mini-batch sample drawn from S.

With VI in hand, we could adapt the common deterministic neural networks to Bayesian neural networks (BNN). A typical neural network with non-linearity function $g(\cdot)$ is a combination of successive representations

$$f^{\mathbf{w}}(x) = \mathbf{W}^L(g(\mathbf{W}^{L-1}g(\dots g(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1)) + \mathbf{b}^{L-1}) + \mathbf{b}^L, \tag{2.8}$$

where the l-th layer representation is $\mathbf{t}^l = g(\mathbf{W}^l\mathbf{t}_{l-1} + \mathbf{b}^l)$ and $\mathbf{t}^1 = g(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1)$. Now the parameter set $\mathbf{w} = \{\mathbf{W}^l, \mathbf{b}^l\}_{l=1}^L$. We usually take a Gaussian assumption for the variational distribution $q_\phi(\mathbf{w})$ and $\phi = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ is the parameters for learning. For BNN, the inference in ELBO (Eq. (2.7)) could hence be approximated by

$$\begin{aligned} \mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{Y}|\mathbf{X}, \mathbf{w})] &\simeq \frac{n}{m} \sum_{i=1}^{m} \frac{1}{K} \sum_{k=1}^{K} \log p(y_i|\mathbf{x}_i, \mathbf{w}_k), \\ &\text{where } \mathbf{w}_k = \boldsymbol{\mu} + \boldsymbol{\sigma}\varepsilon_k,\ \varepsilon_k \sim \mathcal{N}(0, I). \end{aligned} \tag{2.9}$$

Beyond VI, another lighter approaches were proposed for Bayesian inference of DNNs, including Monte Carlo dropout[29], Deep Ensembles[30], DUQ[31], DUN[32], stochastic gradient Langevin dynamics (SGLD)[33], etc. Gal et al.[29] reinterpreted dropout as a Bayesian approximation of the Gaussian process (GP)[34], thus proving that performing forward passes with turning on dropout is equivalent to Monte Carlo sampling. This method is easy-to-implement on networks with dropout modules but relies on coarse approximations for ensuring scalability, then often results in limited or unreliable estimates. Recent works corroborated MC dropout works well only when the DNNs are sufficiently wide[35-37]. Lakshminarayanan et al.[30] trains multiple networks with different random seeds and regards this process as Monte Carlo sampling, but this method is too expensive in computation. van Amersfoort et al.[31] proposed to use an RBF network[38] to quantify uncertainty from a deterministic network. This method obtains excellent uncertainty in a single forward and maintains competitive performance, however, it requires an RBF module hence not aligned with the most state-of-the-art network architectures. Antorán et al.[32] assumed depth of networks as random variables to exploit the sequential structure of feed-forward networks, which also allows Bayesian inference in a single

pass. In DUN, the network depth is assumed following a categorical distribution with trainable parameters. ELBO is then estimated and optimized for updating both weight and depth parameters. This method views each block of the network as a black box, though can yield predictive uncertainty, it cannot explain the exact parameter uncertainty in networks.

A recently proposed Markov chain Monte Carlo (MCMC) based approach by Welling et al.[33], namely stochastic gradient Langevin dynamics (SGLD), became more and more popular in Bayesian deep learning, because it is rather scalable and easy-to-implement. Traditional MCMC sampling needs to generate proposals using the entire dataset. SGLD only relies on a subset of data, i.e., mini-batch, by skipping the accept-reject step and decreasing step-sizes (learning rate) sequence. Technically, SGLD updates by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} U(\mathbf{w}_t) + \sqrt{2\eta}\varepsilon,\ \varepsilon \sim \mathcal{N}(0, I). \tag{2.10}$$

$U(\mathbf{w}_t)$ is the defined *energy function* equivalent to the loss function we utilize for training DNNs and η is the step size. It is easy to find this formula is a simple adaptation from vanilla stochastic gradient descent (SGD) by injecting isotropic Gaussian noise in each single step. The proof of stability and consistency of SGLD was also given later[39-41]. A series of follow-ups tried to develop faster SGLD by pre-conditioning[42-43] or variance reduction[44]. Also, novel energy function was proposed for reaching a better posterior[45-46].

Please refer to [47-49] for a review of Bayesian inference in deep learning.

2.3 Information Bottleneck & Variational Auto-encoders

Although InfoMax was proved useful in learning representations in unsupervised scheme, it has no connection to most of deep learning algorithms, as supervised learning algorithms are at present the mainstream in applications. Moreover, it does not touch the essence of information theory or the source of generalization of deep learning.

On the other hand, information bottleneck fits the principle of representation learning better: extracting information in input X that is relevant for predicting the target Y by encoding X in a compressed bottleneck T. IB offers a mechanism to explain how DNNs train and generalization, as well as being a novel objective for training. IB was initially proposed in 1999 by [12] under the context of communication. Tishby et al.[13] pioneered in modeling DNNs as a Markov chain and offering insight about how IB could explain

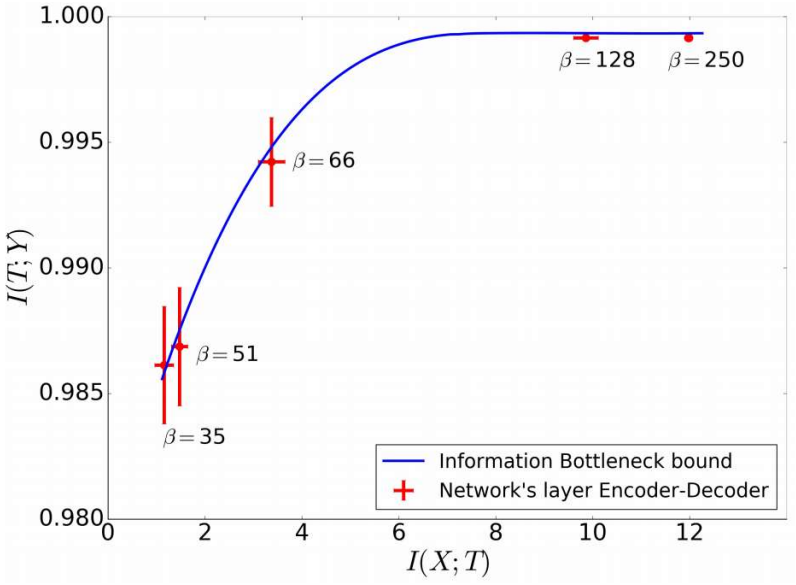


Figure 2.2 DNN converge to the fixed point of IB, for each point on the bound of IB, there is an unique optimal β, reproduced from [14].

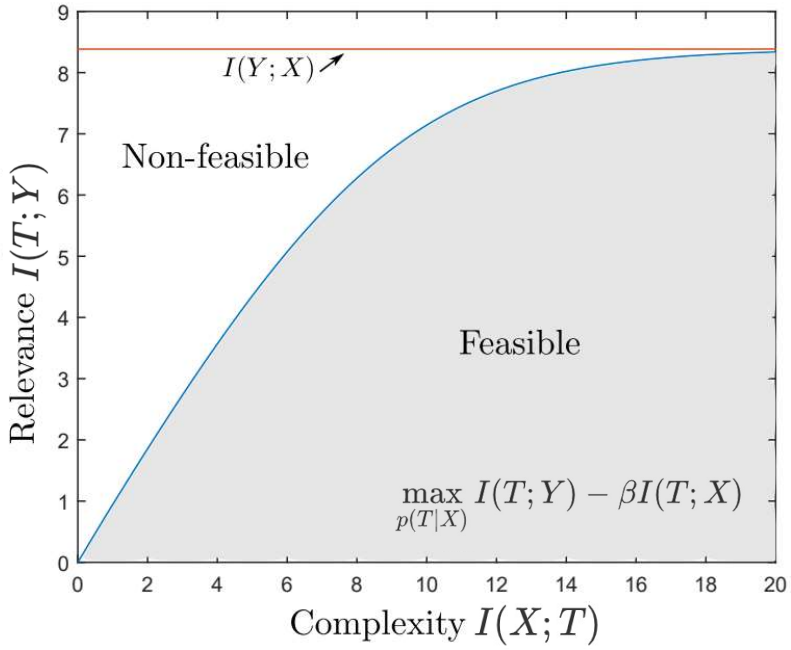


Figure 2.3 The feasible and non-feasible region of IB problem. For a given β, the solution $p(T|X)$ results in a pair of $(I(X;T), I(Y;T))$ on the boundary of the feasible region, reproduced from [50].

the behavior of DNN training. This work was soon followed by Shwartz-Ziv et al.[14] who experimented on a fully-connected feedforward neural network with visualizing the information plane, i.e., the plane of the mutual information values that each layer preserves on the input and output variables. An information plane is plotted in Fig. 2.2. The blue line indicates the optimal IB under different β^*, i.e., $I(T;Y) - \beta^* I(T;X)$, where $\frac{1}{\beta^*}$ is the slope

of that point. This corresponds to the theoretical feasible region of convex information Lagrangian problem, shown by Fig. 2.3.

One would notice that IB described in Eq. (1.4) has similar formula to variatioanl auto-encoders (VAE)[24] learned on the marginal likelihood by

$$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{t}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{t})], \tag{2.11}$$

where ϕ, θ parameterize the output distribution of encoder $q_\phi(\mathbf{t}|\mathbf{x})$ and decoder $p_\theta(\mathbf{x}|\mathbf{t})$ in VAE. The above object can further be written as

$$\log p_\theta(\mathbf{x}|\mathbf{t}) \geq \mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{x}, \mathbf{t}) = \mathbb{E}_{q_\phi(\mathbf{t}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{t})] - \text{KL}[q_\phi(\mathbf{t}|\mathbf{x}) \parallel p(\mathbf{t})]. \tag{2.12}$$

Prior $p(\mathbf{t})$ and posterior $q_\phi(\mathbf{t}|\mathbf{x})$ are usually assumed to be Gaussian distributions for the sake of tractability. Higgins et al.[51] proposed to add a hyperparameter β on the second term thus

$$\mathcal{L}_{\beta-\text{VAE}}(\theta, \phi; \mathbf{x}, \mathbf{t}) = \mathbb{E}_{q_\phi(\mathbf{t}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{t})] - \beta\text{KL}[q_\phi(\mathbf{t}|\mathbf{x}) \parallel p(\mathbf{t})] \tag{2.13}$$

benefits for learning disentangled latent representations $\mathbf{t}$ by carefully chosen β.

On the other hand, β-VAE is closely related to IB when the target task is reconstruction[52-53]. We can think $q_\phi(\mathbf{t}|\mathbf{x})$ as a set of independent additive white Gaussian noise channels. In this perspective, the KL divergence term $\text{KL}[q_\phi(\mathbf{t}|\mathbf{x}) \parallel p(\mathbf{t})]$ is the upper bound of information that is transmitted through this channel per sample as

$$I(T; X) \leq \int\int p(\mathbf{x}) q_\phi(\mathbf{t}|\mathbf{x}) \log \frac{q_\phi(\mathbf{t}|\mathbf{x})}{p(\mathbf{t})} d\mathbf{x} d\mathbf{t}. \tag{2.14}$$

This KL term is zero when $q_\phi(\mathbf{t}|\mathbf{x}) = p(\mathbf{t}) = \mathcal{N}(0, 1)$, in other words, the latent channel has zero capacity. Now taking an information-theoretic perspective, β-VAE works by provably encoding the information from data that contributes most significantly to improve data log-likelihood, i.e., $\mathbb{E}_{q_\phi(\mathbf{t}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{t})]$. The RHS of Eq. (2.14) equals to one component of variational information bottleneck (VIB) which is used for representation learning for image classification[15].

Apart from applications mentioned above, VIB is adapted to a wide range of tasks, e.g., compression of convolutional neural networks (CNNs)[54] and language model embedding compression[55]. Besides, Achille et al.[56] tried to provide a decomposition of information in representations and the interpretation of invariance and disentanglement happened in DNNs. Other adaptations and applications of IB include proposing a second order IB for non-linear encoding and decoding maps[57]; design a graph information bottleneck for graph representation learning[58]; learning disentangled representations[59];

reinforcement learning[60]; mutli-view learning[61]. Please refer to [50,62-63] for a review of recent advancement on using IB in representation learning.

Although the variants adapted from IB were demonstrated to be beneficial, there still lie in critics that doubt these benefits might be borrowed from unrelated mechanism[64-65]. The first and foremost challenge is that mutual information is really hard for estimation[66], which is the direct reason why Shwartz-Ziv et al.[14] adopted the binning trick for approximation and further encourages a surge of work on developing mutual information estimators for DNNs[20,67-69]. Apart from the caveats of mutual information estimation in DNNs described by Eq. (2.2), experiments showed that the compression phase observed by Shwartz-Ziv et al.[14] might be just a side-product of the picked double-sided saturating non-linearities like tanh[64]. When the activation function is changed to rectified linear units (ReLU) or just being removed (linear activation), this compression phase disappear during the training phase. Also, Goldfeld et al.[23] found DNNs with special orthonormal regularization[70] can generalize well but compression does not occur. Instead, they claimed that the clustering of internal representations is the source of generalization. These works cast doubts on the causal relationship between compression and generalization of representation learning under the measure of information contained in representations $I(X;T)$. In a nutshell, our understanding of how information affects DNN generalization still remains unclear.

2.4 PAC-Bayes Learning Theory

When we talk about learning theory, our primary aim is to explain what can a system learn about the underlying phenomenon from examples. Based on our empirical observation and intuition, brutally memorizing the examples usually causes overfitting. Generalization, as the object of learning algorithms, is the ability that the learned model performs well on *unseen data*.

For completeness, we here introduce the mathematical formalization of learning theory. Main body of this framework could be found from [71]. Consider a learning algorithm $\mathcal{A} : \mathcal{Z}^n \mapsto \mathcal{W}$ that learns from the finite-sample data space $\mathcal{Z}^n$ and outputs a parameter located in the parameter space $\mathcal{W}$. The sample space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is the combination of sets of inputs and outputs. We have an unknown data generating distribution $p(Z)$ over $\mathcal{Z}$ while the algorithm has no access to it but just the training set S. Each training sample is i.i.d. from $p(Z)$ such that $S \sim p^n(Z)$. Statistical learning theory tries to

touch the distribution of test errors, under the fixed algorithm $\mathcal{A}$, function class $\mathcal{W}$, sample size n, and finite samples S. In detail, we find bounds which hold with high probability over random samples of size n, i.e., the tail of test error distribution. PAC is the shorthand of probably approximate correct[9]. It uses a *confidence parameter* δ to describe the probability of being misled by the training set $P^n(\text{large error}) \leq \delta$, in other words, the high confidence is represented by $P^n(\text{approximately correct}) \geq 1 - \delta$.

A loss function $\ell(f^{\mathbf{w}}(X), Y)$ measures the discrepancy between the predicted output $f(X)$ and the label Y. Table 2.1 gives a list of examples of loss functions. The *empirical risk* is defined in-sample on the dataset S

$$L_S(\mathbf{w}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(f^{\mathbf{w}}(X_i), Y_i), \tag{2.15}$$

and the *out-of-sample risk* can be defined on the generating distribution $p(Z)$ as

$$L(\mathbf{w}) \triangleq \mathbb{E}_{p(Z)}[\ell(f^{\mathbf{w}}(X), Y)]. \tag{2.16}$$

Hence the generalization gap that evaluates how well a predictor f^w performs on out-of-sample test is given by

$$\Delta(\mathbf{w}) \triangleq L(\mathbf{w}) - L_S(\mathbf{w}). \tag{2.17}$$

A generalization upper bound is usually in the form that $\Delta(w) \leq \epsilon(n, \delta)$. This bound is a *safety check* that guarantees on the performance of an algorithm on *any* unseen data with a prespecified confidence degree.

Table 2.1 A list of commonly used loss functions.

0-1 loss	squre loss	hinge loss	log loss
$\mathbb{1}[f^{\mathbf{w}}(X) \neq Y]$	$(Y - f^{\mathbf{w}}(X))^2$	$(1 - Y f^{\mathbf{w}}(X))_+$	$-\log(f^{\mathbf{w}}(X))$

PAC-Bayes is a generic framework to efficiently rethink generalization for numerous machine learning algorithms. It leverages the flexibility of Bayesian learning and allows to derive new learning algorithms. Prior to the birth of PAC-Bayes theory by McAllester[72], statistical machine learning theory assumes uniform convergence as

$$P^n[\exists \mathbf{w} \in \mathcal{W}, \Delta(\mathbf{w}) > \epsilon] \leq \sum_{\mathbf{w} \in \mathcal{W}} P^n[\Delta(\mathbf{w}) > \epsilon] \tag{2.18}$$

$$\leq |\mathcal{W}| \exp\{-2n\epsilon^2\} = \delta. \tag{2.19}$$

Eq. (2.19) is obtained by the well-known Hoeffding's inequality and holds if the loss function is bounded: $\ell(f^w(X), Y) \in [0, 1]$[73]. Given $\delta \in (0, 1)$, solve equation for ϵ,

we can obtain

$$P^n\left[\Delta(\mathbf{w}) > \sqrt{(\frac{1}{2n})\log(\frac{|\mathcal{W}|}{\delta})}\right] \leq \delta, \tag{2.20}$$

such that $\Delta(\mathbf{w}) = L(\mathbf{w}) - L_S(\mathbf{w}) \leq \sqrt{(\frac{1}{2n})\log(\frac{|\mathcal{W}|}{\delta})}$ with probability $1-\delta$. This upper bound considers the *worst case* as it assumes that hypothesis is uniformly distributed in the hypothesis space. This assumption makes those bounds too *pessimistic* thus being vacuous in various scenarios.

A natural idea to improve this bound is to consider the complexity of the hypothesis space and the *distributions* of hypothesis[74-75]. Take a Bayesian view of learning, before observing the training data, we make an initial guess of the hypothesis distribution $p(\mathbf{w})$ as the so-called *prior*. Corresponding to prior, we would like to learn a distribution $p(\mathbf{w}|S)$ after the observation, i.e., *posterior*. Each prediction is hence drawn from a random predictor $f^{\mathbf{w}}$ where $\mathbf{w} \sim p(\mathbf{w}|S)$. Unlike Eq. (2.15) and Eq. (2.16), empirical risk is now defined based on the averaging over the posterior

$$L_S(\mathbf{w}) = \mathbb{E}_{p(\mathbf{w}|S)}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(f^{\mathbf{w}}(X_i), Y_i)\right], \tag{2.21}$$

and out-of-sample risk is

$$L(\mathbf{w}) = \mathbb{E}_{p(\mathbf{w}|S)p(Z)}[\ell(f^{\mathbf{w}}(X), Y)]. \tag{2.22}$$

In this Bayes viewpoint of PAC learnability, McAllester[72,76] gave the first PAC-Bayesian bound as

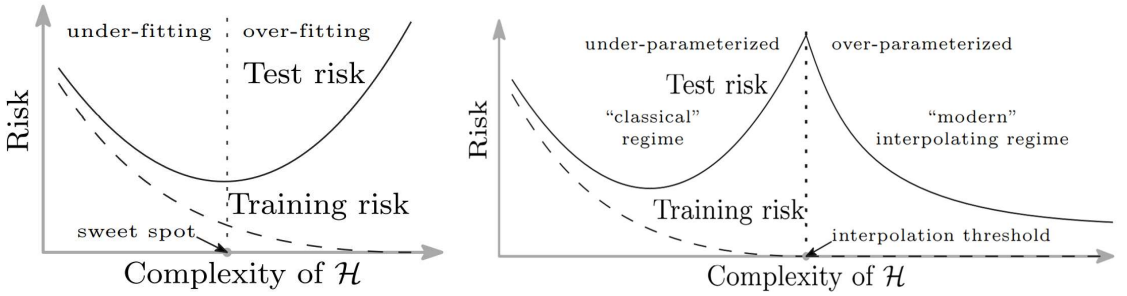
$$P^n\left(L(\mathbf{w}) \leq L_S(\mathbf{w}) + \sqrt{\frac{\mathrm{KL}[p(\mathbf{w}|S) \parallel p(\mathbf{w})] + \log\frac{2\sqrt{n}}{\delta}}{2n}}\right) \geq 1-\delta, \tag{2.23}$$

which holds for $\forall\delta \in (0,1)$ and any prior $p(\mathbf{w})$. Although we describe them as prior and posterior in this paper, please note that they **differ** in their counterparts of Bayesian theory. In PAC-Bayes theorem, the bounds hold even if prior $p(\mathbf{w})$ is incorrect, and **any** chosen posterior $p(\mathbf{w}|S)$. However, in Bayesian inference, the prior must be assumed correct and the posterior is either strictly computed by Bayes theorem

$$p(\mathbf{w}|S) = \frac{p(\mathbf{w})p(S|\mathbf{w})}{p(S)} \tag{2.24}$$

or by statistical modeling like VI.

The scalability of PAC-Bayes theory offers room for explaining why DNN generalizes, especially when classical learning theory is unable to explain the emergence of



(a) Classical bias-variance tradeoff (b) Double descent observed in DNNs

Figure 2.4 DNNs appears to be at odds with the classical bias-variance trade-off and demonstrate the ability to achieve double descent with more and more parameters, reproduced from [77].

double descent in deep learning as shown by Fig. 2.4. Bias-variance tradeoff depicts a threshold where model is under-fitting on the left and over-fitting on the right. However, the modern practice identifies rich DNNs could interpolate (i.e., achieving zero training error) the data while still yield good test performance[77]. This phenomenon is beyond the scope of classical learning theory which assumes an uniform convergence over hypothesis space and neglects data distribution in their bounds. For those reasons, PAC-Bayes inspired a surge of work on deriving non-vacuous generalization bounds for DNNs[78-79].

The most recent PAC-Bayes interpretation of DNN generalization lies in the algorithmic stability towards the perturbation of training data $I(\mathbf{w}; S)$, or it is also called the *information stored in weights*[56,80]. Algorithmic stability has been adopted in PAC-Bayes anlysis by Rivasplata et al.[81], as it is defined by

$$\sup_{i\in[n]} \sup_{z_i, z_{i'}} \|\mathcal{A}(z_{1:i-1}, z_i, z_{i+1:n}) - \mathcal{A}(z_{1:i-1}, z_{i'}, z_{i+1:n})\|_{\mathcal{H}} \tag{2.25}$$

in spirit to *uniform stability*. As it shows, this stability measure is connected to how much the learned weight changes after the replacement of a single data. Similarly, $I(\mathbf{w}; S)$ could also seen as a parameter stability measure that measures how much the perturbation of training data influences the output hypothesis. Inspired by the pioneering work in explaining generalization by this information-theoretic stability[82], Xu et al.[83] proposed a novel PAC-Bayes bound

$$\mathbb{E}_{p(S)}[L(\mathbf{w}) - L_S(\mathbf{w})] \leq \sqrt{\frac{2\sigma^2}{n} I(\mathbf{w}; S)}, \tag{2.26}$$

when $\ell(f^{\mathbf{w}}(X), Y)$ is σ-sub-Gaussian; $L(\mathbf{w})$ and $L_S(\mathbf{w})$ are the *expected* of empirical risk and out-of-sample risk defined by Eq. (2.21) and Eq. (2.22), respectively. Based on this mutual information bound, a series of works proposed tighter bounds for specific algorithms, e.g., on stochastic gradient Langevin dynamics (SGLD)[84-85] and general

iterative algorithms[86], or architectures, e.g., on convolutional neural networks[87]. And several works tried to tighten this bound by novel techniques, e.g., chaining method[88], or changing mutual information to Wasserstein distances[89-91], maximal leakage[92-93], total variation[94], and individual sample mutual information[95].

2.5 Chapter Summary

In this chapter, we introduce four key topics that are deeply connected to the core idea in this paper: information bottleneck for representation learning. On account of the tractability of the information bottleneck objective, we explore the existing literature on how to estimate mutual information in deep neural networks. Although several techniques are able to inject noise into deterministic NN for obtaining an approximate information quantity, e.g., binning, they cannot circumvent the problem inherent in the definition of mutual information when the variable is not random: Mutual information becomes either infinite large or zero in this case. That is the reason why researchers looked for Bayesian inference techniques for DNNs. We then discuss the celebrated variational inference technique used for Bayesian learning and were widely adopted for Bayesian deep learning. Based on VI, deep variational information bottleneck (VIB) was proposed as a tractable objective for representation learning. It was also identified that variational auto-encoders (VAE) and β-VAE are special solutions of VIB, discussed in §2.3. VIB and its variants were applied for a series of tasks, e.g., image classification, graph learning, model compression, etc. In this paper, we extended it to a novel counterfactual learning task where all previous methods fail to work well. We call our method **C**ounterfactual **V**ariational **I**nformation **B**ottleneck (CVIB). We will present the details in §3.

On the other hand, we found there are still critics of the universality of IB in explaining representation learning, as discussed in the last paragraph of §2.3. It is mainly due to the lacking literature of theoretic guarantee of generalization on IB. The celebrated PAC-Bayes learning theory garners our interest to develop a new information-theoretic generalization metric, $I(\mathbf{w}; S)$, which not only measures how much information contained in weights but also how sensitive an algorithm is towards the data perturbation. We then elaborated on the preliminaries of PAC-Bayes theory in §2.4. From the perspective of PAC-Bayes generalization, we propose a novel **P**AC-Bayes **I**nformation **B**ottleneck (PIB). A similar form of IB was proposed by Achille et al.[80]. However, they neither demonstrated the two-phase transition through their appoximated information measure nor gave

the optimal solution and efficient inference approach for IB. In this work, we show how to derive the optimal posterior $p^*(\mathbf{w}|S)$ in principle of PIB. Besides, we demonstrate how to execute Bayesian inference to obtain a series of weights that follow this posterior based on stochastic gradient Langevin dynamics (SGLD) which was introduced as a promising candidate for Bayesian deep learning in §2.2. We will present the technical details of PIB in §4.

CHAPTER 3 WHEN INFORMATION BOTTLENECK MEETS COUNTERFACTUALS

While we have discussed IB and its variants VIB's applications in various topics, there is still room for its extensions to other topics. One domain we identify rather important but still remains under-explored is collaborative filtering with counterfactuals. Collaborative filtering (CF) is well know and widely used for recommender systems[96-97]. CF tries to make automatic predictions (filtering) about the interest of a user by collecting preferences or taste information from many users (collaborating). The idea behind is that if one person A has the same preference as a person B on an item, then person A and B probably hold similar preference over another item than than of a randomly picked person. For example, in Fig. 3.1, we find both User B and User C like Item 3 and dislike Item 1. Meanwhile, we observe User C like Item 4. We would guess since User B has similar preference as User C, it is likely User B should like Item 4 as well.

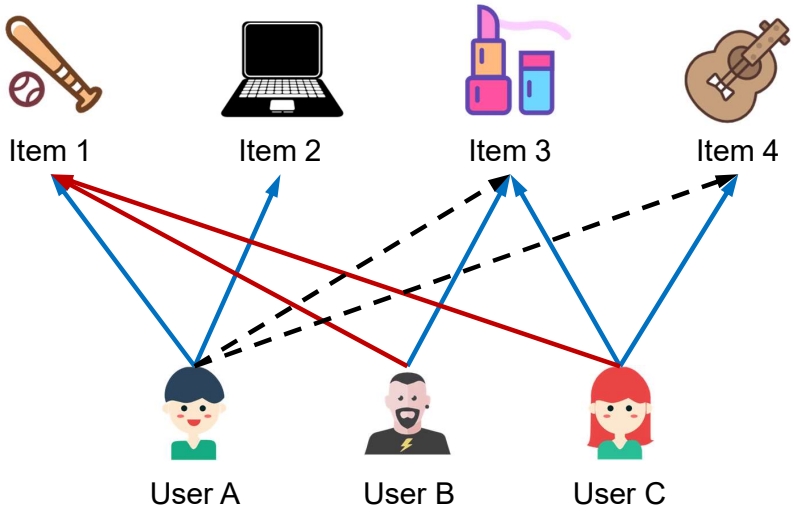


Figure 3.1 A comic of how the relationship between person is built based on their preferences over items, which is the intuition behind collaborative filtering. Blue line indicates "like" and red line indicates "dislike". Dotted line indicates "unknown". Best viewed in color.

Prior to the deep learning era, CF has long been the bedrock of modern recommender systems. Recently, deep learning was introduced to CF and gave rise to the so-called neural collaborative filtering (NCF)[98], neural graph collaborative filtering[99-100] (NGCF), and so on. In this work, we will cover both CF and NCF as they are both under the general arena of representation learning.

As far as we know, there has been by far no previous work on leveraging IB for improving CF. According the superior performance gain by IB in other areas, we believe it will lead to significant improvement in CF as well. However, we identify the original IB is not applicable in this task due to the emergence of *counterfactuals* which we will introduce in the next section. We will show why counterfactuals are harmful for CF and IB based CF and give our answer to it[101].

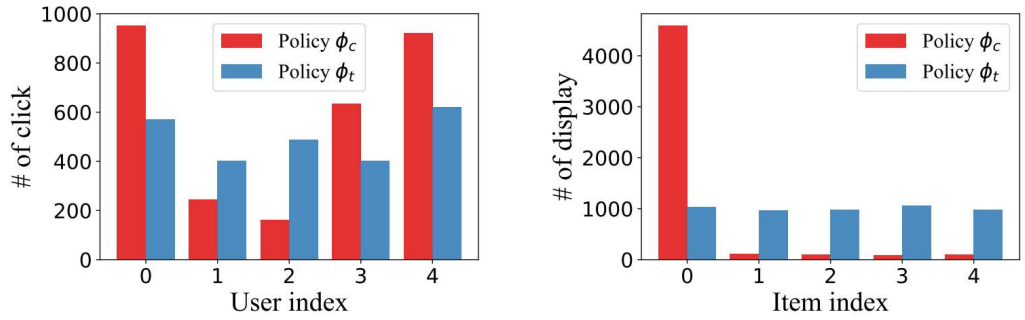
3.1 Emergence of Counterfactuals & Challenges

A surge of research shows that the real-world logging policy often collects missing-not-at-random (MNAR) data (or selective labels)[102]. For example, users tend to reveal ratings for items they like, thus the observed users' feedback, usually described by click-through-rate (CTR), can be substantially higher than those not observed yet.

Table 3.1 Missing ratings in a recommender system, where ✓, ✗ and ? mean positive, negative and unknown outcomes, respectively.

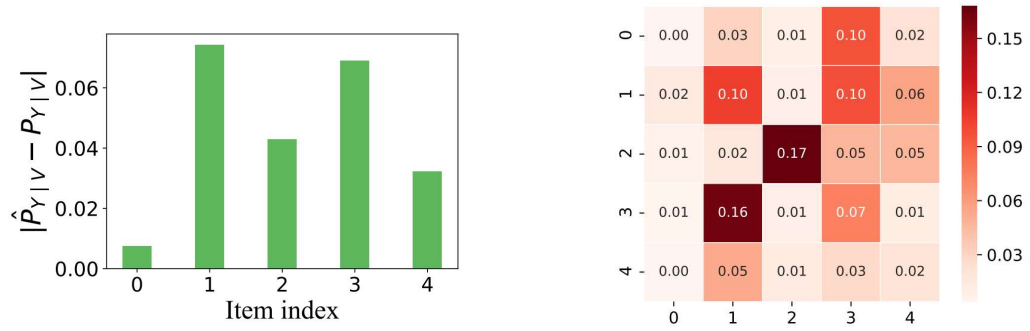
	Item 1	Item 2	Item 3	Item 4
User A	✓	✓	?	?
User B	✗	?	✓	?
User C	✗	?	✓	✓

Let's recap the Fig. 3.1 where there are three users and four items. We can re-formulate the preferences observed from this figure to a preference matrix, shown in Table 3.1. Although we make sure some preferences of users, many of others are still unknown. For instance, we do not know the attitude of User A towards Item 3 and 4. When we ignore the unobserved events, the estimated average CTR from the observed outcomes is $5/7 \approx 0.71$. However, if the rest unobserved outcomes were all 0, the true CTR would be $5/12 \approx 0.42$. This gap between the *factual* and *counterfactual* ratings in MNAR situation further exaggerates due to path-dependence that the learned policy tends to overestimate on the observed events outcomes[103]. This is usually the case in modern recommender systems where there are million of users and billion of items. The total number of possible events, which is the combination of users and items, is astronomical. On the contrary, the observed event outcomes are much less. In fact, we often describe the degree of this phenomenon the *sparsity* of recommender systems. In Table 3.1, the sparsity is $7/12 \approx 0.58$. But in real practice, the sparsity is usually in the degree of 10^{-7} or even much less depending on the size of whole system.



(a) Click count per user (b) Display count of items

Figure 3.2 Analysis of simulation results of a motivating example built by Wang et al.[104]. Best viewed in color.



(a) Bias of item-wise preferences (b) Bias of users' preferences

Figure 3.3 Analysis of the bias of estimated preference from the simulation results of a motivating example built by Wang et al.[104]. Best viewed in color.

In practice, people resort to performing the randomized controlled trials (RCTs) in order to obtain missing at random (MAR) feedback. By deploying a random policy that uniformly displays items to users, we can collect *unbiased* feedback from users. Under this condition, the collected preferences of users will be asymptotically converging to the groundtruth, i.e., the average CTR of observed outcomes equals to the unobserved ones. In other words, the MAR feedback could somehow demonstrate the users' true preferences distribution and could be regarded as the golden rule for evaluating the true performance of recommender systems. A simulation performed by Wang et al.[104] instantiates that a biased system will collapse if the model continues to learn from MNAR data and biasedly displays items, as shown in Fig. 3.2.

In this simulation, We build a tiny system with five users and five items, and compare the user-wise and item-wise numbers of clicks achieved by two policies ϕ_t and ϕ_c. ϕ_t is the uniform policy and ϕ_c is a biased policy. A 5×5 matrix as the users' preference over each item, where each element is generated uniformly within [0, 1], so they are homogeneous. For simulating user's behavior, we deploy an additional latent variable uniformly

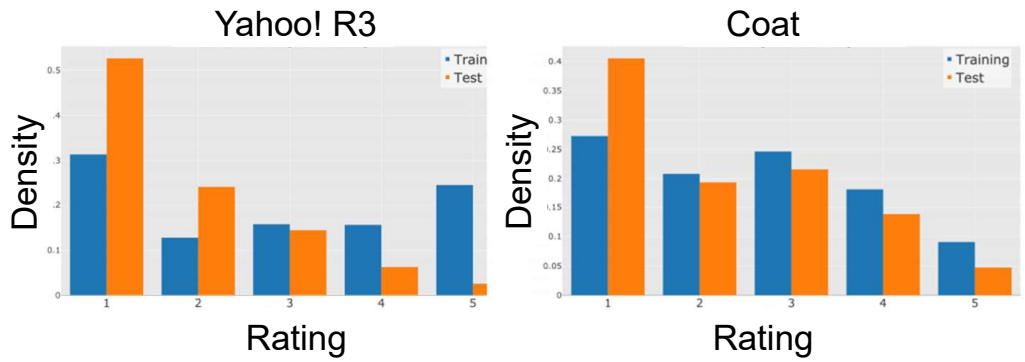


Figure 3.4 The rating distributions of two public datasets Yahoo! R3[102] and Coat[105]. The rating distributions are significantly different between the training (MNAR) and test (MAR) sets for both datasets. Reproduced from [106]. Best viewed in color.

drawn from [0, 1], such that for each time an event x appears, the click happens when the latent variable is smaller than y. We adopt a bandit feedback setting here, that is, the policy ϕ_c firstly ranks the Top-1 predicted reward item for each user, collects the feedbacks as click or not click; then we do policy learning via emiprical risk minimization (ERM) from these logged feedbacks. For controlled experiment, we perform the same setting for an uniform policy ϕ_t as well. After 100 epochs of logging feedbacks and learning policy, we can calculate the empirical outcome distribution from the observed logged data. In Fig. 3.2 (a), we observe the significant bias of the estimated average CTR from the logged feedback generated by ϕ_c. In Fig. 3.2 (b), we find ϕ_c causes devastating reduction of *diversity* to the system, all items except for item #0 are underrepresented, because they are relatively less interested by the users in average. The insufficient exploration over items causes large bias of estimated marginal distribution over items as well as the joint user preferences, shown in Fig. 3.3.

There have also been experiments that reflect the giant gap between MNAR and MAR rating distributions in real-world recommender systems. As shown in Fig. 3.4 where x-axis is the rating ranging from one to five and y-axis is the density. For Yahoo! R3[102], it can be identified there is huge gap on the two sides. In MAR data, the large majority of ratings lies in one and few people give five ratings. However, the distribution is much more flat in MNAR data. As a result, a recommender model that learns from MNAR data should generalize poorly to MAR data and in turn harms diversity and fairness of whole system.

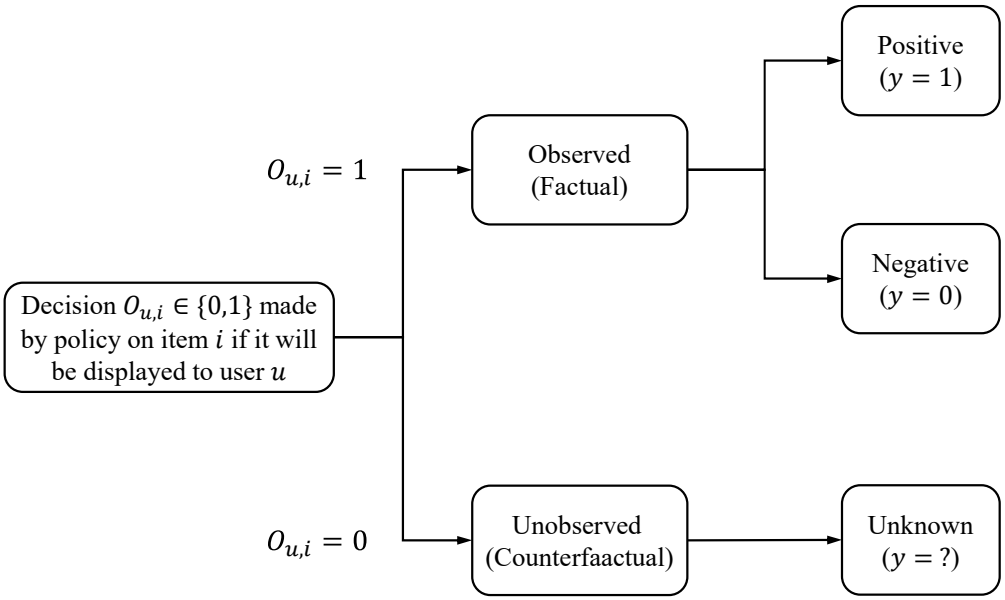


Figure 3.5 A hierarchy of the process how O decides the appearance of the event and depends on the previous recommendation policy.

3.2 Problem Setup

Vast majority of existing works in recommendation neglect the MNAR effect, as they mainly focus on designing novel architectures and training techniques for improving model performance on the observed events[6,107-108], where the objective function is designed in principle of empirical risk minimization (ERM) as

$$L_{ERM} = \frac{1}{N_l} \sum_{(u,i):O_{u,i}=1} \ell_{u,i}(\hat{Y}, Y). \tag{3.1}$$

$x = (u, i)$ is an event composed of user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$; $O_{u,i} \in \{0, 1\}$ indicates whether the outcome of an event $x = (u, i)$ is observed (i.e., whether item i is presented to user u); $\ell_{u,i}(\hat{Y}, Y)$ is the loss function taking true outcome Y and predicted outcome $\hat{Y}$ as its inputs; and N_l and N_{ul} are the numbers of observed and unobserved events, respectively. However, L_{ERM} is not an unbiased estimate of the true risk L[103]:

$$\mathbb{E}_O[L_{ERM}] \neq L := \frac{1}{N_l + N_{ul}} \sum_{u,i} \ell_{u,i}(\hat{Y}, Y), \tag{3.2}$$

which indicates that the naive ERM-based method cannot guarantee model's generalization ability on the counterfactuals.

As the distribution of O depends on the deployed recommendation policy at past, as shown in Fig. 3.5, we can regard it a representative of the *policy bias*, which influences the distribution of factual events and then the learned policy. In order to alleviate the policy bias, there are a series of works emphasizing on employing randomized controlled trials

(RCTs)[109] to collect the so-called *unbiased* dataset. By adopting a uniform policy that randomly displays items to users, the logged feedback can be regarded as missing at random (MAR), which is consistent with the underlying joint distribution $p(x, y)$. Therefore, one can either evaluate the model's true generalization ability on MAR data[110], or utilize MAR data to debias learning via importance sampling[105]. Besides, Rosenfeld et al.[111] and Bonner et al.[112] proposed to employ domain adaptation from MNAR data to MAR data, in order to balance the predictive capability of the learned model over the factuals and counterfactuals. However, RCTs are extraordinarily expensive to be executed in a real-world recommender system. It is tricky because no solid theoretical definition of how large RCTs would be representative enough. It is questionable if small RCTs, compared with the enormous quantity of possible events, can lead to a proper estimate of the users' true preference distribution.

Embedding is a conceptually classical approach for modeling the user and item in rating prediction. For example, in collaborative filtering[113], it represents an event $x = (u, i)$ by concatenation as $\mathbf{t} = (\mathbf{t}_u, \mathbf{t}_i)$, then generates outcome prediction by $\hat{y} = \mathbf{t}_u^\top \mathbf{t}_i$. $\mathbf{t}$ is hence the representation of event x in the sense of representation learning. Recall the Markov chain description of representation learning in Eq. (1.3), we here can introduce an additional nuisance variable O and re-write it as

$$O \to Z \to T \to \hat{Y}, \tag{3.3}$$

where $Z = (X, Y)$ is the combination of event and the true outcome. Also when considering multiple layers in DNNs as described by Eq. (2.8), this Markov chain can be extended to

$$O \to Z \to T_1 \to T_2 \to \ldots \to T_l \to \hat{Y}. \tag{3.4}$$

In this Markov chain, O affects the appearance of events X but is not informative to the true outcomes Y, i.e., $O \perp Y$. In this scenario, we would like T to be independent of O, thus being free of policy bias.

According to information bottleneck, here we can also build an IB objective for CF as

$$\min_T L_{IB} = \beta I(T; X) - I(T; Y). \tag{3.5}$$

The main challenge in adopting IB for optimization is that the mutual information in L_{IB} is cumbersome for calculation. Although previous works try to derive computable proxy for specific tasks[15], they are not suitable for MNAR data. Since we only have *partial*

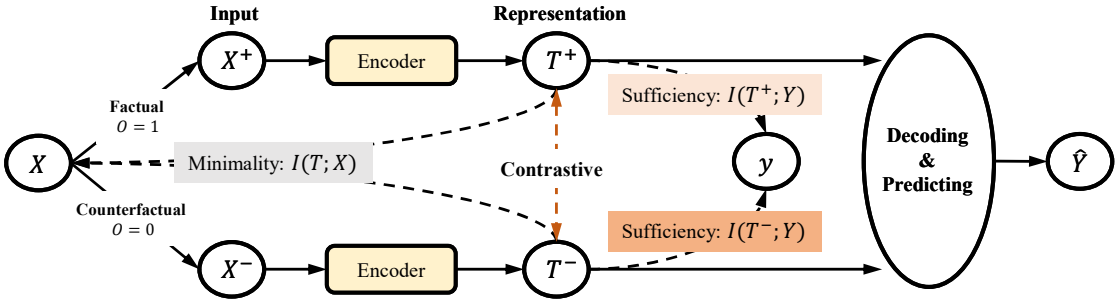


Figure 3.6 The events and embeddings are separated by O and we introduce additional constrastive scheme between T^+ and T^-. Although the minimality term $I(T;X)$ is tractable for each event, the sufficiency term of the counterfactuals $I(T^-;Y)$ is not accessible.

feedback about Y, i.e., the majority of events are counterfactuals, we have no access to their true outcomes. Next, we will turn to derive a new objective function addressing this challenge.

3.3 Building Contrastive Information Regularizer

For conciseness, we focus on a simple model with only the embedding layer T, namely $Y \to X \to T \to \hat{Y}$, and extend it to multi-layer scenario in §3.4. Specifically, the second term in Eq. (3.5) is mutual information between embedding T and target task Y. We separate the embeddings into two parts: T^+ and T^-, which represent factual and counterfactual embeddings, respectively, i.e., $T^+ \sim p(T|X^+)$ and $T^- \sim p(T|X^-)$.

As shown in Fig. 3.6, we factorize the original mutual information term by $I(T;Y) = I(T^+,T^-;Y)$. We postulate that T^+ and T^- are independent, therefore according to the chain rule of mutual information:

$$I(T^+,T^-;Y) = I(T^+;Y|T^-) + I(T^-;Y) = I(T^+;Y) + I(T^-;Y). \tag{3.6}$$

However, as the outcomes Y of the counterfactuals are unknown, we have to identify another refined solution. Specifically, we cast Eq. (3.6) to

$$I(T^+;Y) + I(T^-;Y) = \underbrace{I(T^-;Y) - I(T^+;Y)}_{\text{contrastive}} + 2I(T^+;Y), \tag{3.7}$$

from which we derive a contrastive term between T^+ and T^-. This characterization is helpful for us to introduce a hyperparameter α to control the degree of this contrastive penalty. We then rewrite the original IB loss to

$$\min_{T^+,T^-} L_{CVIB} := \beta I(T;X) + \alpha[I(T^+;Y) - I(T^-;Y)] - I(T^+;Y) \tag{3.8}$$

as our new objective function, where we propose an information theoretic regularization

on T^+ and T^-. Intuitively, minimizing this term corresponds to encouraging T^+ and T^- to be equally informative to the targeted task variable y, thus resulting in a more balanced model. More importantly, it does not need access to the counterfactual outcomes, which will be specified in §3.5.

3.4 Minimal Embedding Insensitive to Policy Bias

Aside from the task-aware mutual information $I(T;Y)$ in Eq. (3.8), $I(T;X)$ corresponds to the minimality of the learned embedding. Recall that we assume that the event X follows the generative process $p(X,O) = p(O)p(X|O)$, where O influences the appearance of X. Because O is independent to task Y, we hope the predicted outcome $\hat{Y}$ is not influenced by O, or the learned embedding T should contain low information about O. In this viewpoint, following the practice by Achille et al.[56], we identify that the minimality term is actually beneficial for embedding's insensitivity against the nuisance O.

Proposition 3.1 (Minimal Representation Insensitive to Policy Bias): With the Markov chain assumption defined by Eq. (1.3), for any hidden embedding T_l, we can derive the upper bound of the $I(T_l;O)$

$$I(T_l;O) \leq I(T_l;X) - I(X;Y) \leq I(T_1;X) - I(X;Y) \leq I(T_l;X), \tag{3.9}$$

where the last term $I(X;Y)$ is a constant with respect to the training process.

Please refer to Appendix A.1 for the proof. Above proposition implies that an embedding T is insensitive to the policy bias O, by simply reducing $I(T;X)$. Meanwhile, by maximizing $I(T;Y)$ in IB Lagrangian, embedding T will be forced to retain minimum information from X that is pertinent to the task Y. In deep models, according to the Data Processing Inequality (DPI)[11], minimizing $I(T;X)$ also works for controlling the policy bias of the successive layers.

3.5 Tractable Optimization Framework

The proposed L_{CVIB} is still intractable for optimization. In this section, we attempt to find a tractable solution for three terms in L_{CVIB}, respectively. And, we present our algorithm of learning debiased embeddings by L_{CVIB} at last.

3.5.1 Minimal Information Term

The minimality term $I(T;X)$ in Eq. (3.8) can be measured with a Kullback-Leibler (KL) divergence as

$$\begin{aligned} I(T;X) &= \mathbb{E}_{p(X)}[\mathrm{KL}[p(T|X) \parallel p(T)]] \\ &= \mathbb{E}_{p(X)}\left[\int p(\mathbf{t}|x)\log p(\mathbf{t}|x)d\mathbf{t} - \int p(\mathbf{t})\log p(\mathbf{t})d\mathbf{t}\right]. \end{aligned} \tag{3.10}$$

To avoid operating on the marginal $p(T) = \int p(T|x)p(x)dx$, we use a variational approximation of $q(T)$ as the marginal $p(T)$, which renders

$$\begin{aligned} -\int p(\mathbf{t})\log p(\mathbf{t})d\mathbf{t} &\leq -\int p(\mathbf{t})\log q(\mathbf{t})d\mathbf{t} \\ &\Rightarrow \mathrm{KL}[p(T|X) \parallel p(T)] \leq \mathrm{KL}[p(T|X) \parallel q(T)]. \end{aligned} \tag{3.11}$$

Suppose the posterior $p(T|X) = \mathcal{N}(\mu(X); \mathrm{diag}(\boldsymbol{\sigma}))$ is a Gaussian distribution, where $\mu(X)$ is the encoded embedding of input event X and $\mathrm{diag}(\boldsymbol{\sigma})$ indicates a diagonal matrix with elements $\boldsymbol{\sigma} = \{\sigma_d\}_{d=1}^{D}$. In other words, we assume the embedding is generated by

$$T = \mu(X) + \varepsilon \odot \sigma, \quad \text{where } \varepsilon \sim \mathcal{N}(0, I). \tag{3.12}$$

If we fix $\sigma_d = \text{const}, \forall d$, then T would default to a deterministic embedding. Moreover, by considering a standard Gaussian variational marginal $q(T) = \mathcal{N}(0, I)$, the KL term reduces to

$$\mathrm{KL}[p(T|X) \parallel q(T)] = \|\mu(X)\|_2^2 + \sum_d \left(\sigma_d - \frac{1}{2}\log\sigma_d\right) - D, \tag{3.13}$$

which means for a deterministic embedding, minimizing term $I(T;X)$ is equivalent to directly applying ℓ_2-norm regularization on the embedding vector. We can also make a more complex assumption, for example, let $\boldsymbol{\sigma}$ be a trainable parameter, then apply a VAE-like architecture that encodes input events to mean and variance terms. We leave it as the future work.

3.5.2 Contrastive Information Term

The mutual information $I(T;Y) = H_p(Y) - H_p(Y|T)$, where $H_p(\cdot)$ demotes the entropy of $p(\cdot)$. The first entropy term is constant, since maximizing $I(T;Y)$ is equivalent to minimizing the second term $H_p(Y|T)$. We then further derive the lower bound of

$-H_p(Y|T)$ as[①]

$$\begin{aligned} I(T;Y) &= \int\int p(y,\mathbf{t}) \log p(y|\mathbf{t}) d\mathbf{t} dy + \text{const} \\ &\geq \int\int p(y,\mathbf{t}) \log q(y|\mathbf{t}) d\mathbf{t} dy + \text{const} = -H_{p,q}(Y|T) + \text{const}. \end{aligned} \tag{3.14}$$

The term $q(y|\mathbf{t})$ is an estimate of $p(y|\mathbf{t})$ with a classifier parameterized by θ, e.g., weight matrices in deep networks or embedding parameters. We can use the cross entropy as a proxy for the mutual information in the IB objective[56,114], because $\max I(T;Y) \Leftrightarrow \min H_{p,q}(Y|T)$ as shown above. Therefore, replacing the contrastive term $I(T^+;Y) - I(T^-;Y)$ in Eq. (3.8) with $H_{p,q}(Y|T^-) - H_{p,q}(Y|T^+)$ obtains

$$\begin{aligned} &H_{p,q}(Y|T^-) - H_{p,q}(Y|T^+) \\ &\qquad = \int\int p(y,\mathbf{t}^+,\mathbf{t}^-) \left[\log q(y|\mathbf{t}^+) - \log q(y|\mathbf{t}^-)\right] dy d\mathbf{t}^+ d\mathbf{t}^-. \end{aligned} \tag{3.15}$$

Since we assume T^+ and T^- are independent, the generative process can be written as $p(Y,T^+,T^-) = p(Y|T^+,T^-)p(T^+)p(T^-)$. In order to make the term tractable, we here approximate $p(Y|T^+,T^-)$ by $q(Y|T^+)$,[②] then cast Eq.(3.15) to

$$\begin{aligned} &\mathbb{E}_{p(T^+,T^-)}\left[\int p(y|\mathbf{t}^+,\mathbf{t}^-)[\log q(y|\mathbf{t}^+) - \log q(y|\mathbf{t}^-)] dy\right] \\ &\qquad \Rightarrow \mathbb{E}_{p(T^+,T^-)}\left[\int q(y|\mathbf{t}^+)[\log q(y|\mathbf{t}^+) - \log q(y|\mathbf{t}^-)] dy\right]. \end{aligned} \tag{3.16}$$

This term further goes to our final results as

$$\begin{aligned} &\mathbb{E}_{p(T^+,T^-)}\left[\int q(y|\mathbf{t}^+) \log q(y|\mathbf{t}^+) dy\right] - \mathbb{E}_{p(T^+,T^-)}\left[\int q(y|\mathbf{t}^+) \log q(y|\mathbf{t}^-) dy\right] \\ &\qquad \Rightarrow H_q(Y|T^+,Y|T^-) - H_q(Y|T^+), \end{aligned} \tag{3.17}$$

where the first term of the right hand side is the cross entropy between $q(Y|T^+)$ and $q(Y|T^-)$. We identify that the second term is in line with the maximum entropy principle[115-116] and the confidence penalty proposed by Pereyra et al.[117] that is imposed on the output distribution of deep neural networks. While out of our derivation, the confidence penalty is only imposed on the factual output $q(Y|T^+)$. We hence propose balanced learning by restricting the distance between factual and counterfactual posterior, which provably strengthens model's generalization over the underlying users' true preference distribution.

① Here we slightly abuse notation by denoting $H_{p,q}(Y|T) := \mathbb{E}_{x\sim p(X)}\mathbb{E}_{\mathbf{t}\sim p(T|X)} \int -p(y|x) \log q(y|\mathbf{t}) dy$.

② We may also pick $p(Y|T^+)$ as the approximation, which renders $H_{p,q}(Y|T^+,Y|T^-) - H_{p,q}(Y|T^+)$. However, it has less operational meaning and its last term cancels out with another task-aware term $I(T^+;Y)$.

Algorithm 3.1 Counterfactual Learning with CVIB in MNAR Data for Recommendation.

Data: Training factual set Ω^+, counterfactual set Ω^-; Hyperparameters α, β, γ
Result: Learned model parameters θ

```
Initialize model's parameters θ;
while Training loss keeps decreasing do
    Sample a batch of paired factuals X^+, Y^+ from Ω^+, and counterfactuals X^- from Ω^-;
    Compute the sufficiency term ① in Eq. (3.19) ;
    Compute the balancing term ② in Eq. (3.19) ;
    Compute the confidence penalty term ③ in Eq. (3.19) ;
    Compute the minimality term ④ in Eq. (3.19) ;
    Compute the batch objective loss as L̂_CVIB = ① + α② − γ③ + β④ ;
    Update the model parameters θ via stochastic gradient descent based on L̂_CVIB ;
end
```

3.5.3 Task-aware Information Term

We next omit the superscript of T^+ in $I(T^+;Y)$ for simplicity in the following. Similar to the operation of task-aware mutual information in Eq. (3.14), we use an approximation $q(Y|T)$ to substitute $p(Y|T)$

$$I(T;Y) \geq \mathbb{E}_{p(T,X)}\left[\int p(y|x)\log q(y|\mathbf{t})dy\right] \Rightarrow -I(T;Y) \leq H_{p,q}(Y|T), \tag{3.18}$$

which indicates that this term is a proxy of the cross entropy loss between the true outcome $p(Y|X)$ and the prediction $q(Y|T)$.

3.5.4 Algorithm Overview

Taking all the above derivations together, we conclude to the final objective function $\hat{L}_{CVIB}$, which encompasses four terms:

$$\hat{L}_{CVIB} = \underbrace{H_{p,q}(Y|T^+)}_{\text{① Sufficiency}} + \underbrace{\alpha H_{q,q}(Y|T^+, Y|T^-)}_{\text{② Balancing}} - \underbrace{\gamma H_q(Y|T^+)}_{\text{③ Penalty}} + \underbrace{\beta(\|\mu(X^+)\|_2^2 + \|\mu(X^-)\|_2^2)}_{\text{④ Minimality}} \tag{3.19}$$

where term ① denotes the cross entropy between $p(Y|T^+)$ and $q(Y|T^+)$; term ② is cross entropy between $q(Y|T^+)$ and $q(Y|T^-)$; and term ③ is entropy of $q(Y|T^+)$. For computing this objective, we need to draw x^+, y^+ from the factual set Ω^+ and x^- from the counterfactual set Ω^- in one iteration. Specifically, term ① is supervised loss on the factual outcomes; terms ②, ③ are contrastive regularization for balancing between factual and counterfactual domains; and term ④ is minimality loss to improve the model's ro-

bustness against policy bias. The whole optimization process over the $\hat{L}_{CVIB}$ is hence summarized in Algorithm 3.1, where we should specify the values of α, β and γ to balance these terms in optimization.

3.6 Experiments

Aiming to validate CVIB's effectiveness, we perform experiments on real-world datasets in this section. We elaborate on the experimental setups, and report the comparison results between CVIB and other baselines, which substantiate its all-round superiority in counterfactual learning.

3.6.1 Datasets

In order to evaluate the learned model's generalization ability on the underlying groundtruth distribution $p(x, y)$, rather than the only on the logged feedback, i.e., the factual domain, we usually need an additional MAR test set, where the users are served with uniformly displayed items, namely RCTs. As far as we know, there are only two open datasets that satisfy this requirement:

Yahoo! R3 Dataset[102]. This is a user-song rating dataset, where there are over 300K ratings self-selected by 15,400 users in its training set, hence it is MNAR data. Besides, they collect an additional MAR test set by asking 5,400 users to rate 10 randomly displayed songs.

Coat Shopping Dataset[105]. This dataset consists of 290 users and 300 items. Each user rates 24 items by themselves, and is asked to rate 16 uniformly displayed items as the MAR test set.

We simplify the underlying rating prediction problem to binary classification in our experiments, by making rating which is 3 or higher be positive feedback and those lower than 3 as negative.

3.6.2 Baselines

Our CVIB is model-agnostic, thus applicable for most models in recommendation that take embeddings to encode the events, including the shallow and deep models. In our experiments, we pick matrix factorization (MF)[118] as shallow backbone and neural collaborative filtering (NCF)[98] as the deep backbone. Both of these methods project users and items into a shared space and represent them with unique vectors, namely em-

beddings.

The most popular technique to debias MNAR data by involving RCTs is the inverse propensity score (IPS) method[105]. Its variants, the self-normalized IPS (SNIPS)[119], doubly robust (DR)[120] and joint learning doubly robust (DRJL)[121] are also widely used. In our experiments, we take 5% of test data to learn the propensity scores via a naive Bayes estimator[105]

$$p(O_{u,i} = 1|Y_{u,i} = y) = \frac{p(y|O = 1)p(O = 1)}{p(y)}, \tag{3.20}$$

for IPS, SNIPS, DR, and DRJL. Applying these methods to both shallow and deep backbones, we involve muliple baselines in comparison with our MF-CVIB and NCF-CVIB. Note that only CVIB is RCT-free among all methods.

3.6.3 Experimental Protocol

We implement all the methods on PyTorch[122]. For both the MF and NCF, we fix the embedding size of both users and items to be 4 because in our experiments, we find when embedding size gets larger, the performance of all methods on the MAR test set decays, which may be caused by overfitting. We randomly draw 30% data from the training set for validation, on which we apply a grid search for hyperparameters to pick the best configuration. Adam[123] is utilized as the optimizer for fast convergence during training, with its learning rate in $\{0.1, 0.05, 0.01, 0.005, 0.001\}$, weight decay in $\{10^{-3}, 10^{-4}, 10^{-5}\}$, and batch size in $\{128, 256, 512, 1024, 2048\}$. For NCF, we set an additional hidden layer with width 8. Specifically for CVIB, we set the hyperparameters $\alpha \in \{2, 1, 0.5, 0.1\}$, and $\gamma \in \{1, 0.1, 10^{-2}, 10^{-3}\}$. Since we already set weight decay for Adam, we do not apply the ℓ_2-norm term on the embeddings. After finding out the best configuration on the validation set, we evaluate the trained models on the MAR test set.

3.6.4 Results & Analysis

The overall evaluation results are reported in Table 3.2. There are three main observations:

(1) Even if they utilize additional RCTs, the IPS, SNIPS, DR and DRJL methods sometimes work worse than the naive model. By contrast, we identify that our RCT-free CVIB method is capable of enhancing both the shallow and deep models significantly in all experiments. It indicates that the contrastive regularization on the task-aware information, contained in embeddings of factual and counterfactual events, results in improvement

Table 3.2 MSE and AUC on the MAR test set of COAT[105] and YAHOO[102], where the best ones are in bold.

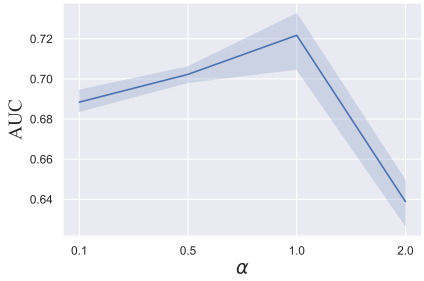
	COAT		YAHOO	
	MSE	AUC	MSE	AUC
MF	0.2451	0.7020	0.2493	0.6767
+IPS[105]	0.2299	0.7156	0.2260	0.6793
+SNIPS[119]	0.2374	0.6960	0.1945	0.6810
+DR[120]	0.2357	0.7058	0.2108	0.6883
+DRJL[121]	0.2423	0.6915	0.2745	0.6892
+CVIB (ours)	**0.2189**	**0.7218**	**0.1671**	**0.7198**
NCF	0.2030	0.7688	0.3313	0.6772
+IPS[105]	0.2008	0.7708	0.1777	0.6708
+SNIPS[119]	**0.1922**	0.7695	0.1699	0.6880
+DR[120]	0.2161	0.7514	**0.1698**	0.6886
+DRJL[121]	0.2097	0.7579	0.2789	0.6820
+CVIB (ours)	0.2017	**0.7713**	0.2820	**0.6989**

Table 3.3 Average nDCG with 10 runs on the MAR test set of COAT and YAHOO where the best ones are in bold.

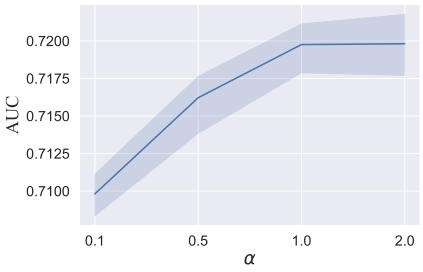
COAT	MF	IPS	SNIPS	DR	DRJL	CVIB
nDCG@5	0.589	0.633	0.603	0.622	0.608	**0.663**
nDCG@10	0.667	0.689	0.676	0.693	0.679	**0.721**
YAHOO	MF	IPS	SNIPS	DR	DRJL	CVIB
nDCG@5	0.633	0.636	0.635	0.659	0.652	**0.734**
nDCG@10	0.762	0.760	0.762	0.774	0.770	**0.820**

of model generalization ability. To further evaluate our method in terms of ranking quality, we report the results of nDCG in Table 3.3. CVIB shows more significant gain over the baselines than on AUC.

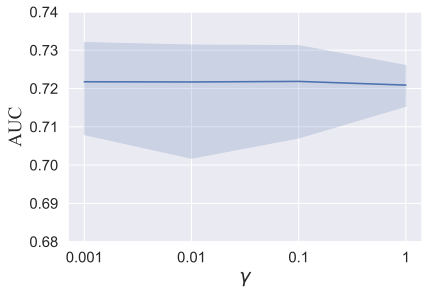
(2) We perform repeated experiments to quantify our CVIB's sensitivity to the balancing term weight α and confident penalty term weight γ in Eq. (3.19). From Fig. 3.7 we identify that α influences results significantly on COAT, while dose not make much difference on YAHOO. In general, increasing α enhances the test AUC, which is aligned



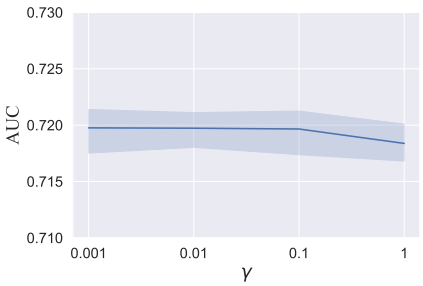
(a) COAT ($\gamma = 10^{-3}$)



(b) YAHOO ($\gamma = 10^{-3}$)



(c) COAT ($\alpha = 1$)



(d) YAHOO ($\alpha = 1$)

Figure 3.7 Test results of MF-CVIB with varying α and γ. Shaded regions show the 90% confidence intervals of the test AUC.

with our argument that contrastive information term benefits in balancing model between factual and counterfactual domains and then leads to better generalization ability. The confidence penalty γ plays less role in accuracy, but we should not set it too high to avoid underfitting.

(3) One would notice that the performance of NCF-CVIB in terms of MSE is relatively weak, while we argue that good recommendation does not necessarily rely on low MSE prediction. For instance, for the true outcomes $y = \{1, 0, 0, 0\}$, the predictions $\hat{y}_1 = \{0, 0.1, 0.1, 0.1\}$ have much lower MSE than $\hat{y}_2 = \{0.6, 0.4, 0.4, 0.4\}$, although the former is obviously a worse prediction than the latter in terms of ranking. It turns out that NCF-CVIB overestimates the outcomes on the test set of YAHOO, nevertheless it still reaches the best ranking quality measured by AUC. In practice of recommender systems, instead of low MSE, we would rather appreciate high ranking quality, namely AUC, which demonstrates the model's capability of finding out the positive examples.

3.7 Chapter Summary

In this chapter, we focused on adapting information bottleneck principle for learning recommender system models that are able to precisely rank the items according to the users' true preferences. As the missing-not-at-random (MNAR) effect is ubiquitous in

modern recommender systems, the original IB based methods are no longer best suited. Missing-at-random (MAR) data, namely randomized controlled trials (RCTs), are usually required by most previous counterfactual learning methods for debiasing learning. However, the execution of RCTs is extraordinarily expensive in practice. To circumvent the use of RCTs, we build an information-theoretic counterfactual variational information bottleneck (CVIB), as an alternative for debiasing learning without RCTs.

Techinically, we proposed an information theoretic counterfactual learning framework in recommender systems. By separating the task-aware sufficiency term in the original information bottleneck objective into factual and counterfactual parts, we derived a contrastive information regularizer and a output confidence penalty, thus obtaining a novel CVIB objective function. The paradigm of information contraction encourages the learned model to be balanced between the factual and counterfactual domains, therefore improves its generalization ability significantly, which is validated by our empirical experiments. Our CVIB provides insight in utilizing information theoretic representation learning methods in recommendation, and sheds light on debiasing learning under MNAR data in the absence of expensive RCTs. In summary, our contributions are as follows:

- We establish a novel CVIB framework adapted from IB, which suggests new avenues in counterfactual learning from MNAR data in an RCT-free paradigm.
- A novel solution is proposed to handle the optimization of CVIB on MNAR data, which specifies a contrastive information term associated with a minimality term.
- We empirically investigate our method's advantages on real-world datasets for correcting MNAR bias without the need of acquiring the RCTs. Code is available at https://github.com/RyanWangZf/CVIB-Rec.

CHAPTER 4 INFORMATION BOTTLENECK WITH PAC-BAYES GENERALIZATION GUARANTEE

Information bottleneck theory is promising for explaining the black-box behavior of deep neural networks during training. Four main claims were made in IB theory: First, the training of DNNs consists of an initial fitting phase and the a subsequent compression phase; Second, the representation compression has causal relationship with the generalization capacity of DNNs; Third, the representation compression is achieved by the random diffusion of stochastic gradient descents after the initial fitting phase; And last, the ultimate goal of DNNs is to optimize the IB trade-off between compression and prediction.

However, a series of recent critics towards the classical IB theory intimidate the universality of those claims (please refer to the last paragraph of §2.3 and next section §4.1 for the details of these critics). In this work, the following three research questions are of our interests:

- Is the two-phase training behaviour of DNNs universal in real practices? If this claim is invalid with the previously proposed representation based information measure $I(T;X)$, how do we validate this claim through another information theoretic perspective?
- As the representation based information $I(T;X)$ was doubted not being causally related to generalization, how do we find another measure with theoretical generalization guarantee? Also, how do we make use of this new measure for facilitating the information bottleneck principle?
- With the new information bottleneck at hand, how do we utilize it for efficient training and inference of DNNs in practice?

In the following sections, we first discuss the caveats of the representation-based information bottleneck, then we elaborate on our methodology to the above questions point-by-point.

4.1 On Caveats of Representation-based Information Bottleneck

Since the proposal of information bottleneck theory, there appeared heat discussions and critics towards the claims of IB's capability of explaining DNNs. At first, Shwartz-

Ziv et al.[14] demonstrated a clear boundary between the fitting and compression phase where training error becomes small and the random diffusion of SGD dominates the learning process. However, Saxe et al.[64] argued that this phenomenon only appears when the double-sided saturating nonlinearities like tanh are deployed. When the activation function is one of those linear or single-sided saturating nonlinearities like ReLU, no apparent compression phase can be observed. The boundary between fitting and compression phase fades away, i.e., compression happens simultaneously with the fitting process, hence IB does not offer meaningful insight in DNNs learning process any more.

Second, the claimed causality between compression and generalization was also denounced[23,64], i.e., networks that do not compress still generalize well, and vice versa. Goldfeld et al.[23] identified that networks with orthonormal weight regularization term can generalize well completely free of compression. Instead, they proposed that the clustering of hidden representations concurrently occurs with good generalization ability. However, this new proposal still lacks solid theoretical guarantee.

Third, the mutual information term becomes trivial in deterministic cases as described by Eq. (2.2). Some other problems encountered in deterministic cases are also pointed out by Kolchinsky et al.[22]. Although several techniques, e.g., bining and adding noise, are adopted for making stochastic approximation for the information term, they might either violate the principle of IB or be contradictory to the high performance of DNNs. We need to design efficient stochastic realization of DNNs both for estimating information and for better inference.

4.2 A New Bottleneck with PAC-Bayes Guarantee

Recall the representation-based information bottleneck $\max_T I(T;Y)-\beta I(X;T)$ that is built on the information complexity of representation $I(X;T)$. We discussed its caveats in last section that it is not causally related to generalization, which renders many follow-ups on deriving new information-theoretic generalization measure. One of the most generalization measure is the information contained in model parameters $I(\mathbf{w};S)$. Let's recap the basic conception of generalization discussed in §2.4.

A loss function $\ell(f^{\mathbf{w}}(X),Y)$ is a measure of the degree of prediction accuracy $f^{\mathbf{w}}(X)$ compared with the groundtruth label Y. Given the groundtruth joint distribution $p(X,Y)$, the expected true risk (out-of-sample risk) is taken on expectation as

$$L(\mathbf{w}) \triangleq \mathbb{E}_{p(\mathbf{w}|S)}\mathbb{E}_{p(X,Y)}[\ell(f^{\mathbf{w}}(X),Y)]. \tag{4.1}$$

It should be noted here we take an additional expectation over $p(\mathbf{w}|S)$ because we are evaluating risk of the learned posterior instead of a single parameter $\mathbf{w}$. In addition, we call $p(\mathbf{w}|S)$ posterior here for convenience while it is not the Bayesian posterior that is computed through Bayes theorem $p(\mathbf{w}|S) = \frac{p(\mathbf{w})p(S|\mathbf{w})}{p(S)}$. The PAC-Bayes bounds introduced later hold even if prior $p(\mathbf{w})$ is incorrect and arbitrarily chosen posterior $p(\mathbf{w}|S)$.

In practice, we only own finite samples for the purpose of learning, namely the training data $S = \{X_i, Y_i\}_{i=1}^n$ with n i.i.d. samples drawn from $p(X, Y)$. This gives rise to the empirical risk defined on S as

$$L_S(\mathbf{w}) = \mathbb{E}_{p(\mathbf{w}|S)}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(f^{\mathbf{w}}(X_i), Y_i)\right]. \tag{4.2}$$

With the above Eq. (4.1) and Eq. (4.2) at hand, the generalization gap when the learned posterior $p(\mathbf{w}|S)$ is testified in out-of-sample is

$$\Delta L(\mathbf{w}) \triangleq L(\mathbf{w}) - L_S(\mathbf{w}). \tag{4.3}$$

Xu et al.[83] proposed a novel PAC-Bayes bound based on the information contained in weights $I(\mathbf{w}; S)$ that

$$\mathbb{E}_{p(S)}[L(\mathbf{w}) - L_S(\mathbf{w})] \leq \sqrt{\frac{2\sigma^2}{n} I(\mathbf{w}; S)}, \tag{4.4}$$

when $\ell(f^{\mathbf{w}}(X), Y)$ is σ-sub-Gaussian; A series of following works also tightened this bound[84-87] and verified it is an effective measure of generalization capability of learning algorithms. Therefore, it is natural to build a new information bottleneck grounded on this PAC-Bayes generalization measure, namely the PAC-Bayes information bottleneck (PIB), as

$$\min_{p(\mathbf{w}|S)} \mathcal{L}_{\text{PIB}} = L_S(\mathbf{w}) + \beta I(\mathbf{w}; S). \tag{4.5}$$

In classification, the loss function term $L_S(\mathbf{w})$ is usually the expectation of negative log-likelihood $\mathbb{E}_{p(\mathbf{w}|S)}[-\log p(S|\mathbf{w})]$, hence the PIB in Eq. (4.5) is equivalent to

$$\max_{p(\mathbf{w}|S)} \mathbb{E}_{p(\mathbf{w}|S)}[\log p(S|\mathbf{w})] - \beta I(\mathbf{w}; S), \tag{4.6}$$

which demonstrates a trade-off between maximizing the log-likelihood on the data S and minimizing information compression of learned parameters $\mathbf{w}$.

4.3 Estimating Information Stored in Weights

4.3.1 Closed-form Solution with Gaussian Assumption

By deriving a new information bottleneck PIB, we can look into the learning process of DNNs in another information-theoretic perspective, i.e., how $I(\mathbf{w}; S)$ and $L_S(\mathbf{w})$ evolves during the learning process of DNNs with SGD optimization. We are wondering if the two-phase transition phenomenon can be re-identified as well as IB does. Now the key challenge ahead is how to estimate the information stored in weights $I(\mathbf{w}; S)$, as

$$I(\mathbf{w}; S) = \mathbb{E}_{p(S)}[\mathrm{KL}(p(\mathbf{w}|S) \parallel p(\mathbf{w}))] \tag{4.7}$$

is the expectation of KL divergence between $p(\mathbf{w}|S)$ and $p(\mathbf{w})$ over the distribution of dataset $p(S)$. $p(\mathbf{w})$ is the marginal distribution of $p(\mathbf{w}|S)$ by definition of mutual information, hence

$$p(\mathbf{w}) \triangleq \mathbb{E}_{p(S)}[p(\mathbf{w}|S)]. \tag{4.8}$$

When we assume both $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_0)$ and $p(\mathbf{w}|S) = \mathcal{N}(\mathbf{w}|\boldsymbol{\theta}_S, \boldsymbol{\Sigma}_S)$ are Gaussian distributions, the KL divergence term in Eq. (4.7) has closed-form solution as

$$\begin{aligned} &\mathrm{KL}(p(\mathbf{w}|S) \parallel p(\mathbf{w})) = \mathbb{E}_{p(\mathbf{w}|S)}[\log p(\mathbf{w}|S) - \log p(\mathbf{w})] \\ &= \frac{1}{2}\left[\log \frac{\det \boldsymbol{\Sigma}_S}{\det \boldsymbol{\Sigma}_0} - D + (\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0) + \mathrm{tr}\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_S\right)\right]. \end{aligned} \tag{4.9}$$

$\det \mathbf{A}$ is the determinant of matrix A; D is the number of dimension of parameter $\mathbf{w}$ and is a constant after the DNN architecture built; $\boldsymbol{\theta}_S$ are the yielded weights after SGD converges on a dataset S. When we further assume that $\boldsymbol{\Sigma}_S = \gamma \boldsymbol{\Sigma}_0$, the logarithmic and trace terms in Eq. (4.9) all become constant that only depends on γ. Herein the mutual information term is proportional to the quadratic term such that

$$I(\mathbf{w}; S) \propto \mathbb{E}_{p(S)}\left[(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)\right] = \mathbb{E}_{p(S)}\left[\boldsymbol{\theta}_S^\top \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\theta}_S\right] - \boldsymbol{\theta}_0^\top \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\theta}_0. \tag{4.10}$$

Now we will see how to set prior covariance $\boldsymbol{\Sigma}_0$.

4.3.2 Bootstrap Covariance of Oracle Prior

In previous PAC-Bayes practice, $\boldsymbol{\Sigma}_0$ is often assumed to be isotropic Gaussian or diagonal Gaussian for the sake of tractability. We doubt it is inappropriate and often causes the information measure meaningless for practical use. Since the computation of exact oracle prior needs the knowledge of $p(S)$, we propose to approximate it by *bootstrapping*

from S, that is

$$\boldsymbol{\Sigma}_0 = \mathbb{E}_{p(S)}\left[(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top\right] \simeq \frac{1}{K}\sum_k (\boldsymbol{\theta}_{S_k} - \boldsymbol{\theta}_S)(\boldsymbol{\theta}_{S_k} - \boldsymbol{\theta}_S)^\top, \tag{4.11}$$

where S_k is a bootstrap sample obtained by re-sampling from the finite data S, and $S_k \sim p(S)$ is still a valid sample follows $p(S)$. Now we are closer to the solution but the above term is still troublesome to calculate. For getting $\{\theta_{S_k}\}_{k=1}^K$, we need to optimize on a series of bootstrapping datasets $\{S_k\}_{k=1}^K$ via SGD until it converges for K times, which is prohibitive in deep learning practices. Therefore, we propose to approximate the difference $\boldsymbol{\theta}_S - \boldsymbol{\theta}_0$ it by *influence functions* drawn from robust statistics literature[124-126].

Lemma 4.1 (Influence function[124-125]): Given a dataset $S = \{Z_i\}_{i=1}^n$ and the parameter $\hat{\boldsymbol{\theta}}_S \triangleq \operatorname{argmin}_{\boldsymbol{\theta}} L_S(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{\theta})$ ① that optimizes the empirical loss function. If we drop sample Z_j in S to get a jackknife sample $S_{\setminus j}$ and retrain our model, the new parameters are

$$\hat{\boldsymbol{\theta}}_{S_{\setminus j}} = \operatorname*{argmin}_{\boldsymbol{\theta}} L_{S_{\setminus j}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{\theta}) - \frac{1}{n}\ell_j(\boldsymbol{\theta}), \tag{4.12}$$

the parameter difference $\hat{\boldsymbol{\theta}}_{S_{\setminus j}} - \hat{\boldsymbol{\theta}}_S$ can be approximated without leave-one-out retraining by influence function $\boldsymbol{\psi}$, as

$$\hat{\boldsymbol{\theta}}_{S_{\setminus j}} - \hat{\boldsymbol{\theta}}_S \simeq -\frac{1}{n}\boldsymbol{\psi}_j, \text{ where } \boldsymbol{\psi}_j = -\mathbf{H}_{\hat{\boldsymbol{\theta}}_S}^{-1}\nabla_{\boldsymbol{\theta}}\ell_j(\hat{\boldsymbol{\theta}}_S) \in \mathbb{R}^D, \tag{4.13}$$

and $\mathbf{H}_{\hat{\boldsymbol{\theta}}_S} \triangleq \frac{1}{n}\sum_{i=1}^n \nabla_{\boldsymbol{\theta}}^2 \ell_i(\hat{\boldsymbol{\theta}}_S) \in \mathbb{R}^{D\times D}$ is Hessian matrix and is positive definite (PD) by assumption that loss function $\ell(\boldsymbol{\theta})$ is twice-differentiable and strictly convex in $\boldsymbol{\theta}$.

The application of influence functions can be further extended to the case when the loss function is perturbed by a vector $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)^\top \in \mathbb{R}^n$ as

$$\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}} = \operatorname*{argmin}_{\boldsymbol{\theta}} L_S(\boldsymbol{\theta}, \boldsymbol{\xi}) = \frac{1}{n}\sum_{i=1}^n \xi_i \ell_i(\boldsymbol{\theta}). \tag{4.14}$$

In this scenario, the parameter difference can be approximated by

$$\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}} - \hat{\boldsymbol{\theta}}_S \simeq \frac{1}{n}\sum_{i=1}^n \left(\xi_i - 1\right)\boldsymbol{\psi}_i = \frac{1}{n}\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1}), \tag{4.15}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n)^\top \in \mathbb{R}^{n\times D}$ is a combination of all influence functions $\boldsymbol{\psi}$; $\mathbf{1} = (1, 1, \ldots, 1)^\top$ is an n-dimensional all-one vector.

Besides influence functions, we need another lemma on bootstrapping performance in big data. When bootstrap resampling from dataset S, each individual sample Z_i has a

① Note $L_S(\boldsymbol{\theta})$ is not the expected empirical risk $L_S(\mathbf{w})$ in Eq. (4.2), instead, it is the deterministic empirical risk that only relates to the mean parameter $\boldsymbol{\theta}$. We also denote $\ell(f^{\boldsymbol{\theta}}(X_i), Y_i)$ by $\ell_i(\boldsymbol{\theta})$ for the notation conciseness.

probability of $\frac{1}{n}$ being picked, causing the weight ξ_i a binomial distribution as

$$\xi_i \sim \text{Binomial}\left(n, \frac{1}{n}\right). \tag{4.16}$$

As a result, all weights $\boldsymbol{\xi}$ follows a multinomial distribution as $\boldsymbol{\xi} \sim \text{Multinomial}\left(n, \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$ with the total number of samples constrained to be n. When it comes to big data, i.e., n is prohibitively large, this multinomial resampling scales w.r.t. sample size n thus becomes rather slow. Therefore, we propose to utilize *Poisson bootstrapping* as a substitute:

Lemma 4.2 (Poisson Bootstrapping[127-128]): Given infinite number of samples, the bootstrap resampling weight ξ has the property that

$$\lim_{n\to\infty} \text{Binomial}\left(n, \frac{1}{n}\right) = \text{Poisson}(1). \tag{4.17}$$

This approximation becomes precise in practice when $n \geq 100$. Also, we know $\mathbb{E}[\xi_i] = 1$ and $\text{Var}[\xi_i] = 1$ by the definition of Poisson distribution when n is large enough.

Based on Lemma 4.1 and Lemma 4.2, we are now able to give our lemma on approximation of oracle prior covariance in Eq. (4.11):

Lemma 4.3 (Approximation of Oracle Prior Covariance): Given the definition of influence functions (Lemma 4.1) and Poisson bootstrapping (Lemma 4.2), the covariance matrix of the oracle prior can be approximated by

$$\begin{aligned}\boldsymbol{\Sigma}_0 = \mathbb{E}_{p(S)}\left[(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top\right] &\simeq \frac{1}{K}\sum_{k=1}^{K}\left(\hat{\boldsymbol{\theta}}_{\boldsymbol{\xi}^k} - \hat{\boldsymbol{\theta}}\right)\left(\hat{\boldsymbol{\theta}}_{\boldsymbol{\xi}^k} - \hat{\boldsymbol{\theta}}\right)^\top \\ &\simeq \frac{1}{n}\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{F}_{\hat{\boldsymbol{\theta}}}\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1} \simeq \frac{1}{n}\mathbf{F}_{\hat{\boldsymbol{\theta}}}^{-1},\end{aligned} \tag{4.18}$$

where we omit the subscript S of $\hat{\boldsymbol{\theta}}_S$ and $\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}}$ for notation conciseness, and $\boldsymbol{\xi}^k$ is the bootstrap resampling weight in the k-th experiment.

Please refer to Appendix B.1 for the proof of this lemma.

4.3.3 Efficient Information Estimation Algorithm

After obtain the approximation of oracle prior covariance, we are now able to rewrite the information $I(\mathbf{w}; S)$ in Eq. (4.10) to

$$\widetilde{I}(\mathbf{w}; S) = n\mathbb{E}_{p(S)}\left[(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top \mathbf{F}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)\right] \simeq n(\bar{\boldsymbol{\theta}}_S - \boldsymbol{\theta}_0)^\top \mathbf{F}_{\hat{\boldsymbol{\theta}}}(\bar{\boldsymbol{\theta}}_S - \boldsymbol{\theta}_0). \tag{4.19}$$

We omit the constant terms in $I(\mathbf{w}; S)$ and denote the rest term by $\widetilde{I}(\mathbf{w}; S)$. We approximate the expectation $\mathbb{E}_{p(S)}[\boldsymbol{\theta}_S^\top F_{\hat{\boldsymbol{\theta}}}\boldsymbol{\theta}_S]$ by taking an quadratic mean of model parameters

Algorithm 4.1 Efficient approximate information estimation

Data: Total number of samples n, batch size B, learning rate η, moving average hyperparameters ρ and K
Result: Calculated approximate information $\widetilde{I}(\mathbf{w}; S)$

Pretrain the model by vanilla SGD to obtain the prior mean $\boldsymbol{\theta}_0$;
Initialize a sequence of gradients set $\nabla\mathcal{L} = \varnothing$;
repeat
  $\nabla L_t \leftarrow \nabla_{\boldsymbol{\theta}} \frac{1}{B} \sum_b \ell_b(\hat{\boldsymbol{\theta}}_{t-1})$; `/* Compute minibatch gradient */`
  $\hat{\boldsymbol{\theta}}_t \leftarrow \hat{\boldsymbol{\theta}}_{t-1} - \eta \nabla L_t$; `/* Vanilla SGD */`
  $\nabla\mathcal{L} \leftarrow \nabla\mathcal{L} \bigcup \{\nabla L_t)\}$; `/* Store gradients */`
  $\bar{\boldsymbol{\theta}}_t \leftarrow \sqrt{\rho \bar{\boldsymbol{\theta}}^2_{t-1} + \frac{1-\rho}{K} \sum_{k=0}^{K-1} \hat{\boldsymbol{\theta}}^2_{t-k}}$; `/* Moving average */`
until *Go across all mini-batches within this epoch*;
`/* Efficient computation of approximated information */`
$\Delta\boldsymbol{\theta} \leftarrow \hat{\boldsymbol{\theta}}_T - \bar{\boldsymbol{\theta}}_T,\ \Delta\mathbf{F}_0 \leftarrow 0$;
for *t=1:T* **do**
  $\Delta\mathbf{F}_t \leftarrow \Delta\mathbf{F}_{t-1} + \Delta\boldsymbol{\theta}^\top \nabla L_t$; `/* Storage-friendly computation */`
end
$\widetilde{I}(\mathbf{w}; S) \leftarrow \frac{n}{T} \Delta\mathbf{F}_T^2$;

during SGD updates, as

$$\bar{\boldsymbol{\theta}}_S = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_k^2} = \left(\sqrt{\frac{1}{K}\sum_{k=1}^{K} \hat{\theta}_{1,k}^2}, \sqrt{\frac{1}{K}\sum_{k=1}^{K} \hat{\theta}_{2,k}^2}, \dots, \sqrt{\frac{1}{K}\sum_{k=1}^{K} \hat{\theta}_{D,k}^2} \right)^\top \tag{4.20}$$

to yield the final information measure. Hereinafter, we encapsulate our algorithm for estimating information stored in weights during SGD by Algorithm 4.1.

In Eq. (4.19), the information consists of two major components, $\bar{\boldsymbol{\theta}}_S - \boldsymbol{\theta}_0 \in \mathbb{R}^D$ and the Fisher information matrix (FIM) $\mathbf{F}_{\hat{\boldsymbol{\theta}}} \in \mathbb{R}^{D\times D}$, which can easily cause out-of-memory error due to the high-dimensional matrix product operations. We therefore hack into FIM to get

$$\begin{aligned} \widetilde{I}(\mathbf{w}; S) &= n\Delta\boldsymbol{\theta}^\top \left[\frac{1}{T} \sum_{t=1}^{T} \nabla_{\boldsymbol{\theta}} \ell_t(\hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\theta}} \ell_t^\top(\hat{\boldsymbol{\theta}}) \right] \Delta\boldsymbol{\theta} \\ &= \frac{n}{T} \sum_{t=1}^{T} \left[\Delta\boldsymbol{\theta}^\top \nabla_{\boldsymbol{\theta}} \ell_t(\hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\theta}} \ell_t^\top(\hat{\boldsymbol{\theta}}) \Delta\boldsymbol{\theta} \right] \\ &= \frac{n}{T} \sum_{t=1}^{T} \left[\Delta\boldsymbol{\theta}^\top \nabla_{\boldsymbol{\theta}} \ell_t(\hat{\boldsymbol{\theta}}) \right]^2, \end{aligned} \tag{4.21}$$

such that the high dimensional matrix vector product reduces to vector inner product, which enables its application in enormous DNNs.

Here we introduce a new notion of information stored in weights in DNNs. This measure is built on Fisher information matrix that relates to the flatness of Remannian manifold. Unlike Hessian eigenvalues of loss functions which are used for identifying flat local minima and generalization but can be made arbitrarily large[129], this notion is invariant to re-parameterization of DNNs. Also, our measure is invariant to the choice of non-linearieties because it is not amortized, i.e., it is not directly influenced by input X like $I(T;X)$. We corroborate that it is capable of reproducing the two-phase transition for varying non-linearities (e.g., ReLU, Linear, Tanh, etc.) in §4.5.1.

4.4 Bayesian Inference for the Optimal Posterior

Recall that we design a new bottleneck on the expected generalization gap drawn from PAC-Bayes theory in §4.2, and then derive an approximation of the information stored in weights in §4.3. The two components of our PAC-Bayes IB in Eq. (4.6) are tractable as a learning objective. We give the following lemma on utilizing it for inference.

Lemma 4.4 (Optimal Posterior for PAC-Bayes Information Bottleneck): Given an observed dataset S^*, the optimal posterior $p(\mathbf{w}|S^*)$ of PAC-Bayes IB in Eq. (4.5) should satisfy the following form that

$$p(\mathbf{w}|S^*) = \frac{1}{Z(S)} p(\mathbf{w}) \exp\left\{-\frac{1}{\beta}\hat{L}_{S^*}(\mathbf{w})\right\} = \frac{1}{Z(S)} \exp\left\{-\frac{1}{\beta}\left(U_{S^*}(\mathbf{w})\right)\right\}, \tag{4.22}$$

where $U_{S^*}(\mathbf{w})$ is the energy function defined by

$$U_{S^*}(\mathbf{w}) = \hat{L}_{S^*}(\mathbf{w}) - \beta \log p(\mathbf{w}), \tag{4.23}$$

and $Z(S)$ is the normalizing constant.

Please refer to Appendix B.2 for the proof. The reason why we write the posterior in terms of an exponential form is that it is a typical *Gibbs distribution*[130] (also called Boltzmann distribution) with *energy function* $U_{S^*}(\mathbf{w})$ and *temperature* β. Crediting to this formula, we are able to adopt Markov chain Monte Carlo (MCMC) for rather efficient Bayesian inference. Specifically, we propose to use stochastic gradient Langevin dynamics (SGLD)[33] that has been proved efficient and effective in large scale posterior inference. SGLD can be realized by a simple adaption of SGD as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \mathbf{g}_k + \sqrt{2\eta_k\beta}\varepsilon_k, \tag{4.24}$$

where η_k is step size, $\varepsilon_k \sim \mathcal{N}(\varepsilon|\mathbf{0}, \mathbf{I}_D)$ is a standard Gaussian noise vector, and $\mathbf{g}_k$ is an unbiased estimate of energy function gradient $\nabla U(\mathbf{w}_k)$. SGLD can be viewed as a discrete

Algorithm 4.2 Optimal Gibbs Posterior Inference by SGLD.

Data: Total number of samples n, batch size B, learning rate η, temperature β
Result: A sequence of weights $\{\mathbf{w}_t\}_{t\geq\hat{k}}$ following $p(\mathbf{w}|S^*)$

repeat
  /* Stochastic gradients of energy function */
  $\nabla\widetilde{U}_{S^*}(\mathbf{w}_{t-1}) \leftarrow \nabla\left(-\frac{B}{n}\sum_b \log p(Y_b|X_b,\mathbf{w}_{t-1}) - \beta_{t-1}\log p(\mathbf{w}_{t-1})\right)$;
  /* Weight update by SGLD */
  $\varepsilon_t \leftarrow \mathcal{N}(\varepsilon|\mathbf{0},\mathbf{I}_D)$;
  $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \eta_{t-1}\nabla\widetilde{U}_{S^*}(\mathbf{w}_{t-1}) + \sqrt{2\eta_{t-1}\beta_{t-1}}\varepsilon_t$;
  /* Decay and annealing */
  $\eta_t \leftarrow \phi_\eta(\eta_{t-1})$, $\beta_t \leftarrow \phi_\beta(\beta_{t-1})$;
until *The weight sequence* $\{\mathbf{w}_t\}_{t\geq\hat{k}}$ *becomes stable*;

Langevin diffusion described by stochastic differential equation[41,131]:

$$d\mathbf{w}(t) = -\nabla U(\mathbf{w}(t))dt + \sqrt{2\beta}d\boldsymbol{B}(t), \tag{4.25}$$

where $\{\boldsymbol{B}(t)\}_{t\geq 0}$ is the standard Brownian motion in $\mathbb{R}^D$. The Gibbs distribution $\pi(\mathbf{w}) \propto \exp(-\frac{1}{\beta}U(\mathbf{w}))$ is the unique invariant distribution of Eq. (4.25). And that distribution of $\mathbf{w}_t$ converges rapidly to $\pi(\mathbf{w})$ when $t \to \infty$ with sufficiently small β[132]. Similarly for SGLD in Eq. (4.24), under the conditions that

$$\sum_t^\infty \eta_t \to \infty \text{ and } \sum_t^\infty \eta_t^2 \to 0, \tag{4.26}$$

and an annealing temperature β, the sequence of $\{\mathbf{w}_k\}_{k\geq\hat{k}}$ converges to Gibbs distribution with sufficiently large $\hat{k}$.

As we assume the oracle prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_0)$, $\log p(\mathbf{w})$ in Eq. (4.23) satisfies

$$-\log p(\mathbf{w}) \propto (\mathbf{w}-\boldsymbol{\theta}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\mathbf{w}-\boldsymbol{\theta}_0) + \log(\det \boldsymbol{\Sigma}_0). \tag{4.27}$$

The algorithm for inference of the optimal posterior is then given by Algorithm 4.2. $\phi_\eta(\cdot)$ and $\phi_\beta(\cdot)$ are learning rate decay and temperature annealing functions, respectively. They can be linear, cosine, or stage-wise decay functions. Our SGLD based algorithm leverages the advantage of MCMC such that is capable of sampling from the optimal posterior ignorant of complexity of DNNs. It can be realized with a minimal adaptation of common auto-differentiation packages, e.g., PyTorch[122], by injecting isotropic noise in the SGD function.

4.5 Experiments

In this section, we aim to verify the effectiveness of the proposed information measure. We are interested in its performance under various DNNs settings for the sake of different non-linearities (§4.5.1), architecture (§4.5.2), noise ratio (§4.5.3), and batch size (§4.5.4). We also substantiate the superiority of optimal Gibbs posterior inference based on the proposed algorithm, where the information stored in weights is proved an effective regularization for training DNNs (§4.5.5). We conclude the empirical observations in §4.5.6 at last.

All experiments are conducted on MNIST[38] or on CIFAR-10[133]. We design two multi-layer perceptron (MLPs): MLP-Small and MLP-Large, where MLP-Small is a two-layer DNN, i.e., 784(3072)-512-10, and MLP-Large is a five-layer DNN, i.e., 784(3072)-100-80-60-40-10. The number of input units is 784 with permutation MNIST inputs or 3072 with permutation CIFAR-10 inputs. In the general setting, we pick Adam optimizer[123] to boost the convergence of DNN training.

4.5.1 Information with Different Non-linearities

One pitfall of previous representation-based IB is that it does explain the DNNs equipped with other non-linearities instead of tanh. When other non-linearities are used, e.g., sigmoid, relu, etc., the phase transition phenomenon does not appear in experiments. Here, we try to train MLP-Small on plain cross entropy loss by Adam on MNIST dataset meanwhile monitor the trajectory of our new information measure $I(\mathbf{w}; S)$. Results are illustrated in Fig. 4.1 where four different non-linearities are testified: linear, tanh, ReLU, and sigmoid.

For all four non-linearities, we identify that there is a clear boundary between fitting and compression phase. For example, for the linear on the first row, the information $I(\mathbf{w}; S)$ increases dramatically within the first 20 iterations then decreases slowly during the next 200 iterations. At the same time, we could see that the training loss on the right reduces sharply at the initial stage, then keeps decreasing simultaneous to the information compression. This result tells us that the rise and fall of information in weights keeps pace to the loss. In other words, as the compression of $I(\mathbf{w}; S)$ is deeply connected to the generalization of DNNs according to PAC-Bayes theorem, our finding verifies this connection empirically.

Although the compression of information in representations $I(T; X)$ is completely

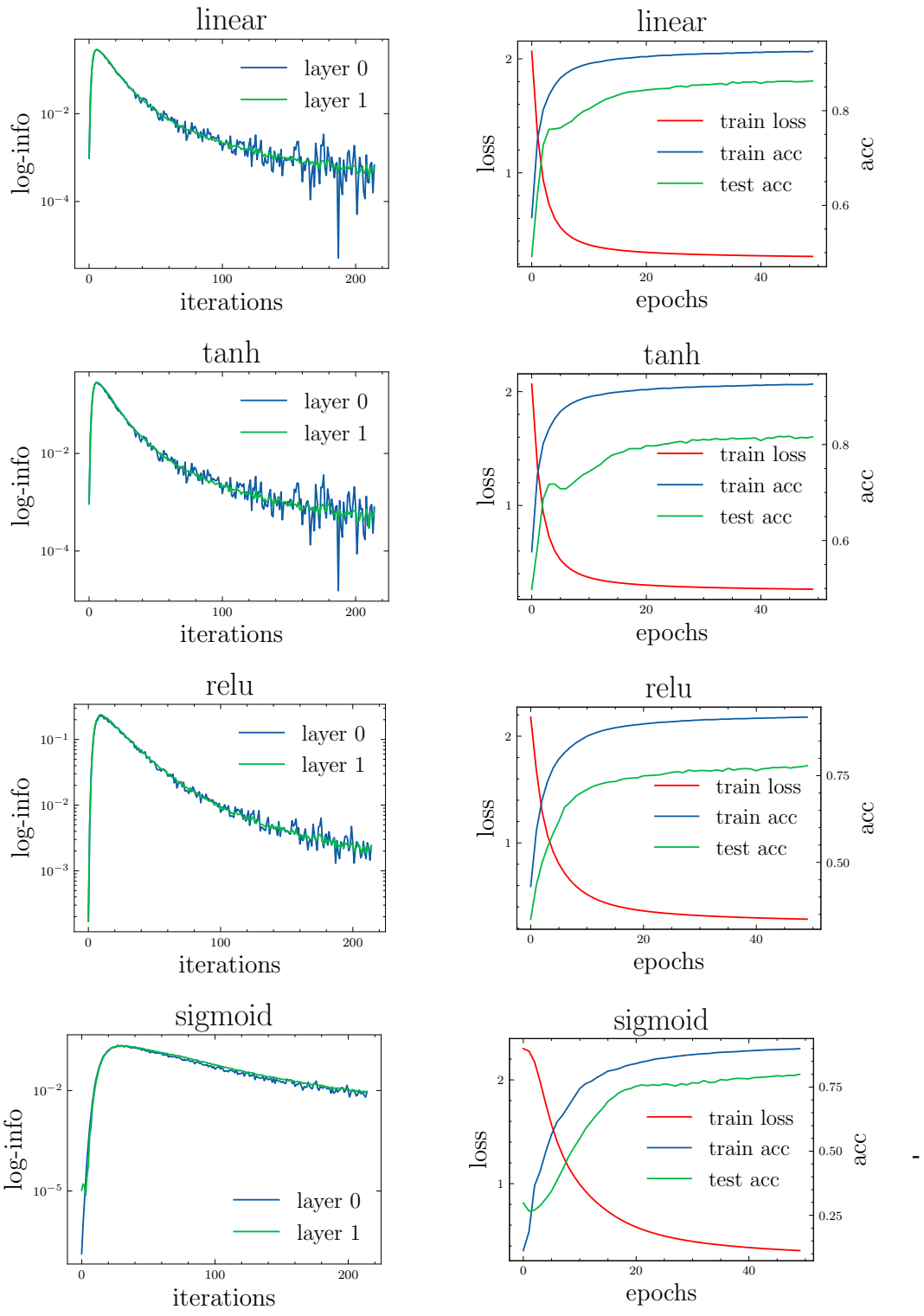


Figure 4.1 Information stored in weights (left), loss and accuracy (right) of DNNs trained with different non-linearities (linear, tanh, ReLU, and sigmoid). The y-axis of information is in logarithmic scale for better display.

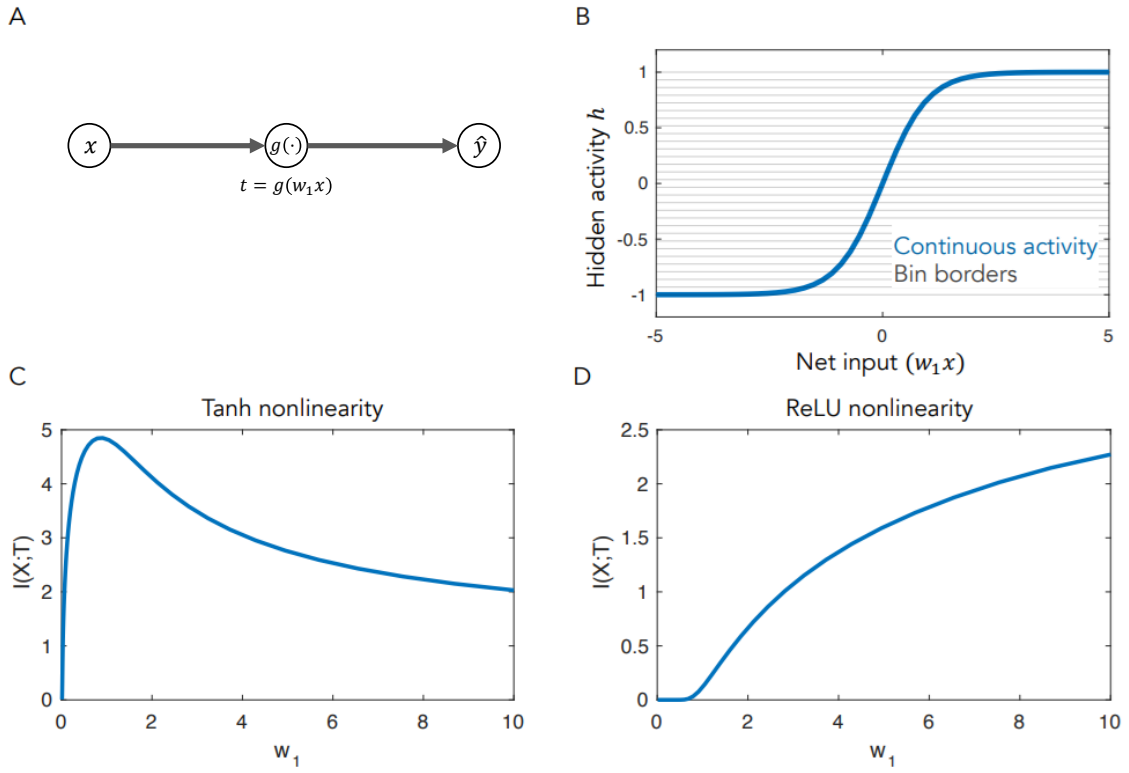


Figure 4.2 Nonlinear compression of a minimal model. (A) A three neuron MLP with Gaussian inputs x, weight w_1, and non-linearity $g(\cdot)$. (B) The representation t produced by tanh activation is binned into a discrete variable T_{binned} for computing information. (C) and (D) are information when tanh and ReLU are picked. This figure is reproduced from [64].

gone for non-saturation activations like ReLU, our measure $I(\mathbf{w}; S)$ shows great universality on digging out the information compression of DNNs. The representation based information $I(T; X)$ is sensitive to the selection of non-linearities due to the amortized inference, i.e., the representation T is a function of inputs X and activation function $g(\cdot)$. A experiment of Saxe et al.[64] shows that the compression of representation based information is subtle to activations, by Fig. 4.2. When we try to evaluate $I(T; X)$ by binning, the continuous tanh non-linear activations are binned into 30 bins evenly spaced between -1 and 1 in (B). Because of the saturation in the sigmoid, a wide range of large magnitude net input values are mapped to the same bin. For (C) and (D), we see that mutual information $I(T; X)$ of tanh and ReLU activations. For tanh, information increases rapidly for small w_1 and then encoungers flat drop for large w_1 since inputs land in one of the two bins corresponding to the saturation regions. On the other hand, ReLU maps half inputs to zero while other half have information that scales with the size of weights.

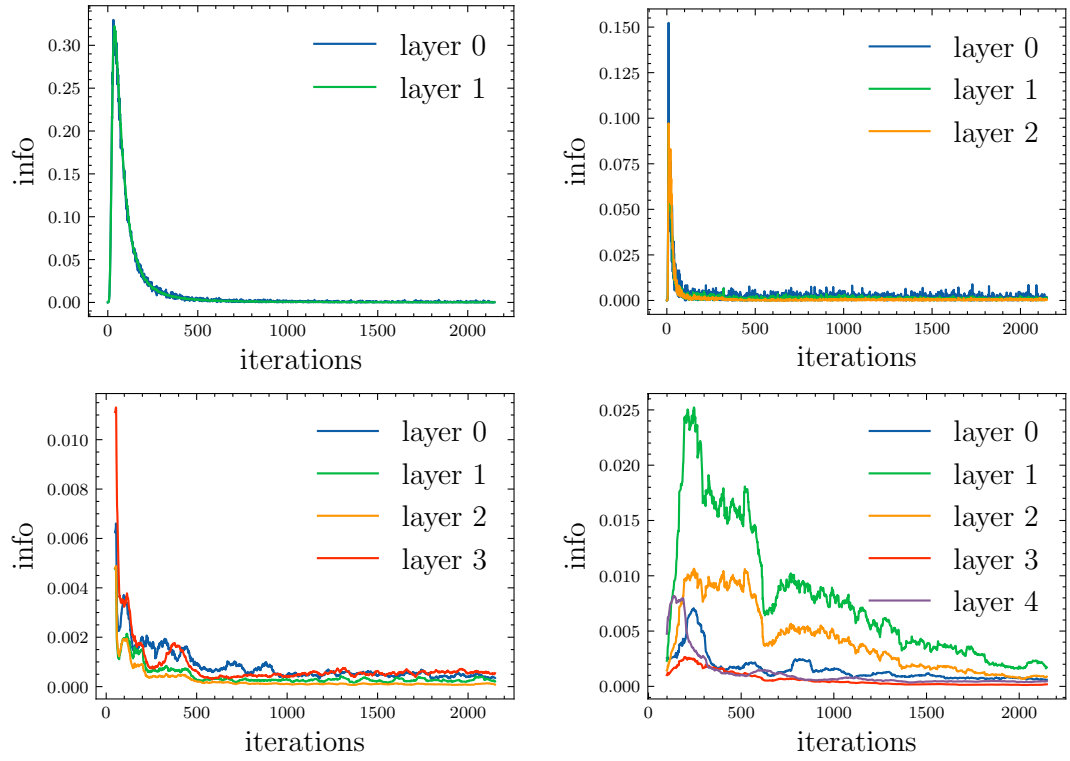


Figure 4.3 Information compression with varying layers (2,3,4,and 5) with tanh non-linearities. We show the 50 and 100 moving average of the 4-layer and 5-layer, respectively, for better display.

4.5.2 Information with Deeper and Wider Architecture

Having identified the phase transition of MLP-Small corresponding to $I(\mathbf{w}; S)$, we focus on its performance under more settings: deeper architecture and different batch size. For the architecture setting, we start from MLP-Large (5 layers) and reduce one layer for each experiment. Results are shown in Fig. 4.3. The first and the second figures show the information trajectory of the reduced two-layer/three-layer version of MLP-Large (784-100-10/784-100-80-10) where a clear two-phase transition happens during the training. One difference between the two is that the three-layer MLP seems to fit faster than the two-layer as there are less iterations executed for it to reach the peak.

Things become more complicated for the more layers cases. The information trajectory fluctuates more frequently hence causes multiple peaks during the SGD training. We thus take the moving average of information to find out what is going on in this process. To illustrate, the third figure demonstrates how the SGD manages to push the model to compress the information. It reaches the peak in very few iterations then experiences multiple tides before it finally becomes stable. The five-layer model is in the similar case. For all those models, we could conclude that they all experience a fitting and com-

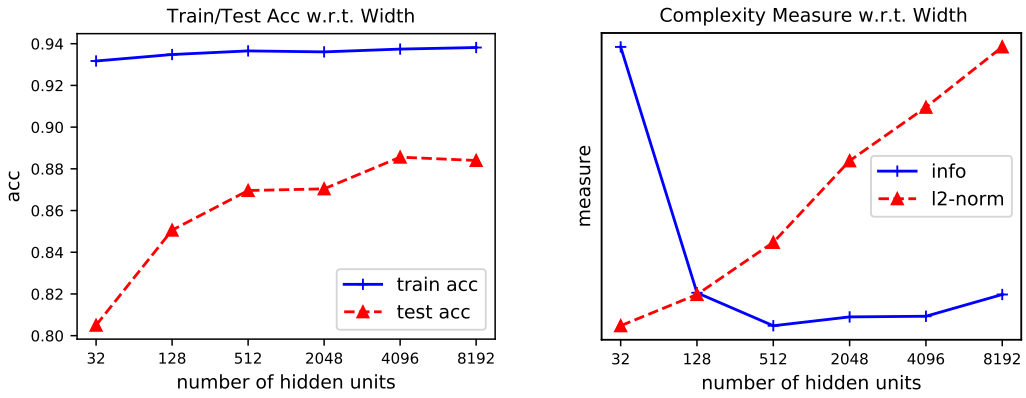


Figure 4.4 **Left**: Train and test accuracy of model reaches with increasing number of hidden units; **Right**: Complexity measure (information in weights and ℓ_2-norm) with increasing unit number.

pression transition during training though there might be several tides after the initial fitting. We believe this subject deserves further examination in future work. Moreover, unlike $I(T_1; X) \geq I(T_2; X)$ because of the Markov chain assumption, please remind that $I(\mathbf{w}_1; S) \ngeq I(\mathbf{w}_2; S)$ here since all weights receive information of S through the back-propagation at the same time.

We also examine how our information measure explains the generalization w.r.t. number of hidden units, a.k.a. the width of DNNs, by Fig. 4.4. We train a two-layer MLP without any regularization on MNIST. The left panel shows the training and testing error for this experiment. While 32 units are enough to (nearly) interpolate the training set, more hidden units still encourage better generalization performance, which illustrates the effect of overparameterization. In this scenario, ℓ_2-norm keeps increasing w.r.t. more units while our information measure decays behaving similar to the test error. We identify that more hidden units do not render much expansion of information complexity.

4.5.3 Random Labels v.s. True Labels

According to PAC-Bayes theorem, the information stored in weights is a promising candidate to explain the generalization capacity of DNNs. Neural networks are often over-parameterized thus can perfectly fit even random labels, obviously without generalizing. For example, MLP-Small has $3072 \times 512 + 512 \times 10 = 1,577,984$ parameters that are much larger than the sample number of CIFAR-10 (50,000). However, the number of parameters is a bad measure of DNN complexity in overparameterization settings[134]. ℓ_2-norm is also often used as a complexity measure to be imposed on regularizing model training in practices while fails to explain generalization (See §4.5.2).

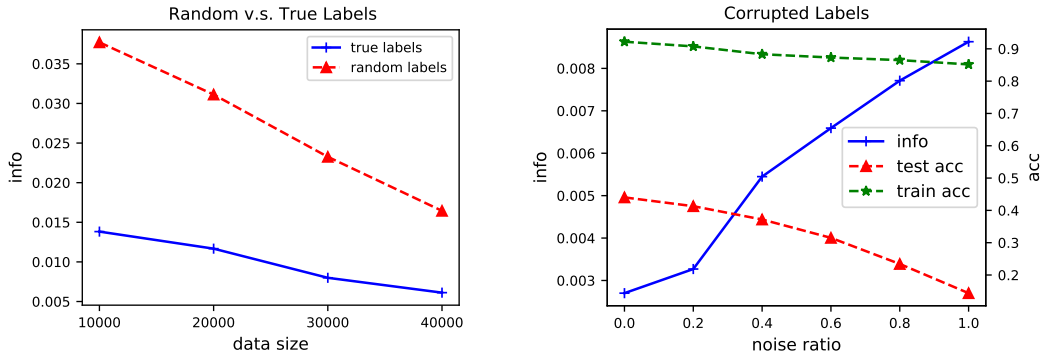


Figure 4.5 **Left**: Information stored in weights with varying size of true-label and random-label data; **Right**: Information, test, and train accuracy when noise ratio in labels changes.

We build a random-label dataset for the purpose of testing if our information measure is capable of discriminating the model learned on the true-label dataset, as shown by the left panel of Fig. 4.5. We train MLP-Small on CIFAR-10. We find the information complexity of model trained on true labels is much lower than the one trained on random labels, which is rightly aligned with their generalization capacity. We also find that regardless of random or true labels, more data generally induces more concise weights on the perspective of information complexity. In this situation, PAC-Bayes bound does not hold any longer due to the non identical distribution of train and test set, which deserves further efforts to solve this challenge.

We further investiage the model trained with different levels of label corruption, as shown by the right panel of Fig .4.5. The model also fits the corrupted data well. The figure depicts the decreasing generalization capacity with the increasing information complexity when the noise ratio increases. These results indicate that our information measure can explain the generalization decay w.r.t. noise degree in labels.

4.5.4 Information Compression w.r.t. Batch Size

On the other hand, we consider how batch size influences the critical point of that transition and the degree of compression at last. Recent effort on bounding $I(\mathbf{w}; S)$ of iterative algorithms (e.g., SGD and SGLD)[84,86] implies that the variance of gradients is a central factor, i.e., the larger the variance, the less the upper bound of $I(\mathbf{w}; S)$. In fact, batch size plays a key role in variance of gradients. For the ultimate case where batch size equals full sample size, the variance of gradient is zero and the model is prone to over-fitting grounded on empirical observations. On the other hand, when batch size equals one, the variance becomes tremendously large while the model usually tends to under-fit.

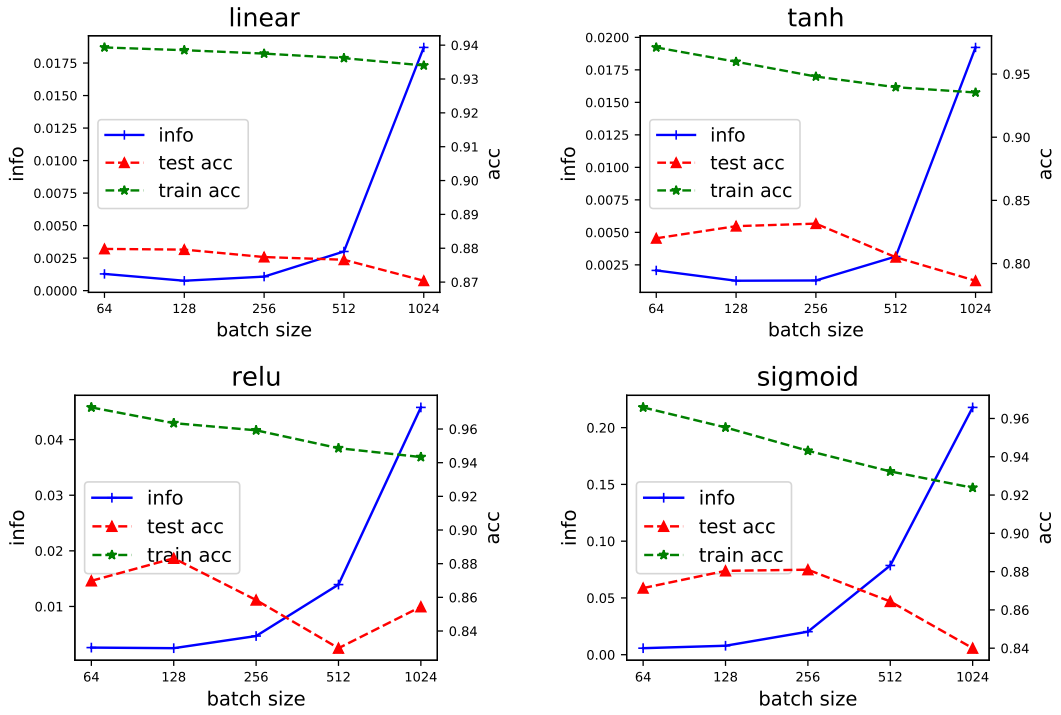


Figure 4.6 Information compression with varying batch size and activation functions. The y-axis on the left indicates the information values and the right indicates the model accuracy. We track the average of minimum information at the end of training and the optimal train/test accuracy the model can reach in the course of whole training process.

In our experiments, we track the average of minimum information with varying batch sizes, results are shown in Fig. 4.6. For each non-linearity, we plot the train and test accuracy when batch size equals 64, 128, 256, 512, and 1024. All models can nearly interpolate, i.e., reach near zero loss on the training set. While batch size is under 256 the compression is sufficient, there is a sharp jump when it equals 512. The test accuracy is generally negatively relevant to information stored in weights seen from all figures. And smaller batch size can enhance information compression thus boosting interpolation on training set and usually leads to better generalization.

4.5.5 Bayesian Inference with Varying Energy Functions

By far, many experiments have been done on To confirm the superiority of our Bayesian inference algorithm for PAC-Bayes IB in §4.4, we compare it with vanilla SGD/SGLD and other two well-known regularization functions: ℓ_2-norm and dropout. We train MLP-Small on MNIST and do grid search to find the optimal hyperparameters: The batch size is picked within $\{32, 64, 128, 256, 512\}$; learning rate is in $\{1e^{-3}, 1e^{-2}, 1e^{-1}\}$; weight decay of ℓ_2-norm is in $\{1e^{-3}, 1e^{-4}, 1e^{-5}, 1e^{-6}\}$; noise scale of SGLD is in $\{1e^{-4}, 1e^{-6}, 1e^{-8}\}$; β of PAC-Bayes IB is in $\{1e^{-1}, 1e^{-2}, 1e^{-3}\}$; and the

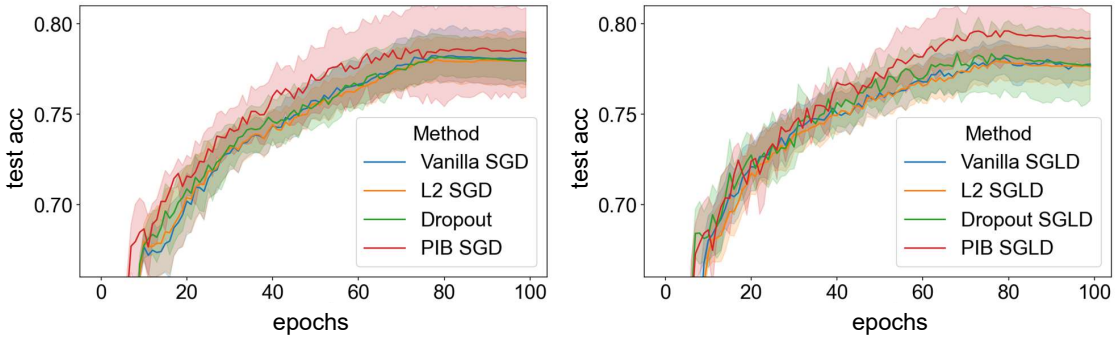


Figure 4.7 10 times repeated experiments of the test accuracy on SGD and SGLD with varying regularization terms.

dropout rate is fixed as 0.5. We repeat the test of the optimal hyperparameter set for 10 times and draw the confidence region as illustrated by Fig. 4.7. For SGD, PIB converges faster than other baselines and keeps the best performance untill convergence. For SGLD, PIB does not demonstrate superiority on fitting speed while still reaches the best performance after convergence. This verifies that SGLD enables sampling from the optimal posterior of PAC-Bayes IB, as derived before in §4.4.

4.5.6 Summary of Experiments

We made the following observations in our experiments:

1. We can clearly identify the fitting-compression phase transition of DNNs during training empirically through our solution of the information measure, i.e., information stored in weights. Unlike the representation-based information measure, our measure applies to varying non-linearities including ReLU, sigmoid, tanh, and linear.
2. We further identify the phase transition applies to deeper and wider architecture while the transition module becomes more complicated, i.e., multiple peaks would appear during the process.
3. Unlike traditional model complexity measure, e.g., ℓ_2-norm of weight matrices, that expands significantly with the model scale, we identify that increasing model scale does not render much expansion of information complexity. This is aligned with the observation that over-parameterized DNNs usually generalize better.
4. Our information measure can explain the generalization decay w.r.t. noisy labels. Besides, more data can contribute to compressing information in weights empirically regardless of the data quality. However, the information complexity of model trained on true labels is much lower than the one trained on random labels.
5. Small batch size can augment information compression thus boosting interpolation on

training set, which often induces good generalization but can also induce over-fitting in some cases.

6. Adopting SGLD based on the energy function derived from PAC-Bayes IB enables good inference to the optimal posterior of DNNs. Similarly, the information regularization also benefits SGD according to our experiments.

4.6 Chapter Summary

In this chapter, we rethought the fitting and compression phases on the aspect of information bottleneck[14] through the lens of information stored in weights $I(\mathbf{w};S)$ of DNNs. Motivated by the caveats of representation-based information measure $I(T;X)$ on explaining learning and generalization of DNNs, we: (1) proposed an new information bottleneck with PAC-Bayesian generalization guarantee, namely PAC-Bayes information bottleneck; (2) gave the solution of intractable oracle prior based information measure $I(\mathbf{w};S)$; (3) designed a Bayesian inference algorithm grounded on SGLD for sampling from the optimal posterior of PIB; and (4) demonstrated that our new information measure covers wide ground of DNNs' behaviour.

We focused the compression and fitting phenomenon of interest, which is illustrated through the lens of information stored in weights by various DNN models. On the contrary, the representation information $I(X;T)$ is influenced by inputs and activation functions, thus only meaningful when the networks is stochastic[23] and saturated non-linearities are implemented[64]. As we gave a rather efficient algorithm for the purpose of estimating information flow in DNNs, we are allowed to testify various architectures of DNNs w.r.t. their width and depth. By numerical tests of DNNs under various settings, we justified $I(\mathbf{w};S)$ is an effective measure of information complexity and generalization capacity. Our solution of $I(\mathbf{w};S)$ by bootstrap also opens the door to the algorithmic design based on PAC-Bayes IB by MCMC, which is theoretically guaranteed to achieve sampling from the optimal posterior efficiently. Crediting to the efficient and easy-to-implement SGLD, we can adapt any existing DNN to a PAC-Bayes IB augmented DNN seamlessly. We aim to further investigate its performance and develop this into practical giant DNNs for production in the near future.

CHAPTER 5 CONCLUSION AND FUTURE WORK

5.1 Conclusion of Current Achievements

This work centers around understanding and utilizing representation learning through the lens of information. Our object of interest is to leverage the information bottleneck principle for the purpose of better representation learning algorithms in application and develop new bottleneck method that covers more aspects of neural network behaviour. In summary, our contributions are three-fold:

In §2, we took a systematic review of the literature of four topics: mutual information estimation & learning, Bayesian inference for deep learning, information bottleneck guided variational auto-encoders, and PAC-Bayes learning theorey. We discussed how information bottleneck for representation learning is located on the intersection of these theories. We also introduced the main framework of Bayesian inference and PAC-Bayes which are the bedrocks for the following two chapters.

In §3, we concentrated on leveraging IB for collaborative filtering (CF) in recommender systems. We proposed **C**ounterfactual **V**ariational **I**nformation Bottleneck (CVIB) in order to debias CF with the emergence of counterfactual events. CVIB separates the task-aware sufficiency into factual/counterfactual terms thus learning balanced representations under missing-not-at-random feedback to improve its generalization ability significantly.

In §4, we delved deeper into IB theory and specifically highlight the pitfalls of representation based IB. We drew the idea from PAC-Bayes learning theory and propose a new weight based IB that is under the umbrella of PAC-Bayes generalization guarantee, namely **P**AC-Bayes **I**nformation **B**ottleneck (PIB). Then, we derived an efficient algorithms for estimating information flow in DNNs and an SGLD-based Bayesian inference algorithm so as to sample from the optimal Gibbs poterior of PAC-Bayes IB. We identified that this new information measure explains broader aspects of learning behaviour of DNNs. Also, this information measure enhances the learned representation capacity when engaged in our algorithm as a regularization.

5.2 Future Works

Although we fixed a series of challenges in applying IB for learning and explaining representation learning, there remain challenges deserving further exploration:

First, information bottleneck can be extended to various applications of deep learning far beyond image classification and collaborative filtering. However, it requires sufficient domain knowledge for adapting IB to a new application, which hinders IB from wider acceptance in production. We are bound to contribute further in studying how to leverage IB principle to other structured data, e.g., graph, and other applications, e.g., disentangled learning.

Second, PAC-Bayes IB was approximated by making Gaussian assumptions at the expense of power in explaining representation learning. Due to the complex structure of DNNs, Guassian assumption is usually too coarse and is like a comprise of allowing tractable solution of KL-divergence. More power distribution assumption of weights, e.g., mixture of Gaussian, is a better fit to the case. Likewise, we are noted that Wasserstein distance based information complexity measure might serve as a promising tool to circumvent limitations of Gaussian assumption and KL-divergence[89-91].

Third, PAC-Bayes learning theory fails to explain when train/test data are non-identically distributed, which is demonstrated by our empirical observations of decreasing information complexity even with more random labels. We are bound to refer to the literature of transfer learning and domain adaption on understanding lower bound of generalization error across domains[135-136], therefore fix the current PAC-Bayes IB.

REFERENCES

[1] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[3] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 248-255.

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[5] Kim Y. Convolutional neural networks for sentence classification[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 1746-1751.

[6] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems[C]// Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. 2016: 7-10.

[7] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[8] Vapnik V. Statistical learning theory[M]. John Wiley, 1998.

[9] Valiant L G. A theory of the learnable[J]. Communications of the ACM, 1984, 27(11): 1134-1142.

[10] Zhang C, Bengio S, Hardt M, et al. Understanding deep learning requires rethinking generalization[J]. arXiv preprint arXiv:1611.03530, 2016.

[11] Cover T M, Thomas J A. Elements of information theory[M]. John Wiley & Sons, 1999.

[12] Tishby N, Pereira F C, Bialek W. The information bottleneck method[J]. arXiv preprint physics/0004057, 2000.

[13] Tishby N, Zaslavsky N. Deep learning and the information bottleneck principle[C]// IEEE Information Theory Workshop (ITW). IEEE, 2015: 1-5.

[14] Shwartz-Ziv R, Tishby N. Opening the black box of deep neural networks via information[J]. arXiv preprint arXiv:1703.00810, 2017.

[15] Alemi A A, Fischer I, Dillon J V, et al. Deep variational information bottleneck[J]. arXiv preprint arXiv:1612.00410, 2016.

[16] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. arXiv preprint arXiv:1310.4546, 2013.

[17] Goodfellow I, Bengio Y, Courville A, et al. Deep learning[M]. MIT press Cambridge, 2016.

[18] Hjelm R D, Fedorov A, Lavoie-Marchildon S, et al. Learning deep representations by mutual information estimation and maximization[C]// International Conference on Learning Representations. 2018.

[19] Bell A J, Sejnowski T J. An information-maximization approach to blind separation and blind deconvolution[J]. Neural computation, 1995, 7(6): 1129-1159.

[20] Belghazi M I, Baratin A, Rajeswar S, et al. MINE: Mutual information neural estimation[J]. arXiv preprint arXiv:1801.04062, 2018.

[21] Velickovic P, Fedus W, Hamilton W L, et al. Deep graph InfoMax[C]// International Conference on Learning Representations. 2019.

[22] Kolchinsky A, Tracey B D, Van Kuyk S. Caveats for information bottleneck in deterministic scenarios[C]// International Conference on Learning Representations. 2018.

[23] Goldfeld Z, Berg E v d, Greenewald K, et al. Estimating information flow in deep neural networks[C]// International Conference on Machine Learning. 2019.

[24] Kingma D P, Welling M. Auto-encoding variational Bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[25] Jordan M I, Ghahramani Z, Jaakkola T S, et al. An introduction to variational methods for graphical models[J]. Machine Learning, 1999, 37(2): 183-233.

[26] Bishop C M. Pattern recognition and machine learning[M]. Springer, 2006.

[27] Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984(6): 721-741.

[28] Hoffman M D, Blei D M, Wang C, et al. Stochastic variational inference.[J]. Journal of Machine Learning Research, 2013, 14(5).

[29] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning[C]// International Conference on Machine Learning. PMLR, 2016: 1050-1059.

[30] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles[C]// Advances in Neural Information Processing Systems. 2017: 6402-6413.

[31] van Amersfoort J, Smith L, Teh Y W, et al. Uncertainty estimation using a single deep deterministic neural network[J]. 2020.

[32] Antorán J, Allingham J U, Hernández-Lobato J M. Depth uncertainty in neural networks[J]. arXiv preprint arXiv:2006.08437, 2020.

[33] Welling M, Teh Y W. Bayesian learning via stochastic gradient Langevin dynamics[C]// International Conference on Machine Learning. Citeseer, 2011: 681-688.

[34] Williams C K, Rasmussen C E. Gaussian processes for machine learning[M]. MIT press Cambridge, MA, 2006.

[35] Lee J, Bahri Y, Novak R, et al. Deep neural networks as Gaussian processes[C]// International Conference on Learning Representations. 2018.

[36] De Matthews A, Hron J, Rowland M, et al. Gaussian process behaviour in wide deep neural networks[C]// International Conference on Learning Representations. 2018.

[37] Hron J, Bahri Y, Novak R, et al. Exact posterior distributions of wide bayesian neural networks[J]. arXiv preprint arXiv:2006.10541, 2020.

[38] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[39] Zhang Y, Liang P, Charikar M. A hitting time analysis of stochastic gradient langevin dynamics[C]// Conference on Learning Theory. PMLR, 2017: 1980-2022.

[40] Teh Y W, Thiery A H, Vollmer S J. Consistency and fluctuations for stochastic gradient langevin dynamics[J]. Journal of Machine Learning Research, 2016, 17.

[41] Raginsky M, Rakhlin A, Telgarsky M. Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis[C]// Conference on Learning Theory. PMLR, 2017: 1674-1703.

[42] Li C, Chen C, Carlson D, et al. Preconditioned stochastic gradient Langevin dynamics for deep neural networks[C]// Proceedings of the AAAI Conference on Artificial Intelligence: volume 30. 2016.

[43] Nado Z, Snoek J, Grosse R, et al. Stochastic gradient Langevin dynamics that exploit neural network structure[J]. 2018.

[44] Dubey A, Reddi S J, Póczos B, et al. Variance reduction in stochastic gradient Langevin dynamics[J]. Advances in Neural Information Processing Systems, 2016, 29: 1154.

[45] Chaudhari P, Choromanska A, Soatto S, et al. Entropy-SGD: Biasing gradient descent into wide valleys[J]. Journal of Statistical Mechanics: Theory and Experiment, 2019, 2019(12): 124018.

[46] Dziugaite G K, Roy D. Entropy-SGD optimizes the prior of a pac-bayes bound: Generalization properties of Entropy-SGD and data-dependent priors[C]// International Conference on Machine Learning. PMLR, 2018: 1377-1386.

[47] Gal Y. Uncertainty in deep learning[D]. University of Cambridge, 2016.

[48] Zhang C, Bütepage J, Kjellström H, et al. Advances in variational inference[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(8): 2008-2026.

[49] Abdar M, Pourpanah F, Hussain S, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges[J]. arXiv preprint arXiv:2011.06225, 2020.

[50] Zaidi A, Estella-Aguerri I, et al. On the information bottleneck problems: Models, connections, applications and information theoretic views[J]. Entropy, 2020, 22(2): 151.

[51] Higgins I, Matthey L, Pal A, et al. Beta-vae: Learning basic visual concepts with a constrained variational framework[C]// International Conference on Learning Representations. 2016.

[52] Burgess C P, Higgins I, Pal A, et al. Understanding disentangling in *beta*-vae[J]. arXiv preprint arXiv:1804.03599, 2018.

[53] Achille A, Soatto S. Information dropout: Learning optimal representations through noisy computation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40 (12): 2897-2905.

[54] Dai B, Zhu C, Guo B, et al. Compressing neural networks using the variational information bottleneck[C]// International Conference on Machine Learning. PMLR, 2018: 1135-1144.

[55] Li X L, Eisner J. Specializing word embeddings (for parsing) by information bottleneck[C]// Conference on Empirical Methods in Natural Language Processing (EMNLP). 2019: 2744-2754.

[56] Achille A, Soatto S. Emergence of invariance and disentanglement in deep representations[J]. The Journal of Machine Learning Research, 2018, 19(1): 1947-1980.

[57] Kolchinsky A, Tracey B D, Wolpert D H. Nonlinear information bottleneck[J]. Entropy, 2019, 21(12): 1181.

[58] Wu T, Ren H, Li P, et al. Graph information bottleneck[J]. arXiv preprint arXiv:2010.12811, 2020.

[59] Pan Z, Niu L, Zhang J, et al. Disentangled information bottleneck[J]. arXiv preprint arXiv:2012.07372, 2020.

[60] Goyal A, Islam R, Strouse D, et al. InfoBot: transfer and exploration via the information bottleneck[C]// International Conference on Learning Representations. 2019.

[61] Wang Q, Boudreau C, Luo Q, et al. Deep multi-view information bottleneck[C]// Proceedings of the SIAM International Conference on Data Mining. SIAM, 2019: 37-45.

[62] Geiger B C, Kubin G. Information bottleneck: Theory and applications in deep learning[M]. Multidisciplinary Digital Publishing Institute, 2020.

[63] Hafez-Kolahi H, Kasaei S. Information bottleneck and its applications in deep learning[J]. Information Systems & Telecommunication, 2019, 3(4): 119.

[64] Saxe A M, Bansal Y, Dapello J, et al. On the information bottleneck theory of deep learning[J]. Journal of Statistical Mechanics: Theory and Experiment, 2019, 2019(12): 124020.

[65] Tschannen M, Djolonga J, Rubenstein P K, et al. On mutual information maximization for representation learning[C]// International Conference on Learning Representations. 2020.

[66] Kirsch A, Lyle C, Gal Y. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning[J]. arXiv preprint arXiv:2003.12537, 2020.

[67] Poole B, Ozair S, Van Den Oord A, et al. On variational bounds of mutual information[C]// International Conference on Machine Learning. PMLR, 2019: 5171-5180.

[68] Noshad M, Zeng Y, Hero A O. Scalable mutual information estimation using dependence graphs[C]// IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 2962-2966.

[69] McAllester D, Stratos K. Formal limitations on the measurement of mutual information[C]// International Conference on Artificial Intelligence and Statistics. PMLR, 2020: 875-884.

[70] Cisse M, Bojanowski P, Grave E, et al. Parseval networks: Improving robustness to adversarial examples[C]// International Conference on Machine Learning. PMLR, 2017: 854-863.

[71] Guedj B. A primer on PAC-Bayesian learning[J]. arXiv preprint arXiv:1901.05353, 2019.

[72] McAllester D A. Some PAC-Bayesian theorems[J]. Machine Learning, 1999, 37(3): 355-363.

[73] Liang P. CS229T/STAT231: Statistical learning theory (winter 2016)[M]. 2016.

[74] Shawe-Taylor J, Williamson R C. A PAC analysis of a bayesian estimator[C]// Proceedings of the Annual Conference on Computational Learning Theory. 1997: 2-9.

[75] Shawe-Taylor J, Bartlett P L, Williamson R C, et al. Structural risk minimization over data-dependent hierarchies[J]. IEEE Transactions on Information Theory, 1998, 44(5): 1926-1940.

[76] McAllester D. A PAC-Bayesian tutorial with a dropout bound[J]. arXiv preprint arXiv:1307.2118, 2013.

[77] Belkin M, Hsu D, Ma S, et al. Reconciling modern machine-learning practice and the classical bias–variance trade-off[J]. Proceedings of the National Academy of Sciences, 2019, 116(32): 15849-15854.

[78] Dziugaite G K, Roy D M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data[J]. arXiv preprint arXiv:1703.11008, 2017.

[79] Neyshabur B, Bhojanapalli S, McAllester D, et al. Exploring generalization in deep learning[J]. arXiv preprint arXiv:1706.08947, 2017.

[80] Achille A, Paolini G, Soatto S. Where is the information in a deep neural network?[J]. arXiv preprint arXiv:1905.12213, 2019.

[81] Rivasplata O, Parrado-Hernández E, Shawe-Taylor J, et al. PAC-Bayes bounds for stable algorithms with instance-dependent priors[C]// Advances in Neural Information Processing Systems. 2018: 9234-9244.

[82] Russo D, Zou J. How much does your data exploration overfit? controlling bias via information usage[J]. IEEE Transactions on Information Theory, 2019, 66(1): 302-323.

[83] Xu A, Raginsky M. Information-theoretic analysis of generalization capability of learning algorithms[C]// Advances in Neural Information Processing Systems. 2017: 2521-2530.

[84] Mou W, Wang L, Zhai X, et al. Generalization bounds of SGLD for non-convex learning: Two theoretical viewpoints[C]// Conference on Learning Theory. PMLR, 2018: 605-638.

[85] Negrea J, Haghifam M, Dziugaite G K, et al. Information-theoretic generalization bounds for SGLD via data-dependent estimates[J]. arXiv preprint arXiv:1911.02151, 2019.

[86] Pensia A, Jog V, Loh P L. Generalization error bounds for noisy, iterative algorithms[C]// IEEE International Symposium on Information Theory (ISIT). IEEE, 2018: 546-550.

[87] Zhang J, Liu T, Tao D. An information-theoretic view for deep learning[J]. arXiv preprint arXiv:1804.09060, 2018.

[88] Asadi A R, Abbe E, Verdú S. Chaining mutual information and tightening generalization bounds[C]// Advances in Neural Information Processing Systems. 2018: 7245-7254.

[89] Raginsky M, Rakhlin A, Tsao M, et al. Information-theoretic analysis of stability and bias of learning algorithms[C]// IEEE Information Theory Workshop (ITW). IEEE, 2016: 26-30.

[90] Lopez A T, Jog V. Generalization error bounds using Wasserstein distances[C]// IEEE Information Theory Workshop (ITW). IEEE, 2018: 1-5.

[91] Wang H, Diaz M, Santos Filho J C S, et al. An information-theoretic view of generalization via Wasserstein distance[C]// IEEE International Symposium on Information Theory (ISIT). IEEE, 2019: 577-581.

[92] Issa I, Gastpar M. Computable bounds on the exploration bias[C]// IEEE International Symposium on Information Theory (ISIT). IEEE, 2018: 576-580.

[93] Issa I, Esposito A R, Gastpar M. Strengthened information-theoretic bounds on the generalization error[C]// IEEE International Symposium on Information Theory (ISIT). IEEE, 2019: 582-586.

[94] Alabdulmohsin I M. Algorithmic stability and uniform generalization[J]. Advances in Neural Information Processing Systems, 2015, 28: 19-27.

[95] Bu Y, Zou S, Veeravalli V V. Tightening mutual information-based bounds on generalization error[J]. IEEE Journal on Selected Areas in Information Theory, 2020, 1(1): 121-130.

[96] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence, 2009, 2009.

[97] Koren Y, Bell R. Advances in collaborative filtering[J]. Recommender Systems Handbook, 2015: 77-118.

[98] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]// International Conference on World Wide Web. 2017: 173-182.

[99] Wang X, He X, Wang M, et al. Neural graph collaborative filtering[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 165-174.

[100] Wang Z, Wen R, Chen X, et al. Online disease self-diagnosis with inductive heterogeneous graph convolutional networks[C]// The Web Conference. 2021.

[101] Wang Z, Chen X, Wen R, et al. Information theoretic counterfactual learning from missing-not-at-random feedback[C]// Advances in Neural Information Processing Systems. 2020.

[102] Marlin B M, Zemel R S. Collaborative prediction and ranking with non-random missing data[C]// ACM Conference on Recommender Systems. ACM, 2009: 5-12.

[103] Steck H. Evaluation of recommendations: rating-prediction and ranking[C]// ACM Conference on Recommender Systems. ACM, 2013: 213-220.

[104] Wang Z, Chen X, Wen R, et al. On the fairness of randomized trials for recommendation with heterogeneous demographics and beyond[J]. arXiv preprint arXiv:2001.09328, 2020.

[105] Schnabel T, Swaminathan A, Singh A, et al. Recommendations as treatments: Debiasing learning and evaluation[J]. arXiv preprint arXiv:1602.05352, 2016.

[106] Saito Y. Asymmetric tri-training for debiasing missing-not-at-random explicit feedback[C]// Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 309-318.

[107] Guo H, Tang R, Ye Y, et al. DeepFM: A factorization-machine based neural network for CTR prediction[J]. arXiv preprint arXiv:1703.04247, 2017.

[108] Lian J, Zhou X, Zhang F, et al. xDeepFm: Combining explicit and implicit feature interactions for recommender systems[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2018: 1754-1763.

[109] Chalmers T C, Smith Jr. H, Blackburn B, et al. A method for assessing the quality of a randomized control trial[J]. Controlled Clinical Trials, 1981, 2(1): 31-49.

[110] Lakkaraju H, Kleinberg J, Leskovec J, et al. The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables[C/OL]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 275–284. DOI: 10.1145/3097983.3098066.

[111] Rosenfeld N, Mansour Y, Yom-Tov E. Predicting counterfactuals from large historical data and small randomized trials[C]// International Conference on World Wide Web Companion. 2017: 602-609.

[112] Bonner S, Vasile F. Causal embeddings for recommendation[C]// ACM Conference on Recommender Systems. 2018: 104-112.

[113] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009(8): 30-37.

[114] Hu J, Ji R, Zhang S, et al. Information competing process for learning diversified representations[C]// Advances in Neural Information Processing Systems. 2019: 2175-2186.

[115] Jaynes E T. Information theory and statistical mechanics[J]. Physical Review, 1957, 106(4): 620.

[116] Berger A L, Pietra V J D, Pietra S A D. A maximum entropy approach to natural language processing[J]. Computational Linguistics, 1996, 22(1): 39-71.

[117] Pereyra G, Tucker G, Chorowski J, et al. Regularizing neural networks by penalizing confident output distributions[J]. arXiv preprint arXiv:1701.06548, 2017.

[118] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2008: 426-434.

[119] Swaminathan A, Joachims T. The self-normalized estimator for counterfactual learning[C]// Advances In Neural Information Processing Systems. 2015: 3231-3239.

[120] Jiang N, Li L. Doubly robust off-policy value evaluation for reinforcement learning[J]. arXiv preprint arXiv:1511.03722, 2015.

[121] Wang X, Zhang R, Sun Y, et al. Doubly robust joint learning for recommendation on data missing not at random[C]// International Conference on Machine Learning: volume 97. PMLR, 2019: 6638-6647.

[122] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[C]// Advances in Neural Information Processing Systems. 2019: 8024-8035.

[123] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[124] Cook R D, Weisberg S. Residuals and influence in regression[M]. New York: Chapman and Hall, 1982.

[125] Koh P W, Liang P. Understanding black-box predictions via influence functions[C]// International Conference on Machine Learning. PMLR, 2017: 1885-1894.

[126] Wang Z, Zhu H, Dong Z, et al. Less is better: Unweighted data subsampling via influence function[C]// AAAI Conference on Artificial Intelligence: volume 34. 2020: 6340-6347.

[127] Efron B. Bootstrap methods: another look at the jackknife[M]// Breakthroughs in Statistics. Springer, 1992: 569-593.

[128] Chamandy N, Muralidharan O, Najmi A, et al. Estimating uncertainty for massive data streams[J]. 2012.

[129] Liang T, Poggio T, Rakhlin A, et al. Fisher-Rao metric, geometry, and complexity of neural networks[C]// International Conference on Artificial Intelligence and Statistics. PMLR, 2019: 888-896.

[130] Kittel C. Elementary statistical physics[M]. Courier Corporation, 2004.

[131] Borkar V S, Mitter S K. A strong approximation theorem for stochastic recursive algorithms[J]. Journal of Optimization Theory and Applications, 1999, 100(3): 499-513.

[132] Chiang T S, Hwang C R, Sheu S J. Diffusion for global optimization in r^n[J]. SIAM Journal on Control and Optimization, 1987, 25(3): 737-753.

[133] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images[R]. Citeseer, 2009.

[134] Neyshabur B, Tomioka R, Srebro N. In search of the eal inductive bias: On the role of implicit regularization in deep learning.[C]// International Conference on Learning Representations Workshop. 2015.

[135] Ben-David S, Blitzer J, Crammer K, et al. A theory of learning from different domains[J]. Machine Learning, 2010, 79(1): 151-175.

[136] Blitzer J, Crammer K, Kulesza A, et al. Learning bounds for domain adaptation[C]// International Conference on Neural Information Processing Systems. 2007: 129-136.

[137] Martens J. New insights and perspectives on the natural gradient method[J]. Journal of Machine Learning Research, 2020, 21: 1-76.

APPENDIX A PROOF IN CHAPTER 3

A.1 Proof of Proposition 3.1

Proposition A.1 (Minimal Representation Insensitive to Policy Bias): With the Markov chain assumption defined by Eq. (3.4), for any hidden embedding T_l, we can derive the upper bound of the $I(T_l; O)$

$$I(T_l; O) \leq I(T_l; X) - I(X; Y) \leq I(T_1; X) - I(X; Y) \leq I(T_l; X), \tag{A.1}$$

where the last term $I(X; Y)$ is a constant with respect to the training process.

Proof: From the Data Processing Inequality (DPI)[11], in this Markov chain, we can obtain

$$I(T; X) \geq I(T; Y, O) = I(T; O) + I(T; Y|O). \tag{A.2}$$

For the second term $I(T; Y|O)$, suppose Y and O are independent, we can further factorize it and derive

$$\begin{aligned} I(T; Y|O) &= H(Y|O) - H(Y|Y, O) && \text{(A.3)} \\ &= H(Y) - H(Y|T, O) && \text{(A.4)} \\ &\geq H(Y) - H(Y|T) && \text{(A.5)} \\ &= I(T; Y). && \text{(A.6)} \end{aligned}$$

As we assume that T is sufficient, we have $I(T; Y) = I(X; Y)$. Plugging above result back into Eq.(A.2) yields

$$\begin{aligned} I(T; X) &\geq I(T; O) + I(T; Y|O) && \text{(A.7)} \\ &\geq I(T; O) + I(T; Y) && \text{(A.8)} \\ &= I(T; O) + I(X; Y), && \text{(A.9)} \end{aligned}$$

which indicates that $I(T; X) - I(X; Y)$ bounds $I(T; O)$. And for any hidden embeddings T_l, according to DPI, we have

$$I(T_l; T_1) \leq I(T_1; X) \quad \forall l \in \{2, \ldots, L\}, \tag{A.10}$$

which yields the final result. ∎

APPENDIX B PROOF IN CHAPTER 4

B.1 Proof of Lemma 4.3

Before the proof of Lemma 4.3, we need to introduce a lemma by Martens[137] as:

Lemma B.1 (Approximation of Hessian matrix in DNNs[137]): The Hessian matrix of DNNs on a local minima $\hat{\boldsymbol{\theta}}$ can be decomposed based on Fisher information matrix as

$$\mathbf{H}_{\hat{\theta}} = \mathbf{F}_{\hat{\theta}} + \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}[\nabla_{\hat{\mathbf{y}}}\ell_i(\hat{\boldsymbol{\theta}})]_c\mathbf{H}_{[f]_c}, \tag{B.1}$$

where C is the total number of classes, $\hat{\mathbf{y}}$ is the output (or prediction) of the network given input bx, and $\mathbf{H}_{[f]_c}$ is the Hessian of the c-th component of $\hat{\mathbf{y}}$. Specifically, for a well-trained DNN, we could have $\nabla_{\hat{\mathbf{y}}}\ell_i(\hat{\boldsymbol{\theta}}) \simeq 0$ thus $\mathbf{H}_{\hat{\theta}} \simeq \mathbf{F}_{\hat{\theta}}$.

Lemma B.2 (Approximation of Oracle Prior Covariance): Given the definition of influence functions (Lemma 4.1) and Poisson bootstrapping (Lemma 4.2), the covariance matrix of the oracle prior can be approximated by

$$\begin{aligned}\boldsymbol{\Sigma}_0 = \mathbb{E}_{p(S)}\left[(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)(\boldsymbol{\theta}_S - \boldsymbol{\theta}_0)^\top\right] &\simeq \frac{1}{K}\sum_{k=1}^{K}\left(\hat{\boldsymbol{\theta}}_{\xi^k} - \hat{\boldsymbol{\theta}}\right)\left(\hat{\boldsymbol{\theta}}_{\xi^k} - \hat{\boldsymbol{\theta}}\right)^\top \\ &\simeq \frac{1}{n}\mathbf{H}_{\hat{\theta}}^{-1}\mathbf{F}_{\hat{\theta}}\mathbf{H}_{\hat{\theta}}^{-1} \simeq \frac{1}{n}\mathbf{F}_{\hat{\theta}}^{-1},\end{aligned} \tag{B.2}$$

where we omit the subscript S of $\hat{\boldsymbol{\theta}}_S$ and $\hat{\boldsymbol{\theta}}_{S,\xi}$ for notation conciseness, and $\boldsymbol{\xi}^k$ is the bootstrap resampling weight in the k-th experiment.

Proof: Recall that in the k-th bootstrap resampling process, the loss function is reweighted by $\boldsymbol{\xi}_k = (\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,n})^\top$. Also, we have a influence matrix $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n)^\top \in \mathbb{R}^{n\times D}$. The original risk minimizer on the full dataset S is

$$\hat{\boldsymbol{\theta}}_S \triangleq \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\ell_i(\boldsymbol{\theta}), \tag{B.3}$$

and the reweighted empirical risk minimizer (after bootstrapping) is defined by

$$\hat{\boldsymbol{\theta}}_{S,\xi} \triangleq \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n}\xi_i\ell_i(\boldsymbol{\theta}), \tag{B.4}$$

where we omit the subscript k for the sake of conciseness.Given the definition of influence function from Lemma 4.1, the difference between the two risk minizers above, $\hat{\boldsymbol{\theta}}_S$ and

$\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}}$, can be written as

$$\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}} - \hat{\boldsymbol{\theta}}_S \simeq \frac{1}{n}\sum_{i=1}(\xi_i - 1)\boldsymbol{\psi}_i = \frac{1}{n}\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1}). \tag{B.5}$$

As a result, the oracle prior can be transformed as

$$\begin{aligned}
\boldsymbol{\Sigma}_0 &\simeq \mathbb{E}_{p(S)}\left[(\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}} - \hat{\boldsymbol{\theta}}_S)(\hat{\boldsymbol{\theta}}_{S,\boldsymbol{\xi}} - \hat{\boldsymbol{\theta}}_S)^\top\right] && \text{(B.6)}\\
&\simeq \mathbb{E}_{p(S)}\left[\left(\frac{1}{n}\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1})\right)\left(\frac{1}{n}\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1})\right)^\top\right] && \text{(B.7)}\\
&= \frac{1}{n^2}\mathbb{E}_{p(S)}\left[\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1})(\boldsymbol{\xi} - \mathbf{1})^\top\boldsymbol{\Psi}\right]. && \text{(B.8)}
\end{aligned}$$

Furthermore, based on the definition of influence function, we know $\boldsymbol{\Psi}^\top\mathbf{1} = \mathbf{1}^\top\boldsymbol{\Psi} = 0$. The term in Eq. (B.8) can be further writen to

$$\boldsymbol{\Sigma}_0 \simeq \frac{1}{n^2}\mathbb{E}_{p(S)}\left[\boldsymbol{\Psi}^\top(\boldsymbol{\xi} - \mathbf{1})(\boldsymbol{\xi} - \mathbf{1})^\top\boldsymbol{\Psi}\right] = \frac{1}{n^2}\boldsymbol{\Psi}^\top\mathbb{E}_{p(S)}[\boldsymbol{\xi}\boldsymbol{\xi}^\top]\boldsymbol{\Psi}. \tag{B.9}$$

From Lemma 4.2 we know $\mathbb{E}[\xi_i] = 1$ and $\mathrm{Var}[\xi_i] = 1$ when $n \geq 100$. We also know that

$$\mathbb{E}_{p(S)}[\xi_i\xi_j] = \begin{cases} \mathbb{E}_{p(S)}[\xi_i]\mathbb{E}_{p(S)}[\xi_j] = 1, & i \neq j,\\ \mathbb{E}_{p(S)}[\xi_i^2] = \mathrm{Var}[\xi_i] + \mathbb{E}^2[\xi_i] = 2, & i = j. \end{cases}$$

This gives rise to the final solution that

$$\begin{aligned}
\boldsymbol{\Psi}^\top\mathbb{E}_{p(S)}[\boldsymbol{\xi}\boldsymbol{\xi}^\top]\boldsymbol{\Psi} &= \boldsymbol{\Psi}^\top\left(\mathbf{1}\mathbf{1}^\top + \mathbf{I}_n\right)\boldsymbol{\Psi} && \text{(B.10)}\\
&= \sum_{i=1}^{n}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\top && \text{(B.11)}\\
&= \sum_{i=1}^{n}\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\nabla_{\boldsymbol{\theta}}\ell_i(\hat{\boldsymbol{\theta}})\ell_i^\top(\hat{\boldsymbol{\theta}})\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1} && \text{(B.12)}\\
&= n\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}}\ell_i(\hat{\boldsymbol{\theta}})\nabla_{\boldsymbol{\theta}}\ell_i^\top(\hat{\boldsymbol{\theta}})\right]\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1} && \text{(B.13)}\\
&= n\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{F}_{\hat{\boldsymbol{\theta}}}\mathbf{H}_{\hat{\boldsymbol{\theta}}}^{-1} && \text{(B.14)}\\
&\simeq n\mathbf{F}_{\hat{\boldsymbol{\theta}}}^{-1}. && \text{(B.15)}
\end{aligned}$$

$\mathbf{I}_n$ in Eq. (B.10) is an identity matrix with size $n \times n$; Eq. (B.11) is true by re-applying the property of influence functions that $\boldsymbol{\Psi}^\top\mathbf{1} = \mathbf{1}^\top\boldsymbol{\Psi} = 0$; and Eq. (B.15) is achieved by the result from Lemma B.1. Concluding all the above results, we know the oracle prior covariance can be approximated by

$$\boldsymbol{\Sigma}_0 \simeq \frac{1}{n^2}\left(n\mathbf{F}_{\hat{\boldsymbol{\theta}}}^{-1}\right) = \frac{1}{n}\mathbf{F}_{\hat{\boldsymbol{\theta}}}^{-1}. \tag{B.16}$$

∎

B.2 Proof of Lemma 4.4

Lemma B.3 (Optimal Posterior for PAC-Bayes Information Bottleneck): Given an observed dataset S^*, the optimal posterior $p(\mathbf{w}|S^*)$ of PAC-Bayes IB in Eq. (4.5) should satisfy the following form that

$$p(\mathbf{w}|S^*) = \frac{1}{Z(S)} p(\mathbf{w}) \exp\left\{-\frac{1}{\beta}\hat{L}_{S^*}(\mathbf{w})\right\} = \frac{1}{Z(S)} \exp\left\{-\frac{1}{\beta}\left(U_{S^*}(\mathbf{w})\right)\right\}, \quad \text{(B.17)}$$

where $U_{S^*}(\mathbf{w})$ is the energy function defined by

$$U_{S^*}(\mathbf{w}) = \hat{L}_{S^*}(\mathbf{w}) - \beta \log p(\mathbf{w}), \quad \text{(B.18)}$$

and $Z(S)$ is the normalizing constant.

Proof: Recap the PAC-Bayes information bottleneck in Eq. (4.5) is

$$\min_{p(\mathbf{w}|S)} \mathcal{L}_{\text{PIB}} = L_S(\mathbf{w}) + \beta I(\mathbf{w}; S). \quad \text{(B.19)}$$

Given an observed dataset S^*, our object of interest is to find the optimal posterior $p(\mathbf{w}|S^*)$ that minimizes the $\mathcal{L}_{\text{PIB}}$. Consider a constraint of posterior distribution that

$$\int p(\mathbf{w}|S) d\mathbf{w} = 1, \ \forall S \sim p(X, Y)^{\otimes n}, \quad \text{(B.20)}$$

we can formulate the problem by

$$\begin{aligned} \min_{p(\mathbf{w}|S)} \mathcal{L}_{\text{PIB}} &= L_S(\mathbf{w}) + \beta I(\mathbf{w}; S), \\ \text{s.t.} &\int p(\mathbf{w}|S) d\mathbf{w} = 1. \end{aligned} \quad \text{(B.21)}$$

A Lagragian can hence be built to relax the above optimization problem by

$$\begin{aligned} &\min_{p(\mathbf{w}|S)} \widetilde{\mathcal{L}}_{\text{PIB}} = L_S(\mathbf{w}) + \beta I(\mathbf{w}; S) + \int \alpha_S \int \left(p(\mathbf{w}|S) - 1\right) d\mathbf{w} dS \\ &= \int p(\mathbf{w}|S) \left[\hat{L}_S(\mathbf{w})\right] d\mathbf{w} + \beta \int p(\mathbf{w}, S) \left[\log p(\mathbf{w}|S) - \log p(\mathbf{w})\right] d\mathbf{w} dS \\ &+ \int \alpha_S \int \left(p(\mathbf{w}|S) - 1\right) d\mathbf{w} dS, \end{aligned} \quad \text{(B.22)}$$

with $\square = \left\{\alpha_S | \forall S \sim p(X, Y)^{\otimes n}\right\}$ corresponding to Lagrange multipliers; we denote the empirical risk by $\hat{L}_S(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\mathbf{w})$.

Differentiating $\widetilde{\mathcal{L}}_{\text{PIB}}$ w.r.t. $p(\mathbf{w}|S^*)$ results in

$$\nabla_{p(\mathbf{w}|S^*)} \widetilde{\mathcal{L}}_{\text{PIB}} = \hat{L}_{S^*}(\mathbf{w}) + \beta \log p(\mathbf{w}|S^*) - \beta \log p(\mathbf{w}) + \beta + \alpha_{S^*}. \quad \text{(B.23)}$$

Setting $\nabla_{p(\mathbf{w}|S^*)}\widetilde{\mathcal{L}}_{\text{PIB}} = 0$ and solving for $p(\mathbf{w}|S^*)$ yields

$$\begin{aligned} \log p(\mathbf{w}|S^*) &= -\frac{1}{\beta}\hat{L}_{S^*}(\mathbf{w}) + \log p(\mathbf{w}) - 1 - \frac{\alpha_{S^*}}{\beta} \\ p(\mathbf{w}|S^*) &= p(\mathbf{w}) \exp\left\{-\frac{1}{\beta}\hat{L}_{S^*}(\mathbf{w})\right\} \exp\left\{-1 - \frac{\alpha_{S^*}}{\beta}\right\}. \end{aligned} \tag{B.24}$$

The second exponential term $\exp\left\{-1 - \frac{\alpha_{S^*}}{\beta}\right\}$ is the partition function that normalizes the posterior distribution. We hence obtain the optimal posterior solution as

$$\begin{aligned} p(\mathbf{w}|S^*) &= \frac{1}{Z(S)} p(\mathbf{w}) \exp\left\{-\frac{1}{\beta}\hat{L}_{S^*}(\mathbf{w})\right\} \\ &= \frac{1}{Z(S)} \exp\left\{-\frac{1}{\beta}\left[\hat{L}_{S^*}(\mathbf{w}) - \beta \log p(\mathbf{w})\right]\right\}. \end{aligned} \tag{B.25}$$

∎

ACKNOWLEDGEMENTS

I would like to highlight and sincerely thank the supervision of Prof. Shao-Lun Huang and Prof. Khalid M. Mosalam in my three years of graduate study. Without them, I would not have been able to get a foot in the door of research, especially after I transferred my major from Civil Engineering in undergraduate to Data Science as my graduate focus. I would also thank the support of Prof. Mosalam for my application to PhD programs.

I wish to express my genuine gratitude to Prof. Ercan E. Kuruoglu, who comes from ISTI-CNR, Italy and is Visiting Professor at TBSI, for his guidance to Bayesian theory, then to my first research project towards information principled deep learning, as well as his powerful support to my application of PhD programs.

I am deeply thankful to Dr. Yefeng Zheng and Dr. Xi Chen as my supervisors at Jarvis Lab, Tencent. During my internship over one and a half year, they offered me significant guidance to my research on AI healthcare and coordinating me to other colleagues at Tencent. I also thank Dr. Zheng's support to my application of PhD programs.

I would also like to thank Xinyi Zhao for the companion with me to enjoy every precious moment last three years and to deal with any challenges ahead in the future. I am grateful to work with my colleagues Dr. Hong Zhu, Dr. Zhenhua Dong, Yuyang Zhang, Rui Wen, and Yifan Yang who all did me a great favor to my life and reasearch. I am also happy to be with all members of Lab 2C at TBSI who make there a so vibrant place.

Last but not least, I genuinely thank my parents for their understanding on my choice of going abroad for PhD study and their unreserved support by my side. I would have not took the courage to pursue my dream without their encouragement and belief.

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：王子丰　日　期：2021/5/28

RESUME

Resume

Zifeng Wang was born November 10, 1996 in Yueyang city, Hunan province.

In September 2014, he was admitted to the College of Civil Engineering at Tongji University. In July 2018, he graduated with a Bachelor of Science degree.

In September 2018, he was admitted to Tsinghua-Berkeley Shenzhen Institute (TBSI) at Tsinghua University for a Master of Science degree in Data Science and Information Technology.

Research Achievements

Published Papers:

[1] **Wang Z**, Chen X, Wen R, et al. Information theoretic counterfactual learning from missing-not-at-random feedback[C]. Advances in Neural Information Processing Systems (**NeurIPS**). 2020, 33: 1854-1864. (EI, CCF-A)

[2] **Wang Z**, Zhu H, Dong Z, et al. Less is better: Unweighted data subsampling via influence function[C]. Proceedings of the AAAI Conference on Artificial Intelligence (**AAAI**). 2020, 34(04): 6340-6347. (EI, CCF-A)

[3] **Wang Z**, Wen R, Chen X, et al. Online disease self-diagnosis with inductive heterogeneous graph convolutional networks[C]. Proceedings of the Web Conference (**WWW**). 2021. (EI, CCF-A)

[4] **Wang Z**, Yang Y, Wen R, et al. Lifelong learning based disease diagnosis on clinical notes[C]. Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (**PAKDD**). 2021. (EI, CCF-C, **Best Student Paper, 2 out of 768**)

[5] **Wang Z**, Zhang Y, Mosalam K.M., et al. Deep semantic segmentation for visual understanding on construction sites[J]. Computer-Aided Civil And Infrastructure Engineering (**CACAIE**). 2021. (SCI, IF 8.552, Chinese Academy of Sciences ranking Q1)

[6] **Wang Z**, Li S. Data-driven Risk Assessment on Urban Pipeline Network based on a Cluster Model [J]. Reliability Engineering & System Safety (**RESS**). 2019. (SCI, IF 5.040, Chinese Academy of Sciences ranking Q1)

Papers Under Review:

[7] **Wang Z**, Huang S-L, Kuruoglu E.E., et al. PAC-Bayes information bottleneck[C]. Submitted to Advances in Neural Information Processing Systems (**NeurIPS**), under review. (May, 2021, EI, CCF-A)

[8] **Wang Z**, Wen R, Chen X, et al. Finding influential instances for distantly supervised relation extraction[C]. Submitted to Conference on Empirical Methods in Natural Language Processing (**EMNLP**), under review. (May 2021, EI, CCF-B)

COMMENTS FROM THESIS SUPERVISOR GROUP

This thesis is focused on extending the application of information bottleneck theory for representation learning and improving the Information Bottleneck (IB) theory for the purpose of explaining deep learning. The author made a systematic review of the relevant literature and had solid contributions to both application and theory within the scope of information-theoretic representation learning. Besides, it is noted that he published six papers in top-tier journals and conferences during his MS study. Grounded on his achievements and the thesis, I agree that he is ready to submit the thesis and apply for an MS degree by the defense.

RESOLUTION OF THESIS DEFENSE COMMITTEE

The thesis investigates the information bottleneck (IB) principle for representation learning from two aspects: application and theory. On the application part the IB is leveraged for collaborative filtering (CF) in recommender systems. On the theory part, the PAC-Bayes IB is proposed which has the generalization guarantee.

The main contributions of this thesis include：1. Counterfactual Variational Information Bottleneck (CVIB) is proposed in order to debias CF with the emergence of counterfactual events. 2. A new weight based IB named PAC-Bayes is proposed for estimating information flow in DNNs.

The topic of this thesis is on the leading discipline frontier and is significant in both theory and application. The work of this thesis shows that the author possesses solid knowledge of the discipline, have strong academic research ability. Thesis is well formatted and writing is clear and logical. The contributions of the thesis are outstanding and innovative. The presentation in defense is informative and well organized, question answering is insightful.

The defense committee voted, (5 agree out of 5 vote / unanimously) agreed to pass the thesis defense, and proposed to confer Zifeng Wang a Master's degree in Data Science and Information Technology.

The committee also recommended this thesis to be submitted to the Tsinghua University Outstanding Master Thesis Competition.