Rethinking InfoNCE: How Many Negative Samples Do You Need?

Chuhan Wu[1] , **Fangzhao Wu**[2]* and **Yongfeng Huang**[1]
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China
{wuchuhan15, wufangzhao}@gmail.com, yfhuang@tsinghua.edu.cn

Abstract

InfoNCE is a widely used contrastive training loss. It aims to estimate the mutual information between a pair of variables by discriminating between each positive pair and its associated K negative pairs. It is proved that when the sample labels are clean, the lower bound of mutual information estimation is tighter when more negative samples are incorporated, which usually yields better model performance. However, in practice the labels often contain noise, and incorporating too many noisy negative samples into model training may be suboptimal. In this paper, we study how many negative samples are optimal for InfoNCE in different scenarios via a semi-quantitative theoretical framework. More specifically, we first propose a probabilistic model to analyze the influence of the negative sampling ratio K on training sample informativeness. Then, we design a training effectiveness function to measure the overall influence of training samples based on their informativeness. We estimate the optimal negative sampling ratio using the K value that maximizes the training effectiveness function. Based on our framework, we further propose an adaptive negative sampling method that can dynamically adjust the negative sampling ratio to improve InfoNCE-based model training. Extensive experiments in three different tasks show our framework can accurately predict the optimal negative sampling ratio, and various models can benefit from our adaptive negative sampling method.

1 Introduction

InfoNCE [Oord *et al.*, 2018] is a popular choice of contrastive learning loss, which aims to maximize a lower bound of the mutual information between a pair of variables [Arora *et al.*, 2019; He *et al.*, 2020]. It is widely used in various fields like language modeling [Chi *et al.*, 2021; Sun *et al.*, 2020], web search [Huang *et al.*, 2013; Shao *et al.*, 2020] and personalized recommendation [Wu *et al.*, 2019b; Sun *et al.*, 2019], to learn discriminative deep representations. In the InfoNCE framework, each positive sample is associated with K randomly selected negative samples, and the task is typically formulated as a K+1-way classification problem, i.e., classifying which sample is the positive one [Oord *et al.*, 2018]. In this way, the model is required to discriminate the positive sample from the negative ones, which can help estimate the variable mutual information and learn discriminative representations [Hjelm *et al.*, 2019]. It is proved that if the sample labels are clean, a larger negative sampling ratio K can lead to a tighter lower bound of variable mutual information [Oord *et al.*, 2018], which usually yields better performance. This is intuitive because more information of negative samples is exploited in model training.

Unfortunately, in many real-world tasks, sample labels are not perfect and may contain noise [Natarajan *et al.*, 2013]. For example, the non-clicked items are usually used as the negative samples in recommender system scenario. However, some non-clicked items may match users' interest and they are not clicked due to many other reasons, e.g., shown in a low position. They can be false negative sample for recommendation model training. It is not suitable to incorporate too many noisy negative samples because they may lead to misleading gradients that are harmful to model learning [Menon *et al.*, 2019]. Thus, it is important to find the appropriate negative sampling ratio K under different sample qualities to train accurate models. However, existing studies on InfoNCE mainly focus on the selection of hard negative samples [Behrmann *et al.*, 2020; Robinson *et al.*, 2020; Kalantidis *et al.*, 2020], while the study on the choice of negative sampling ratio is very limited.

In this paper, we study the problem of how many negative samples are optimal for InfoNCE in different tasks via a semi-quantitative theoretical framework. More specifically, we first propose a probabilistic model to analyze the influence of negative sampling ratio K of InfoNCE on the informativeness of training samples. Then, we propose a training effectiveness function to semi-quantitatively measure the overall influence of training samples on model learning based on their informativeness. We use the K value that maximizes this function to estimate the optimal negative sampling ratio for InfoNCE. Based on our framework, we further propose an adaptive negative sampling *(ANS)* method that can dynamically adjust the negative sampling ratio to improve model training based on

*The corresponding author.

the characteristics of different model training stages. We conduct experiments on multiple real-world datasets. The results show that our framework can effectively estimate the optimal negative ratio in different tasks, and our proposed adaptive negative sampling method can consistently achieve better performance than the commonly used negative sampling technique with fixed negative sampling ratio.

The main contributions of this paper include:

- We propose a semi-quantitative theoretical framework to analyze the influence of negative sampling ratio on InfoNCE and further estimate its optimal value.
- We propose an adaptive negative sampling method that can dynamically change the negative sampling ratio for different stages of model training.
- We conduct extensive experiments to verify the validity of our theoretical framework and the effectiveness of the proposed adaptive negative sampling method for InfoNCE based model training.

2 Related Work

2.1 InfoNCE Loss Function

InfoNCE [Oord *et al.*, 2018] is a widely used loss function for contrastive learning. It aims to estimate a lower bound of the mutual information between two variables. A relevance function $f(\cdot,\cdot)$ is used to measure the non-normalized mutual information score between them. For each positive sample (x^+, c), it is associated with K random negative samples, which are denoted as $[(x_1^-, c), (x_2^-, c), ..., (x_K^-, c)]$.[1] Then, the InfoNCE loss function $\mathcal{L}_K$ is formulated as follows:

$$\mathcal{L}_K = -\log\left(\frac{e^{f(x^+,c)}}{e^{f(x^+,c)} + \sum_{i=1}^{K} e^{f(x_i^-,c)}}\right). \quad (1)$$

According to [Oord *et al.*, 2018], if the labels are fully reliable, the lower bound of the mutual information I between x^+ and c estimated by InfoNCE is formulated as:

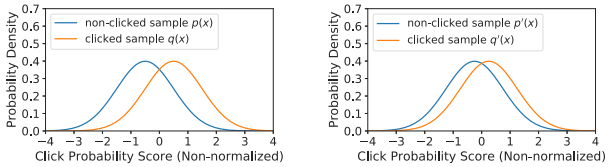
$$I(x^+, c) \geq \log(K+1) - \mathcal{L}_K. \quad (2)$$

We can see that this lower bound is tighter when the number of negative samples K is larger, which usually improves the model performance [He *et al.*, 2020].

2.2 Applications of InfoNCE

InfoNCE has wide applications in many fields like language modeling [Sun *et al.*, 2020; Chi *et al.*, 2021], search [Huang *et al.*, 2013; Chang *et al.*, 2020] and recommendation [Wu *et al.*, 2019b; Wu *et al.*, 2019c; Sun *et al.*, 2019; Wu *et al.*, 2021]. For example, Huang et al. [2013] proposed a deep structured semantic model (DSSM) for document retrieval. For each pair of query and clicked document (regarded as positive sample), they randomly sampled 4 unclicked documents as negative samples, and used the InfoNCE loss to train the model by optimizing the log-likelihood of the posterior click probability of the positive sample. Wu et al. [2019b] proposed a personalized news recommendation method named NPA. In this method, they regarded each clicked news as a positive sample, and randomly sampled 4 non-clicked news displayed in the same impression as negative samples. The model was trained via the InfoNCE framework in a similar way. Chi et al. [2021] proposed a multilingual pre-trained language model named InfoXLM. They incorporated the InfoNCE framework into several self-supervion tasks like multilingual masked language modeling and cross-lingual contrastive learning. They used the momentum contrast [He *et al.*, 2020] method to construct a queue of negative samples with a size of 131,072 for model training. In these methods, the negative sampling ratio is usually empirically selected, which requires heavy hyper-parameter search. In addition, these methods leverage a fixed negative sampling ratio, which may not be optimal for different stages of model training. Our proposed framework can help estimate the optimal negative sampling ratio in different scenarios. In addition, our proposed adaptive negative sampling method *ANS* can dynamically adjust the negative sampling ratio according to the characteristics of different model training stages. Our approach has the potential to benefit many tasks that involve InfoNCE loss in model training.

[1] We assume the selection of x is independent on c for simplicity.



(a) Real. (b) Predicted.

Figure 1: Illustrated click probability score distributions of real positive and negative samples and the predicted ones.

3 Our Framework

In this section, we introduce our semi-quantitative theoretical framework to analyze the influence of negative sampling ratio on the performance of the models trained based on InfoNCE loss. We take news recommendation as an example scenario to better explain our assumptions and framework. In this scenario, it is a common practice to regard the clicked news as positive samples and non-clicked news as negative ones [Wu *et al.*, 2019b]. A recommendation model is trained with the binary click labels to predict whether a user will click a candidate news. We assume that there exists a click probability for each sample, which reflects the relevance between the candidate news and user interest. We denote the click probability score distributions of clicked and non-clicked news as $q(x)$ and $p(x)$ respectively. In addition, we denote the click probability score distributions of predicted clicked and non-clicked news as $q'(x)$ and $p'(x)$. Fig. 1 shows an example.We expect the click probability scores of clicked news to be higher than non-clicked ones. However, since users' click behaviors have rich randomness, there are overlaps between the click probability curves of positives and negatives, which means that the labels are noisy. In addition, we assume the real labels are more discriminative (the curve overlap is smaller) than

the predicted ones, which means the prediction accuracy is limited by the label quality.

Next, we introduce several key concepts in our framework. Given a user u, a positive news x^+ and its associated K negative news $[x_1^-, x_2^-, ..., x_K^-]$ are combined as a training sample. We denote their click probability scores as y^+ and $[y_1^-, y_2^-, ..., y_K^-]$, respectively. If they satisfy $y^+ > \max(y_1^-, y_2^-, ..., y_K^-)$, we call the label of this training sample "reliable". In a similar way, we denote the click probability scores of the predicted positive and negative news as $\hat{y}^+$ and $[\hat{y}_1^-, \hat{y}_2^-, ..., \hat{y}_K^-]$, respectively. We regard the prediction for them as "reliable" if $\hat{y}^+ > \max(\hat{y}_1^-, \hat{y}_2^-, ..., \hat{y}_K^-)$. For simplicity, we assume that all the $K+1$ candidate news are independent. We define events A and B as the "reliability" of label and model prediction respectively. Then, the probabilities $P(A)$ and $P(B)$ are formulated as follows:

$$P(A) = \int_{-\infty}^{\infty} q(x)[\int_{-\infty}^{x} p(y)dy]^K dx, \tag{3}$$

$$P(B) = \int_{-\infty}^{\infty} q'(x)[\int_{-\infty}^{x} p'(y)dy]^K dx. \tag{4}$$

Then, we introduce how to measure the informativeness of training samples and their influence on model learning. If the model prediction on a sample is "unreliable" while the label is "reliable", then this sample is very informative for calibrating the model. We regard this kind of samples as "good samples", and denote their set as $\mathcal{G}$. On the contrary, if the model prediction on a sample is "reliable" but the label is "unreliable", this sample is harmful for model training because it will produce misleading gradients. We regard this kind of samples as "bad samples", and their set is denoted as $\mathcal{B}$. For the rest samples, the model predictions and labels have the same "reliability", and these samples are usually less informative for model training.[2] We call this kind of samples as "easy samples", and their set is denoted as $\mathcal{E}$. For simplicity, here we make an assumption that the events A and B are independent.[3] We denote the total number of training samples as N. Then, the number of each kind of samples introduced above is formulated as follows:

$$\begin{aligned} |\mathcal{G}| &= P(A)[1-P(B)]N, \\ |\mathcal{B}| &= P(B)[1-P(A)]N, \\ |\mathcal{E}| &= N - |\mathcal{G}| - |\mathcal{B}|. \end{aligned} \tag{5}$$

To semi-quantitatively measure the influence of training samples on model learning, we propose a training effectiveness metric v based on the different informativeness of training samples, which is formulated as follows:

$$v = \frac{1}{N}[\lambda(|\mathcal{G}| - |\mathcal{B}|) + (1-\lambda)|\mathcal{E}|], \tag{6}$$

where λ is a hyperparameter that controls the relative importance of good and bad samples. According to our massive experiments, we find that setting $\lambda = 0.9$ is appropriate for estimating the negative sampling ratio. In Eq. (6), given the distributions $p(x)$, $q(x)$, $p'(x)$ and $q'(x)$, v is only dependent on the negative sampling ratio K. Without loss of generality, we assume that these distributions are all Gaussian distributions with the same standard deviation 1, and the mean values of $p(x)$ and $p'(x)$ (denoted as $\mu_{p(x)}$ and $\mu_{p'(x)}$) satisfy $\mu_{p(x)} = \mu_{p'(x)} = 0$. We still need to estimate the mean values of $q(x)$ and $q'(x)$ (denoted as $\mu_{q(x)}$ and $\mu_{q'(x)}$). In practice, we can estimate $\mu_{q(x)}$ using the training AUC score under $K = 1$ before overfitting[4] by solving the following equation[5]:

$$AUC = \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{\frac{-(x-\mu_{q(x)})^2}{2}} [\int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy] dx. \tag{7}$$

In a similar way, the value of $\mu_{q'(x)}$ can be estimated based on the AUC score on the validation set under $K = 1$. When $\mu_{q(x)}$ and $\mu_{q'(x)}$ are known, we can compute the current training effectiveness. The relation between the AUC score and the value of $\mu_{q(x)}$ (or $\mu_{q'(x)}$) in our framework is shown in Fig. 9 in supplements. The value of $\mu_{q(x)}$ is 0 if AUC is 0.5, and increases when AUC gets close to 1.

Finally, we introduce how to estimate the optimal negative sampling ratio based on our proposed framework. Since the model may produce different $p'(x)$ and $q'(x)$ during the training process, the training effectiveness measurement is time-variant. Thus, we denote the training effectiveness value at the i-th iteration step as v_i, and the overall training effectiveness v is computed as $v = \frac{1}{T}\sum_{i=1}^{T} v_i$, where T is the number of iterations needed for model convergence. Since v is the function of K and other variables can be approximated, we can estimate the optimal value of K that maximizes v.

MIND			
# News	161,013	# User	1,000,000
# Impression	15,777,377	# Click behavior	24,155,470
Avg. title len.	11.52	Avg. body len.	585.05
ML-1M			
# Item	3,706	# User	6,040
# Interaction	1,000,209	Avg. history len.	165.6

Table 1: Detailed dataset statistics.

4 Experiments

In this section, we conduct experiments to verify our proposed framework. We first introduce the datasets and experimental settings, and then introduce the results and findings.

4.1 Datasets and Experimental Settings

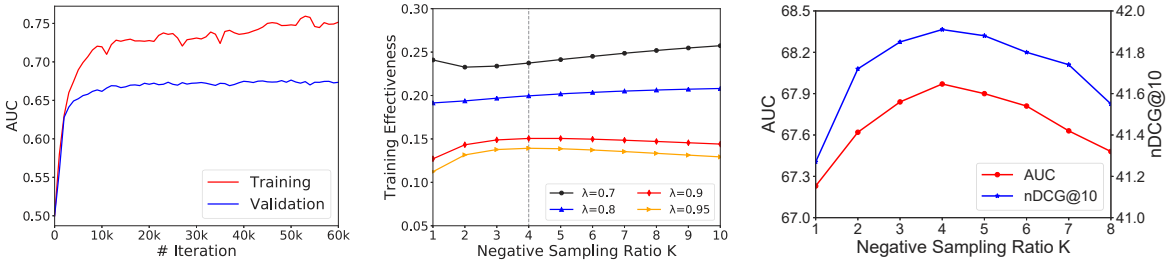
In our experiments, we verify our framework in three tasks, i.e., news recommendation, news title-body matching and item recommendation. The first two tasks are performed on

[2]When predictions and labels are both "reliable", the model can simply optimize the losses on these samples, which may lead to overfitting. When predictions and labels are both not "reliable", the gradients are also not very informative.

[3]We assume that in the sense of expectation, the events A and B can be regarded as"less correlated" because the prediction errors on different samples may be counteracted.

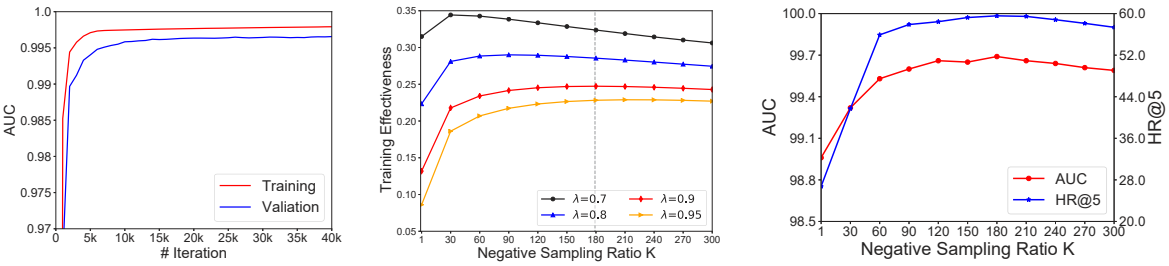
[4]If the model can well fit the training data and meanwhile do not overfit, the training AUC can indicate the label quality.

[5]We cannot obtain a closed-form solution of $\mu_{q(x)}$ due to the characteristic of Gaussian distribution. Thus, we need to solve $\mu_{q(x)}$ with numerical methods (e.g., bisection method).



(a) Training and validation AUC. (b) Overall training effectiveness under different K and λ. (c) Performance under different K.

Figure 2: Model convergence curves as well as the estimated training effectiveness and real experimental performance under different K in news recommendation. Gray dashed line indicates the optimal K under $\lambda = 0.9$.



(a) Training and validation AUC. (b) Overall training effectiveness under different K and λ. (c) Performance under different K.

Figure 3: Model convergence curves as well as the estimated training effectiveness and real experimental performance under different K in movie recommendation. Gray dashed line indicates the optimal K under $\lambda = 0.9$, which is consistent with Figs. 2 and 11.

the MIND [Wu *et al.*, 2020] dataset[6], which contains the news impression logs of 1 million users in 6 weeks. For the news recommendation task, each clicked news is regarded as a positive sample and the news displayed in the same impression but not clicked by the user are regarded as negative samples. The logs in the last week are used for test, and the rest are used for training and validation. For the news title-body matching task, we regard the original news title-body pairs as positive samples, and negative samples are created by randomly pairing titles and bodies.[7] We use the news in the training set of MIND for model training, and those in the validation and test sets (except the news included in the training set) for validation and test respectively. The item recommendation task is performed on the MovieLens dataset [Harper and Konstan, 2015], and we use the ML-1M [8] version for experiments. We use the same experimental settings as [Sun *et al.*, 2019]. The statistics of datasets are listed in Table 1.

In these experiments, we use NRMS [Wu *et al.*, 2019c] as the base model for news recommendation, a Siamese Transformer model [Reimers and Gurevych, 2019] for title-body matching, and use BERT4Rec [Sun *et al.*, 2019] for item recommendation. We use Adam [Kingma and Ba, 2015] as the optimizer and the learning rate is 1e-4. The batch size is 32. The hidden dimension is 256 in the news recommendation and title-body matching tasks, and 64 in the item recommendation task. We use AUC and nDCG@10 to measure the performance of news and item recommendation, and use AUC and HR@5 as the metrics for news title-body matching. All these models are trained with the InfoNCE loss under different negative sampling ratios. Each experiment is repeated 5 times and the average performance is reported.

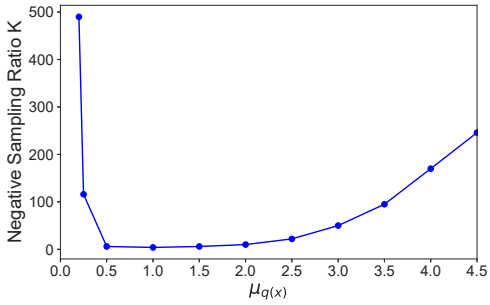


Figure 4: The simulated optimal K under different $\mu_{q(x)}$.

[6]https://msnews.github.io/

[7]Some randomly combined title-body pairs happen to be from the same news, i.e., they are actually positive samples. We remove these pairs from the negative sample set.

[8]https://grouplens.org/datasets/movielens/1m/

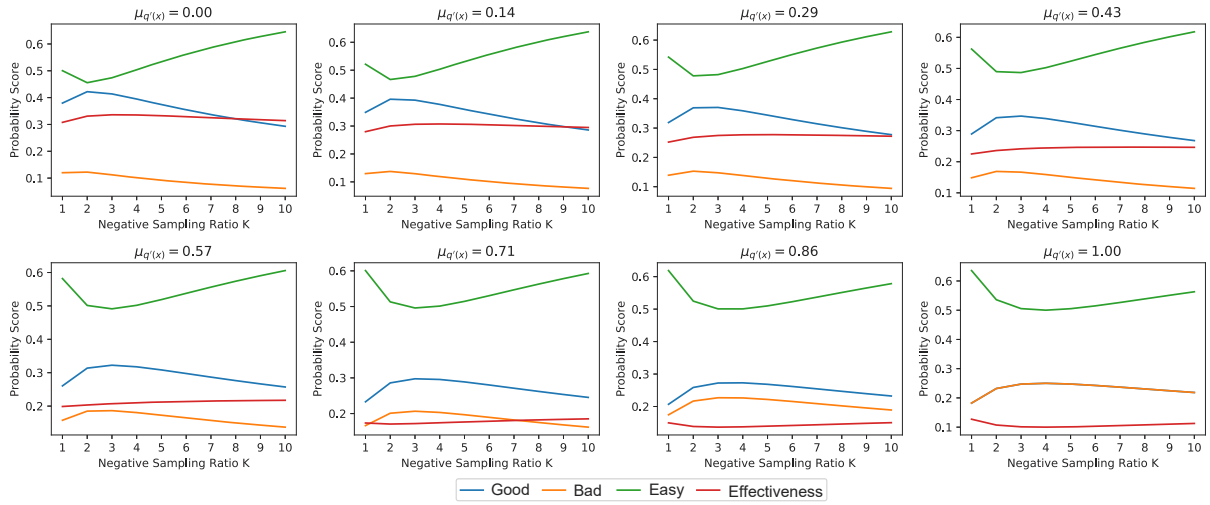


Figure 5: Training effectiveness and proportion of different kinds of samples in different training stages in the news recommendation task. Good and bad curves overlap in the last plot.

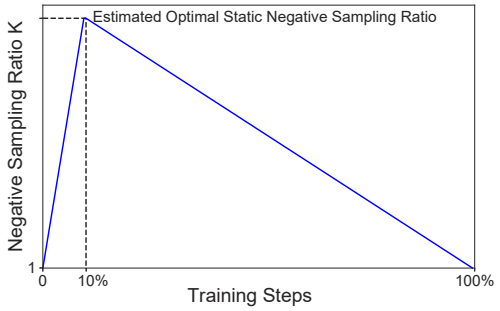


Figure 6: The curve of negative sampling ratio in *ANS*.

4.2 Experimental Results

In this section, we verify our framework according to the experiments on different tasks. Fig. 2 shows the model training and validation AUC curves, the estimated overall training effectiveness under different negative sampling ratio K and hyperparameter λ, and the real model performance under different negative sampling ratios K in the news recommendation task. From Fig. 2(a), we can observe that the training AUC is around 0.75 when model converges, which means that $\mu_{q(x)}$ is about 1. From Figs. 2(c), we find that the real model performance is not optimal when K is too small or too large, and $K = 4$ yields the best performance, which is consistent with the red ($\lambda = 0.9$) and orange ($\lambda = 0.95$) curves in Fig. 2(b). It shows that the value of λ should be relatively large, which is intuitive because the influence of good and bad samples on the model training is usually dominant. Fig. 3 shows the results in the movie recommendation task. The training AUC is about 0.95, which approximately corresponds to $\mu_{q(x)} = 2.4$ in our theoretical framework. We find it interesting that when $\lambda = 0.9$, the estimated optimal negative sampling ratio (about 20) in Fig. 3(b) is consistent with the experimental results in Fig. 3(c). Thus, setting the hyperparameter λ to 0.9 would be appropriate for estimating the optimal negative sampling ratio. In addition, from these figures we find that the model performance improves first when K increases, and then starts to decline when K becomes too large. This is because useful information in negative samples cannot be exploited when K is too small, while the label noise harms model training when K is too large. Thus, a medium value of K may be more suitable for model training with noisy labels.

4.3 Analysis

In this section, we use our framework to further analyze the negative sampling ratio in InfoNCE. We set the value of λ to 0.9 in the following analysis. First, we study the influence of the label quality (represented by $\mu_{q(x)}$) on the estimated optimal value of K. We use the function $\mu^t_{q'(x)} = \mu_{q(x)}(1 - e^{-t}), t \in [0, 3]$ to simulate the model training curve[9], and the estimated optimal values K under different $\mu_{q(x)}$ are shown in Fig. 4. We find it interesting that the optimal value of K boosts when $\mu_{q(x)}$ is close to 0 or relatively large. This may be because when $\mu_{q(x)}$ is very small, we need many negative samples to counteract the noise. And when $\mu_{q(x)}$ is large, we may also need increase the number of negative samples because the labels are relatively reliable.

Next, we study how the training effectiveness and the proportions of different kinds of samples change at model training stages (represented by different $\mu_{q'(x)}$). The results in the news recommendation task is shown in Fig. 5 (results on other tasks show similar phenomena). We have two interesting findings from them. First, the optimal negative sampling ratio at the beginning of model training is smaller than the globally optimal one estimated in the previous section. This may be because when model is not discriminative, the ratio of good samples is much larger than bad samples, and using too many negative samples may introduce unwanted noise and reduce the ratio of good samples. Thus, a smaller value of K may be more appropriate at the beginning. Second, when the

[9]We find the shape of this curve is similar to real training curves.

model gets to converge, the negative sampling ratio also needs to be smaller, and the optimal one is 1 when $\mu_{q(x)} = \mu_{q'(x)}$. This is because easy samples are dominant when the numbers of good and bad samples are almost even, and focusing on optimizing the loss on easy samples may lead to overfitting. Thus, a fixed negative sampling ratio may be suboptimal for model training at different stages.

5 ANS: Adaptive Negative Sampling

Based on the findings in above section, we further propose an Adaptive Negative Sampling *(ANS)* method that can dynamically adjust the negative sampling ratio K during model training to overcome the limitation of using a fixed one. We first introduce an extension of the standard negative sampling method where the negative sampling ratio K is real-valued. We denote the set of negative samples associated with each positive sample as $\mathcal{N}$, which satisfies the following formulas:

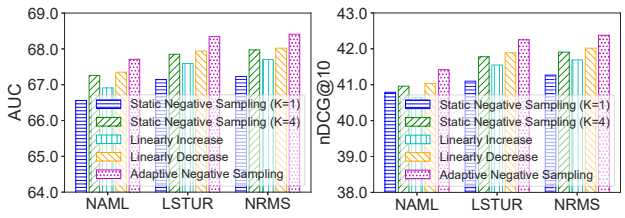
$$\begin{aligned} P(|\mathcal{N}| = [K]) &= 1 - \{K\}, \\ P(|\mathcal{N}| = [K] + 1) &= \{K\}, \end{aligned} \tag{8}$$

where $[x]$ and $\{x\}$ represent the integral and fractional parts of x respectively. Based on this extension, we then introduce our proposed *ANS* method. Motivated by the analysis in the previous section and the popular learning rate warm-up strategy [Goyal *et al.*, 2017], we propose to adjust the negative sampling ratio K as the curve shown in Fig. 6. The value of K quickly grows from 1 to the optimal value estimated by our framework, and then slowly declines to 1 as the training continues. This is because the optimal negative sampling ratio may be small at the beginning and the end of model training, and the performance of model usually improves quickly at the beginning. The turning point on this curve is 10% of the training steps, which is empirically selected based on experimental results. Our *ANS* method can adapt to different stages in the model training process, which can overcome the drawbacks of using a fixed negative sampling ratio.

6 Experiments on the ANS Method

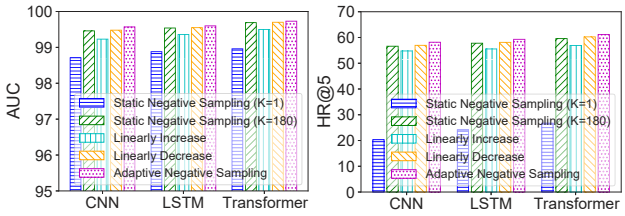
In this section, we conduct experiments to verify the effectiveness of our proposed *ANS* method. In the news recommendation task, we apply our *ANS* method to several state-of-the-art baseline methods, including NRMS [Wu *et al.*, 2019c], LSTUR [An *et al.*, 2019] and NAML [Wu *et al.*, 2019a]. In the news title-body matching task, we apply *ANS* to the Siamese Transformer [Reimers and Gurevych, 2019] network and its variants based on LSTM or CNN. We compare their performance with those trained with static negative sampling strategies with a fixed negative sampling ratio and two dynamic negative sampling methods that linearly increase K from 1 to the estimated optimal value or decrease K from the optimal value to 1.[10] The results are shown in Figs. 7 and 8. We find that using the optimal negative sampling ratio estimated by our framework is better than popular default negative sampling ratio (e.g., 1) [Rendle *et al.*, 2009]. Moreover, using our proposed adaptive negative sampling method

[10]We do not compare hard-negative sampling methods because this work focuses on choosing the number of negative samples.



(a) AUC. (b) nDCG@10.

Figure 7: News recommendation experiments with different negative sampling strategies.



(a) AUC. (b) HR@5.

Figure 8: News title-body matching experiments with different negative sampling strategies.

can achieve better performance than using static or linearly scheduled negative sampling ratios. This is because *ANS* can use different negative sampling ratios to better fit the characteristics of different training stages.

7 Conclusion

In this paper, we study how many negative samples are optimal for InfoNCE-based model learning in different tasks using a semi-quantitative theoretical framework. We first propose a probabilistic model to analyze the influence of negative sampling ratio in InfoNCE on the informativeness of training samples. Then, we propose a training effectiveness function to measure the overall influence of training samples on model learning based on their informativeness. We further estimate the optimal value of K that maximizes this measurement. Based on our framework, we further propose an adaptive negative sampling method that can dynamically adjust the negative sampling ratio according to the characteristics of different model training stages. We conduct extensive experiments on different real-world datasets for different tasks. The results show that our framework can accurately estimate the optimal negative sampling ratio, and our adaptive negative sampling method can consistently outperform the commonly used fixed negative sampling ratio strategy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant numbers 2021ZD0113902, U1936208, U1836204, U1936216, and 61862002.

References

[An *et al.*, 2019] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. Neural news recommendation with long-and short-term user representations. In *ACL*, pages 336–345, 2019.

[Arora *et al.*, 2019] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, pages 9904–9923, 2019.

[Behrmann *et al.*, 2020] Nadine Behrmann, Jurgen Gall, and Mehdi Noroozi. Unsupervised video representation learning by bidirectional feature prediction. In *CVPR*, pages 1670–1679, 2020.

[Chang *et al.*, 2020] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*, 2020.

[Chi *et al.*, 2021] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. 2021.

[Goyal *et al.*, 2017] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *TIIS*, 5(4):1–19, 2015.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[Hjelm *et al.*, 2019] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338, 2013.

[Kalantidis *et al.*, 2020] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Menon *et al.*, 2019] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *ICLR*, 2019.

[Natarajan *et al.*, 2013] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *NIPS*, 26:1196–1204, 2013.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3973–3983, 2019.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Robinson *et al.*, 2020] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[Shao *et al.*, 2020] Jie Shao, Xin Wen, Bingchen Zhao, Changhu Wang, and Xiangyang Xue. Context encoding for video retrieval with contrastive learning. *arXiv preprint arXiv:2008.01334*, 2020.

[Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, pages 1441–1450, 2019.

[Sun *et al.*, 2020] Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. Contrastive distillation on intermediate representations for language model compression. In *EMNLP*, pages 498–508, 2020.

[Wu *et al.*, 2019a] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869, 2019.

[Wu *et al.*, 2019b] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Npa: Neural news recommendation with personalized attention. In *KDD*, pages 2576–2584, 2019.

[Wu *et al.*, 2019c] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*, pages 6390–6395, 2019.

[Wu *et al.*, 2020] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606, 2020.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, pages 726–735, 2021.