

PAC-Bayesian Contrastive Unsupervised Representation Learning

Kento Nozawa, Pascal Germain, Benjamin Guedj

▶ **To cite this version:**

Kento Nozawa, Pascal Germain, Benjamin Guedj. PAC-Bayesian Contrastive Unsupervised Representation Learning. UAI 2020 - Conference on Uncertainty in Artificial Intelligence, Aug 2020, Toronto, Canada. hal-02401282

HAL Id: hal-02401282
https://inria.hal.science/hal-02401282

Submitted on 9 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:1910.04464v1 [cs.LG] 10 Oct 2019

PAC-Bayesian Contrastive Unsupervised Representation Learning

Kento Nozawa
University of Tokyo & RIKEN

Pascal Germain
Inria

Benjamin Guedj
Inria & University College London

Abstract

Contrastive unsupervised representation learning (CURL) is the state-of-the-art technique to learn representations (as a set of features) from unlabelled data. While CURL has collected several empirical successes recently, theoretical understanding of its performance was still missing. In a recent work, Arora et al. (2019) provide the first generalisation bounds for CURL, relying on a Rademacher complexity. We extend their framework to the flexible PAC-Bayes setting, allowing to deal with the non-iid setting. We present PAC-Bayesian generalisation bounds for CURL, which are then used to derive a new representation learning algorithm. Numerical experiments on real-life datasets illustrate that our algorithm achieves competitive accuracy, and yields generalisation bounds with non-vacuous values.

1 Introduction

Unsupervised representation learning (Bengio et al., 2013) aims at extracting features representation from an unlabelled dataset for downstream tasks such as classification and clustering (see Zhang et al., 2016; Noroozi and Favaro, 2016; Caron et al., 2018; Mikolov et al., 2013; Devlin et al., 2019). An unsupervised representation learning model is typically learned by solving a pretext task without supervised information. Trained model work as a feature extractor for supervised tasks.

In unsupervised representation learning, contrastive loss is a widely used objective function class. Contrastive loss uses two types of data pair, namely, similar pair and dissimilar pair. Their similarity is defined without label information of a supervised task. For example, in word representation learning, Mikolov et al. (2013) define a similar pair as co-occurrence words in the same context, while dissimilar pairs are randomly sampled from a fixed distribution. Intuitively, by minimising a contrastive loss, similar data samples are mapped to similar representations in feature space in term of some underlying metric (as the inner product), and dissimilar samples are not mapped to similar representations.

Contrastive unsupervised representation learning improves the performance of supervised models in practice, although usage is still quite far ahead theoretical understanding. Recently, Arora et al. (2019) introduced a theoretical framework for contrastive unsupervised representation learning and derived the first generalisation bounds for CURL. In parallel, PAC-Bayes is emerging as a principled tool to understand and quantify the generalisation ability of many machine learning algorithms, including deep neural networks (as recently studied by Dziugaite and Roy, 2017; Letarte et al., 2019).

Our contributions. We extend the framework introduced by Arora et al. (2019), by adopting a PAC-Bayes approach to contrastive unsupervised representation learning. We derive the first PAC-Bayes generalisation bounds for CURL, both in iid and non-iid settings. Our bounds are then used to derive new CURL algorithms, for which we provide a complete implementation. The paper closes with numerical experiments on two real-life datasets (`CIFAR-100` and `AUSLAN`) showing that our bounds are non-vacuous.

2 Contrastive Unsupervised Representation Learning

2.1 Learning Framework

Inputs are denoted $\mathbf{x} \in X = \mathbb{R}^{d_0}$, and outputs are denoted $y \in Y$, where Y is a discrete and finite set. The *representation* is learned from a (large) unlabelled dataset $U = \{\mathbf{z}_i\}_{i=1}^m$, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_{i1}^-, \ldots, \mathbf{x}_{ik}^-)$ is a tuple of $k+2$ elements; $\mathbf{x}_i$ being *similar* to $\mathbf{x}_i^+$

Preprint.

and *dissimilar* to every elements of the *negative sample set* $\{\mathbf{x}_{ij}^-\}_{j=1}^k$. The predictor is learned from a labelled dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

In the following, we present the contrastive framework proposed by Arora et al. (2019) in a simplified scenario in order to highlight the key ideas, where the supervised prediction task is binary and the negative sample sets for unsupervised representation learning contain one element. Thus, we choose the label set to be $Y = \{-1, 1\}$, and the unsupervised set U contains triplets $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$. The extension to a more generic setting (for $|Y| > 2$ and $k > 1$) bears no particular difficulty and is deferred to the appendix. It is important to note at this stage that both U and S are assumed to be **iid** (independent, identically distributed) collections, as also assumed by Arora et al. (2019).

Latent classes and data distributions. The main assumption is the existence of a set of *latent classes* $\mathcal{C}$. Let us denote by ρ a probability distribution over $\mathcal{C}$. Moreover, with each class $c \in \mathcal{C}$, comes a class distribution $\mathcal{D}_c$ over the input space X. A similar pair $(\mathbf{x}, \mathbf{x}^+)$ is such that both $\mathbf{x}$ and $\mathbf{x}^+$ are generated by the same class distribution. Note that an input $\mathbf{x}$ possibly belongs to multiple classes: take the example of $\mathbf{x}$ being an image and $\mathcal{C}$ a set of latent classes including "the image depicts a dog" and "the image depicts a cat" (both classes are not mutually exclusive).

Definition 1. Let ρ^2 be a shorthand notation for the joint distribution (ρ, ρ). We refer to the *unsupervised data distribution* $\mathcal{U}$ as the process that generates an unlabelled sample $\mathbf{z} = (\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)$ according to the following scheme:

1. Draw two latent classes $(c^+, c^-) \sim \rho^2$;
2. Draw two similar samples $(\mathbf{x}, \mathbf{x}^+) \sim (\mathcal{D}_{c^+})^2$;
3. Draw a negative sample $\mathbf{x}^- \sim \mathcal{D}_{c^-}$.

The labelled sample S is obtained by fixing two classes $c^\pm = \{c^-, c^+\} \in \mathcal{C}^2$ (from now on, the shorthand notation $c^\pm$ is used to refer to a pair of latent classes). Each class is then mapped on a label of Y. We fix $y_{c^-} = -1$ and $y_{c^+} = 1$; Thus we can write $Y = \{y_{c^-}, y_{c^+}\}$ as an ordered set. The label is obtained from the latent class distribution restricted to two values $\rho_{c^\pm}$:

$$\rho_{c^\pm}(c^-) = \frac{\rho(c^-)}{\rho(c^-) + \rho(c^+)}\,, \quad \rho_{c^\pm}(c^+) = \frac{\rho(c^+)}{\rho(c^-) + \rho(c^+)}\,.$$

Definition 2. We refer to the *supervised data distribution* $\mathcal{S}$ as the process that generates a labelled sample $(\mathbf{x}, y)$ according to the following scheme:

1. Draw a class $c \sim \rho_{c^\pm}$ and set label $y = y_c$;
2. Draw a sample $\mathbf{x} \sim \mathcal{D}_c$.

Loss function. The learning process is divided in two sequential steps, the unsupervised and supervised steps. In order to relate these two steps, the key is to express them in terms of a common convex loss function $\ell : \mathbb{R} \to \mathbb{R}^+$. Typical choices are

$$\ell_{\log}(v) \coloneqq \log_2(1 + e^{-v})\,, \qquad \textit{(logistic loss)} \tag{1}$$
$$\ell_{\text{hinge}}(v) \coloneqq \max\{0, 1-v\}\,, \qquad \textit{(hinge loss)} \tag{2}$$

where the loss argument v expresses a notion of *margin*.

In the first step, an unsupervised representation learning algorithm produces a feature map $\mathbf{f} : X \to \mathbb{R}^d$. The *contrastive loss* associated to $\mathbf{f}$ is defined as

$$L_{\text{un}}(\mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{(c^+,c^-)\sim\rho^2} \mathop{\mathbb{E}}_{\substack{(\mathbf{x},\mathbf{x}^+)\sim\mathcal{D}_{c^+}^2 \\ \mathbf{x}^-\sim\mathcal{D}_{c^-}}} \ell\Big(\mathbf{f}(\mathbf{x})\cdot\big[\mathbf{f}(\mathbf{x}^+)-\mathbf{f}(\mathbf{x}^-)\big]\Big)$$
$$= \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-)\sim\mathcal{U}} \ell\Big(\mathbf{f}(\mathbf{x})\cdot\big[\mathbf{f}(\mathbf{x}^+)-\mathbf{f}(\mathbf{x}^-)\big]\Big)\,.$$

More precisely, from the unsupervised training dataset $U = \{(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)\}_{i=1}^m \sim \mathcal{U}^m$, we are interested in learning the feature map $\mathbf{f}$ that minimises the following empirical contrastive loss:

$$\widehat{L}_{\text{un}}(\mathbf{f}) \coloneqq \frac{1}{m}\sum_{i=1}^m \ell\Big(\mathbf{f}(\mathbf{x}_i)\cdot\big[\mathbf{f}(\mathbf{x}_i^+) - \mathbf{f}(\mathbf{x}_i^-)\big]\Big)\,. \tag{3}$$

In the second step, a supervised learning algorithm is given the *mapped* dataset $\widehat{S} \coloneqq \{(\hat{\mathbf{x}}_i, y_i)\}_{i=1}^n$, with $\hat{\mathbf{x}}_i \coloneqq \mathbf{f}(\mathbf{x}_i)$, and returns a predictor $g : \mathbb{R}^d \to \mathbb{R}$. For a fixed pair $c^\pm = \{c^-, c^+\}$, the predicted label on an input $\mathbf{x}$ is then obtained from $\hat{y} = \text{sgn}[g(\hat{\mathbf{x}})]$ (recall that $Y = \{-1, 1\}$), and we aim to minimise the supervised loss

$$L_{\text{sup}}(g \circ \mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{c\sim\rho_{c^\pm}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_c} \ell\Big(y_c\, g(\mathbf{f}(\mathbf{x}))\Big)$$
$$= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \ell\Big(y\, g(\mathbf{f}(\mathbf{x}))\Big)\,.$$

Given a labelled dataset $S \sim \mathcal{S}^n$, the empirical counterpart of the above supervised loss is

$$\widehat{L}_{\text{sup}}(g \circ \mathbf{f}) \coloneqq \frac{1}{n}\sum_{i=1}^n \ell\Big(y_i\, g(\mathbf{f}(\mathbf{x}_i))\Big)\,.$$

Mean classifier. Following Arora et al. (2019), we study the mean classifier defined from the linear function

$$g_{c^\pm}(\hat{\mathbf{x}}) \coloneqq \mathbf{w}_{c^\pm} \cdot \hat{\mathbf{x}}\,,$$

where $\mathbf{w}_{c^\pm} \coloneqq \boldsymbol{\mu}_{c^+} - \boldsymbol{\mu}_{c^-}$, and $\boldsymbol{\mu}_c \coloneqq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_c} \mathbf{f}(\mathbf{x})$.

Then, the *supervised average loss* of the mean classifier is the expected loss on a dataset whose pair of labels is sampled from the latent class distribution ρ.

$$L_{\text{sup}}^{\mu}(\mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{c^\pm\sim\rho_{\text{w/o}}^2} L_{\text{sup}}(g_{c^\pm} \circ \mathbf{f})\,, \tag{4}$$

with $\rho^2_{\text{w/o}}$ being a shorthand notation for the sampling *without replacement* of two classes among $\mathcal{C}$. Indeed, we want positive and negative samples that are generated by distinct latent class distributions, *i.e.*, $c^- \neq c^+$.

2.2 Generalisation Guarantees

A major contribution of the framework introduced by Arora et al. (2019) is that it rigorously links the unsupervised representation learning task and the subsequent prediction task: it provides generalisation guarantees on the supervised average loss of Eq. (4) in terms of the empirical contrastive loss in Eq. (3). Central to this result is the upcoming Lemma 3, that relates the supervised average loss of the mean classifier to its unsupervised loss.

Lemma 3 (Arora et al., 2019, Lemma 4.3)**.** *Given a latent class distribution ρ on $\mathcal{C}$ and a convex loss $\ell : \mathbb{R} \to \mathbb{R}$, for any feature map $\mathbf{f} : \mathbb{R}^{d_0} \to \mathbb{R}^d$, we have*

$$L^{\mu}_{\text{sup}}(\mathbf{f}) \;\leq\; \frac{1}{1-\tau}\left(L_{\text{un}}(\mathbf{f}) - \tau\right),$$

where τ is the probability of sampling twice the same latent class ($\mathbf{1}[\cdot]$ is the indicator function):

$$\tau \;\coloneqq\; \mathop{\mathbb{E}}_{c^{\pm}\sim\rho^2} \mathbf{1}[c^+ \neq c^-] \;=\; \sum_{c\in\mathcal{C}} [\rho(c)]^2 . \tag{5}$$

Arora et al. (2019) upper bound the unsupervised contrastive loss in Lemma 3 by its empirical estimates. The obtained generalisation guarantee is presented by the following Theorem 4. The bound focuses on a class of feature map functions $\mathcal{F}$ through its empirical Rademacher complexity on a training dataset U, defined by

$$\mathcal{R}_U(\mathcal{F}) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim\{\pm1\}^{3dm}} \left(\sup_{f\in\mathcal{F}} \left[\boldsymbol{\sigma} \cdot \mathbf{f}_{|U} \right] \right),$$

where $\mathbf{f}_{|U} \coloneqq \text{vec}(\{\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_i^+), \mathbf{f}(\mathbf{x}_i^-)\}_{i=1}^m) \in \mathbb{R}^{3dm}$ is the concatenation of all feature mapping given by $\mathbf{f}$ on U, and $\boldsymbol{\sigma}\sim\{\pm1\}^{3dm}$ denotes the uniformly sampled Rademacher variables over that "representation" space.

Theorem 4 (Arora et al., 2019, Theorem 4.1)**.** *Let $B \in \mathbb{R}_+$ be such that $\|\mathbf{f}(\cdot)\| \leq B$, with probability $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathbf{f}\in\mathcal{F}$*

$$L^{\mu}_{\text{sup}}(\widehat{\mathbf{f}}) \leq \frac{1}{1-\tau}\left(L_{\text{un}}(\mathbf{f}) - \tau\right) + \frac{1}{1-\tau}\,\mathcal{O}\left(B\frac{\mathcal{R}_U(\mathcal{F})}{m} + B^2\sqrt{\frac{\log\frac{1}{\delta}}{m}} \right),$$

where $\hat{\mathbf{f}} \coloneqq \operatorname*{argmin}_{\mathbf{f}\in\mathcal{F}} \widehat{L}_{\text{un}}(\mathbf{f})$.

3 PAC-Bayes analysis

Among the different techniques to analyse generalisation in statistical learning theory, PAC-Bayes has emerged in the late 90s as a promising alternative to Rademacher complexity. PAC-Bayes (pioneered by Shawe-Taylor and Williamson, 1997; McAllester, 1998; Catoni, 2003, 2004, 2007 – see Guedj, 2019 for a recent survey) consists in obtaining PAC (probably approximately correct, Valiant, 1984) generalisation bounds for Bayesian-flavoured predictors. PAC-Bayes bounds typically hold with arbitrarily high probability and express a tradeoff between the empirical risk on the training set and a measure of complexity of the predictors class. A particularity of PAC-Bayes is that the complexity term relies on a divergence measure between a prior belief and a data-dependent posterior distribution (typically the Kullback-Leibler divergence).

3.1 Supervised Learning Framework

Let $\mathcal{P}$ be a prior over a predictor class $\mathcal{H}$, which cannot depend on training data, and let $\mathcal{Q}$ be a posterior over the predictor class $\mathcal{H}$, which can depend on the training data. Any predictor $h \in \mathcal{H}$ is a classification function $h : X \to Y$. Most PAC-Bayes results measure the *discrepancy* between the prior and the posterior distributions through the Kullback–Leibler divergence,

$$\text{KL}(\mathcal{P}\|\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{h\sim\mathcal{P}} \ln\frac{\mathcal{P}(h)}{\mathcal{Q}(h)} . \tag{6}$$

Moreover, PAC-Bayes provides bounds on the expected loss of the predictors under the distribution $\mathcal{Q}$. Let us present the classical setup, where the zero-one loss is used. We refer to this loss as the classification risk, denoted by $r(y, \hat{y}) \coloneqq \mathbf{1}[y\,\hat{y} < 0]$.[1] Given a data-generating distribution $\mathcal{S}$ on $X \times Y$, the expected $\mathcal{Q}$-risk is

$$R(\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} r(y, h(\mathbf{x})) ,$$

and the empirical counterpart, *i.e.*, the $\mathcal{Q}$-weighted empirical risk on a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{S}^n$, is given by

$$\widehat{R}(\mathcal{Q}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} r(y_i, h(\mathbf{x}_i)) .$$

The following Theorem 5 expresses an upper bound on the risk $R(\mathcal{Q})$, from the empirical risk $\widehat{R}(\mathcal{Q})$ and the posterior-prior divergence $\text{KL}(\mathcal{Q}\|\mathcal{P})$.

[1] See Appendix B for a contrastive risk with k negative samples.

Theorem 5 (Catoni, 2007, Theorem 1.2.6)**.** *Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{H}$, with probability at least $1-\delta$ over training samples $S \sim \mathcal{S}^n$, $\forall \mathcal{Q}$ over $\mathcal{H}$,*

$$R(\mathcal{Q}) \leq \frac{1 - \exp\left(-\lambda \widehat{R}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{n}\right)}{1 - \exp\left(-\lambda\right)} . \quad (7)$$

3.2 PAC-Bayes Representation Learning

We now proceed to the first of our contributions. We prove a PAC-Bayesian bound on the contrastive unsupervised representation loss, by replacing the Rademacher complexity in Theorem 4 with a Kullback-Leibler divergence.

To do so, we consider a prior $\mathcal{P}$ and posterior $\mathcal{Q}$ distributions over a class of feature mapping functions $\mathcal{F} \coloneqq \{\mathbf{f} \in X \to \mathbb{R}^d\}$.

First, let us remark that we can adapt Theorem 5 to a bound on the unsupervised expected *contrastive risk* defined as

$$R_{\mathrm{un}}(\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-)\sim\mathcal{U}} \mathop{\mathbb{E}}_{\mathbf{f}\sim\mathcal{Q}} r\Big(\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-), \mathbf{f}(\mathbf{x})\Big),$$

where $r(\mathbf{y}, \hat{\mathbf{y}}) \coloneqq \mathbf{1}[\mathbf{y} \cdot \hat{\mathbf{y}} < 0]$ is the zero-one loss extended to vector arguments. We denote $\widehat{R}_{\mathrm{un}}(\mathcal{Q})$ the empirical counterpart of $R_{\mathrm{un}}(\mathcal{Q})$ computed on the unsupervised training set $U \sim \mathcal{U}^m$. Once expressed this way, Theorem 5—devoted to classical supervised learning—can be straightforwardly adapted for the expected contrastive risk. Thus, we obtain the following Corollary 6.

Corollary 6. *Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathcal{Q}$ over $\mathcal{F}$,*

$$R_{\mathrm{un}}(\mathcal{Q}) \leq \frac{1 - \exp\left(-\lambda \widehat{R}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp\left(-\lambda\right)} .$$

Unfortunately, the bound on the contrastive risk $R_{\mathrm{un}}(\cdot)$ does not translate directly to a bound on the supervised average risk

$$R^{\mu}_{\mathrm{sup}}(\mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{c^{\pm}\sim\rho^2_{\mathrm{w/o}}} R_{\mathrm{sup}}(g_{c^{\pm}} \circ \mathbf{f}) . \quad (8)$$

This is because the zero-one loss is not convex, preventing us to apply Lemma 3 to obtain a result analogous to Theorem 4. However, note that both loss functions defined by Equations (1-2) are upper bound on the zero-one loss:

$$\forall \mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^d : \; r(\mathbf{y}, \hat{\mathbf{y}}) \leq \ell(\mathbf{y} \cdot \hat{\mathbf{y}}) , \text{ with } \ell \in \{\ell_{\log}, \ell_{\mathrm{hinge}}\} .$$

Henceforth, we study the $\mathcal{Q}$ expected loss $L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) = \mathbb{E}_{\mathbf{f}\sim\mathcal{Q}} L^{\mu}_{\mathrm{sup}}(\mathbf{f})$ in regards to $L_{\mathrm{un}}(\mathcal{Q}) = \mathbb{E}_{\mathbf{f}\sim\mathcal{Q}} L_{\mathrm{un}}(\mathbf{f})$. By assuming that the representation vectors are bounded, *i.e.*, $\|\mathbf{f}(\cdot)\| \leq B$ for some $B \in \mathbb{R}^+$ as in Theorem 4, we also have that the loss function is bounded. Thus, by rescaling in $[0,1]$ the loss function, Theorem 5 can be used to derive the following Theorem 7, which is the PAC-Bayesian doppelgänger of Theorem 4.

Theorem 7. *Let $B \in \mathbb{R}_+$ such that $\|\mathbf{f}(\cdot)\| \leq B$ for all $\mathbf{f} \in \mathcal{F}$. Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathcal{Q}$ over $\mathcal{F}$,*

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \leq \frac{1}{1-\tau}\left(B_\ell \frac{1 - \exp\left(-\frac{\lambda}{B_\ell}\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp(-\lambda)} - \tau\right), \quad (9)$$

with $B_\ell \coloneqq \max\{\ell(-2B), \ell(2B)\}$ and τ given by Eq. (5).

Proof. Since $\|\mathbf{f}(\cdot)\| \leq B$, we have $\forall \mathbf{x}, \mathbf{x}^+, \mathbf{x}^- \in X$[3]:

$$-2B \leq \mathbf{f}(\mathbf{x}) \cdot [\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)] \leq 2B .$$

Thus, $\ell(\mathbf{f}(\mathbf{x}) \cdot [\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)]) \leq B_\ell$, as ℓ is both convex and positive. Therefore, the output of the rescaled loss function $\ell'(\cdot) \coloneqq \frac{1}{B_\ell}\ell(\cdot)$ belongs to $[0,1]$. From that point, we apply Theorem 5 to obtain[2], with probability at least $1-\delta$,

$$\frac{1}{B_\ell} L_{\mathrm{un}}(\mathcal{Q}) \leq \frac{1 - \exp\left(-\lambda \frac{1}{B_\ell}\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp\left(-\lambda\right)} .$$

Also, since the inequality stated in Lemma 3 holds true for all $\mathbf{f} \in \mathcal{F}$, taking the expected value according to $\mathcal{Q}$ gives

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \;\leq\; \frac{1}{1-\tau}\left(L_{\mathrm{un}}(\mathcal{Q}) - \tau\right) .$$

The desired result is obtained by replacing $L_{\mathrm{un}}(\mathcal{Q})$ in the equation above by its bound in terms of $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$. □

The Rademacher bound of Theorem 4 and the PAC-Bayes bound of Theorem 7 convey a similar message: finding a good representation mapping (in terms of the empirical contrastive loss) guarantee to generalise well, on average, on the supervised tasks. An asset of the PAC-Bayesian bound lies in the fact that its exact value is easier to compute than the Rademacher one. Indeed, for a well chosen prior-posterior family, the complexity term $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ has a closed form solution, while computing $\mathcal{R}_U(\mathcal{F})$ involves a combinatorial

[2] Theorem 5 is given for the zero-one loss, but many works show that the same argument holds for any $[0,1]$-bounded loss (*e.g.*, Higgs and Shawe-Taylor, 2010).

complexity. From a algorithm design perspective, the fact that $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ varies with $\mathcal{Q}$ suggests a trade-off between accuracy and complexity to drive the learning process, while $\mathcal{R}_U(\mathcal{F})$ is constant for a given choice of class $\mathcal{F}$. We leverage on these assets to propose a bound-driven optimisation procedure for neural networks in Section 4.

3.3 Relaxing the iid assumption

An interesting byproduct of Arora et al. (2019)'s approach is that the proof of the main bound (Theorem 4) is modular: we mean that in the proof of Theorem 7, instead of plugging in Catoni's bound (Theorem 5), we can use any relevant bound. We therefore leverage the recent work of Alquier and Guedj (2018) who proved a PAC-Bayes generalisation bound which no longer needs to assume that data is iid, and even holds when the data-generating distribution is heavy-tailed. We can therefore cast our results onto the non-iid setting.

We believe removing the iid assumption is especially relevant for contrastive unsupervised learning, as we deal with triplets of data points governed by a relational causal link (similar and dissimilar examples). The limits of the iid assumption in contrastive learning has also been pointed out by Logeswaran and Lee (2018).

Alquier and Guedj (2018)'s framework generalises the Kullback-Leibler divergence in the PAC-Bayes bound with the class of f-divergences (see Csiszár and Shields, 2004, for an introduction). Given a convex function f such that $f(1) = 0$, the f-divergence between two probability distributions is given by

$$D_f(\mathcal{P}\|\mathcal{Q}) = \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} f\left(\frac{\mathcal{P}(h)}{\mathcal{Q}(h)}\right). \tag{10}$$

Theorem 8. *Given $p > 1, q = \frac{p}{p-1}$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta, \forall \mathcal{Q}$ over $\mathcal{F}$,*

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \leq \frac{1}{1-\tau}\left(\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \tau\right) + \frac{1}{1-\tau}\left(\frac{\mathcal{M}_q}{\delta}\right)^{\frac{1}{q}} \left(D_{\phi_p{-}1}(\mathcal{Q}\|\mathcal{P}) + 1\right)^{\frac{1}{p}}, \tag{11}$$

where $\mathcal{M}_q = \mathbb{E}_{\mathbf{f}\sim\mathcal{P}} \mathbb{E}_{U\sim\mathcal{U}^m}(|L_{\mathrm{un}}(\mathbf{f}) - \widehat{L}_{\mathrm{un}}(\mathbf{f})|^q)$ and $\phi_p(x) = x^p$.

The proof is a straightforward combination of aforementioned results, substituting Theorem 1 in Alquier and Guedj (2018) to Catoni's bound Theorem 5 in the proof of Theorem 7.

Up to our knowledge, Theorem 8 is the first generalisation bound for contrastive unsupervised representation learning which holds without the iid assumption, therefore extending the framework introduced by Arora et al. (2019) in a non-trivial and promising direction.

4 From Bounds to Algorithms

In this section, we propose a contrastive unsupervised representation learning algorithm derived from the PAC-Bayes bound stated in Theorem 7. The algorithm is obtained by optimising the weights of a neural network by minimising the right-hand side of (9). Our training method is inspired by the work of Dziugaite and Roy (2017), who optimise a PAC-Bayesian bound in a supervised classification framework, and show that it leads to non-vacuous bounds values and accurately detects overfitting.

4.1 Neural Network Optimisation

We consider a neural network architecture with N real-valued learning parameters. Let us denote $\mathbf{w} \in \mathbb{R}^N$ the concatenation into a single vector of all the weights, and $\mathbf{f_w} : X \to \mathbb{R}^d$ the output of the neural network whose output is a d-dimensional *representation* vector of its input. From now on, $\mathcal{F}_N = \{\mathbf{f_w} | \mathbf{w} \in \mathbb{R}^N\}$ is the set of all possible neural networks for the chosen architectures. We restrict the posterior and prior over $\mathcal{F}_N$ to be Gaussian distributions, that is

$$\mathcal{Q} \coloneqq \mathcal{N}(\boldsymbol{\mu}_{\mathcal{Q}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathcal{Q}})), \quad \mathcal{P} \coloneqq \mathcal{N}(\boldsymbol{\mu}_{\mathcal{P}}, \sigma^2_{\mathcal{P}} I),$$

where $\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\mu}_{\mathcal{P}} \in \mathbb{R}^N$, $\boldsymbol{\sigma}^2_{\mathcal{Q}} \in \mathbb{R}^N_+$, and $\sigma^2_{\mathcal{P}} \in \mathbb{R}_+$.

Given a fixed λ in Theorem 7, since τ is a constant value, minimising the upper bound is equivalent to minimising the following expression:

$$\lambda\, m\, \widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}. \tag{12}$$

Since $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ is still intractable (as it is expressed as the expectation with respect to the posterior distribution on predictors), we resort to an unbiased estimator; the weights parameter is sampled at each iteration of a gradient descent, according to

$$\mathbf{w} = \boldsymbol{\mu}_{\mathcal{Q}} + \boldsymbol{\sigma}_{\mathcal{Q}} * \boldsymbol{\epsilon}\,; \text{ with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I).$$

Therefore we optimise the posterior's parameters $\boldsymbol{\mu}_{\mathcal{Q}}$ and $\boldsymbol{\sigma}^2_{\mathcal{Q}}$. In addition, we optimise the prior variance $\sigma_{\mathcal{P}}$ in the same way as Dziugaite and Roy (2017, Section 3.1). That is, given fixed $b, c \in \mathbb{R}_+$, we consider the bound value for

$$\sigma^2_{\mathcal{P}} \in \{c \exp\left(-\tfrac{j}{b}\right) | j \in \mathbb{N}\}. \tag{13}$$

From the union bound argument, the obtained result is valid with probability $1-\delta$ by computing each bound with a confidence parameter $\delta_j := 1 - \frac{6}{\pi^2 j^2}$, where $j = b \ln \frac{c}{\sigma_{\mathcal{P}}^2}$.

Given δ, b, c, and λ, our final objective based on Theorem 7 is

$$\min_{\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\sigma}^2_{\mathcal{Q}}, \sigma^2_{\mathcal{P}}} \lambda m \widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + 2 \ln \left(b \ln \tfrac{c}{\sigma^2_{\mathcal{P}}} \right),$$

where

$$\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) = \frac{1}{2}\Big(\tfrac{\|\boldsymbol{\mu}_{\mathcal{Q}} - \boldsymbol{\mu}_{\mathcal{P}}\|_2^2}{\sigma^2_{\mathcal{P}}} - N + \tfrac{\|\boldsymbol{\sigma}^2_{\mathcal{Q}}\|_1}{\sigma^2_{\mathcal{P}}} + N \ln \sigma^2_{\mathcal{P}} - \sum_{i=1}^{N} \ln \boldsymbol{\sigma}^2_{\mathcal{Q},i} \Big).$$

4.2 Parameter Selection

We use the following two criteria for parameter selection: (i) the validation contrastive risk and (ii) the PAC-Bayes bound. On validation risk criterion, we select a model with the best hyper-parameters such that it achieves the lowest contrastive risk $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ on the validation data. We approximate $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ in a Monte Carlo fashion by sampling several $\mathbf{f}_{\mathbf{w}}$ from $\mathcal{Q}$.

Empirically, stochastic neural networks learnt by minimising the PAC-Bayes bound perform quite conservatively (Dziugaite and Roy, 2017). Therefore we also use a validation loss computed with a deterministic neural network (with weights taken as the mean vector of the posterior, rather than sampled from it).

The other criterion, the PAC-Bayes bound, does not use validation data; it only requires training data. We select a model with the best hyperparameters such that it minimises the following PAC-Bayes bound seen as a function of λ:

$$\min_{\lambda > 0} \left[\frac{1 - \exp\left(-\lambda \widehat{R}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + 2\ln(j) + \ln \frac{2\sqrt{m}}{\delta}}{m} \right)}{1 - \exp(-\lambda)} \right], \tag{14}$$

where all other parameters are fixed. This criterion is given by Corollary 6, where the term $\ln \frac{1}{m}$ is replaced by $2\ln(j) + \ln \frac{2\sqrt{m}}{\delta}$. The first summand comes from the union bound over the prior's variances–see Eq. (13). The second summand replaces $\frac{1}{\delta}$ by $\frac{2\sqrt{m}}{\delta}$, as Letarte et al. (2019, Theorem 3) showed that this suffices to make the bound valid uniformly for all $\lambda > 0$, which allows for minimising the bound over λ.

5 Numerical Experiments

Our experimental codes are publicly available at `https://github.com/nzw0301/pb-contrastive`.

We implemented all algorithms with `PyTorch` (Paszke et al., 2017).

5.1 Protocol

Datasets. We use `CIFAR-100` (Krizhevsky, 2009) image classification task, containing 60 000 images, equally distributed into 100 labels. We created train/validation/test splits of 47 500/2 500/10 000 images. We preprocess the images by normalising all pixels per channel based on the training data. We build the unsupervised contrastive learning dataset by considering each of the 100 label as a latent class, using a block size of 2 and a number of negative samples of 4 (see Appendix A for the extended theory for block samples and more than one negative samples).

We also use the `AUSLAN` (Kadous, 2002) dataset, that contains 95 labels, each one being a sign language's motion, and having 22 dimensional features. We split the dataset into 85 785/12 255/12 255 training/validation/test sets. As pre-processing, we normalise feature vectors per dimension based on the training data. The contrastive learning dataset then contains 95 latent classes. The block size and the number of negative samples are the same than for `CIFAR-100` setting. More details are given in Appendix C.1.

Neural networks architectures. For `CIFAR-100` experiments, we use a two hidden convolutional layers neural network (CNN). The two hidden layers are convolutions (kernel size of 5 and 64 channels) with the ReLU activation function, followed by max pooling (kernel size of 3 and stride of 2). The final layer is a fully connected linear layer (100 neurons) without activation function. For `AUSLAN` experiments, we used a fully connected one hidden layer network with the ReLU activation function. Both hidden and last layer have 50 neurons. More architecture details are given in Appendix C.2.

PAC-Bayes bound optimisation. We learn the neural network parameters by minimising the bound given by Theorem 7, using the strategy proposed in Section 4.1. We rely on the logistic loss given by Eq. (1). We fix the following PAC-Bayes bound's parameters: $b = 100, c = 0.1$, and $\delta = 0.05$. The prior variance is initialised at e^{-8}, and the prior mean parameters $\boldsymbol{\mu}_{\mathcal{P}}$ coincide with the random initialisation of the gradient descent.

We repeat the optimisation procedure with different combination of hyperparameters. Namely, the PAC-Bayes bound constant λ is chosen in $\{\frac{10^a}{m} | a = 2, 3, \ldots, 9\}$ for `CIFAR-100`, and in $\{\frac{10^a}{m} | a = 0, 1, \ldots, 5\}$ for `AUSLAN`. We also con-

sider as an hyperparameter the choice of the gradient descent optimiser, here between *RMSProp* (Tieleman and Hinton, 2012) and *Adam* (Kingma and Lei Ba, 2015). The learning rate is in $\{10^{-3}, 10^{-4}\}$. In all cases, 500 epochs are performed and the learning rate is divided by 10 at the 375^{th} epoch. To select the final model among the ones given by all these hyperparameter combinations, we experiment three model selection criteria based on approaches described in Section 4.2, as detailed below.

– *Stochastic validation* (`s-valid`). This metric is obtained by randomly sampling 10 set of network parameters according to the learnt posterior $\mathcal{Q}$, and averaging the corresponding empirical contrastive loss values computed on validation data. The same procedure is used to perform *early-stopping* during optimisation (we stop the learning process when the loss stops decreasing for 20 consecutive epochs).

– *Deterministic validation* (`det-valid`). This metric corresponds the empirical contrastive loss values computed on validation data of the deterministic network $\mathbf{f}^*$, which corresponds to the mean parameters of the posterior (*i.e.*, the *maximum-a-posteriori* network given by $\mathcal{Q}$). Early stopping is performed in the same way as for `s-valid`.

– *PAC-Bayes bound* (`PB`). The bound value of the learnt posterior $\mathcal{Q}$ is computed using Eq. (14). Note that since this method does not require validation data, we perform optimisation over the union of the validation data and the training data. We do not perform early stopping since the optimised objective function is directly the model selection metric.

Benchmark methods. We compare our results with two benchmarks, described below (more details are provided in Appendix C.3)

– *Prior contrastive unsupervised learning* (Arora et al., 2019). Following the original work, we minimise the empirical contrastive loss $\widehat{L}_{\text{un}}(\mathbf{f})$. Hyper-parameter selection is performed on the validation dataset as for `s-valid` and `det-valid` described above.

– *Supervised learning* (`supervised`). We also train the neural network in a supervised way, using the label information; Following the experiment of Arora et al. (2019), we add a prediction linear layer to our architectures (with 100 output neurons for `CIFAR-100`, and 95 output neurons for `AUSLAN`), and minimise the multiclass logistic loss function $\ell_{\log}(\mathbf{v}) := \log_2(1 + \sum_{i=1}^{|Y|} e^{-v_i})$. Once done, we drop the last (prediction) layer. Then, we use the remaining network to extract feature representation.

5.2 Experimental Results

Supervised classification. Table 1 contains supervised accuracies obtained from the representation learned with the two benchmark methods, as well as with our three model selection strategies on the PAC-Bayes learning algorithms. For each method, two types supervised predictor of are used: μ and μ-5 (as in Arora et al., 2019).[3] The μ classifier is obtained $\boldsymbol{\mu}_c$ that was the average vector of feature vectors $\hat{\mathbf{f}}_{\mathbf{w}}$ mapped from training data per supervised label, and μ-5 classifier had $\boldsymbol{\mu}_c$ that was average of 5 random training samples feature vectors. For μ-5, we used averaged evaluation scores over 5 times samplings on each experiment.

For the two datasets, we report three accuracy on the testing set, described below. Values are calculated by averaging over three repetitions of the whole experiments using different random seeds.

– *predictors-2 accuracy* (`AVG-2`). This is the empirical counterpart of Eq. (8), *i.e.*, given a test dataset $T := \{(\mathbf{z}_i, c_i)\}_{i=1}^{|T|}$ where $c_i \in \mathcal{C}$ is a latent class, we define $\texttt{AVG-2}(\mathbf{f}_{\mathbf{w}}) := 1 - \widehat{R}^{\mu}_{\text{sup}}(\mathbf{f}_{\mathbf{w}})$, given

$$\widehat{R}^{\mu}_{\text{sup}}(\mathbf{f}_{\mathbf{w}}) := \frac{C(C-1)}{2} \sum_{1 \le c^+ < c^- \le C} \widehat{R}_{T_{c^\pm}}(\hat{g}_{c^\pm} \circ \hat{\mathbf{f}}_{\mathbf{w}}), \quad (15)$$

where C is the number of latent classes (*e.g.*, C=100 for the `CIFAR-100` dataset), $\hat{\mathbf{f}}_{\mathbf{w}}$ is a feature map learnt from the training data, $\hat{g}_{c^\pm}(\hat{\mathbf{x}}) := (\hat{\mathbf{w}}_{c^+} - \hat{\mathbf{w}}_{c^-}) \cdot \hat{\mathbf{x}}$ is the predictor based on the centre of mass $\hat{\mathbf{w}}_{c^+}, \hat{\mathbf{w}}_{c^-}$ of the training data mapped features of classes c^+, c^-, and $\widehat{R}_{T_{c^\pm}}$ is the supervised risk on the dataset $T_{c^\pm} := \{(\mathbf{x}, 1) | (\mathbf{x}, c^+) \in T\} \cup \{(\mathbf{x}, -1) | (\mathbf{x}, c^-) \in T\}$:

$$\widehat{R}_{T_{c^\pm}}(\hat{g}_{c^\pm} \circ \hat{\mathbf{f}}_{\mathbf{w}}) := \frac{1}{|T_{c^\pm}|} \sum_{(\mathbf{x},y) \in T_{c^\pm}} r\big(\hat{g}_{c^\pm}(\hat{\mathbf{f}}_{\mathbf{w}}(\mathbf{x})), y\big).$$

– *Top-1 accuracy* (`TOP-1`). This is the accuracy on the multiclass labelled test data T. We predicted the label $\hat{y}_i = \operatorname{argmax}_y \boldsymbol{\mu}_y \cdot \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i)$ on the test data. Therefore,

$$\texttt{TOP-1}(\mathbf{f}_{\mathbf{w}}) := \frac{1}{|T|} \sum_{i=1}^{|T|} \mathbf{1}[y_i = \hat{y}_i].$$

– *Top-5 accuracy* (`TOP-5`). For each test instance $(\mathbf{x}_i, y_i) \in T$, let $\hat{Y}_i$ be the set of 5 labels having the

[3] Our neural network architecture on `CIFAR-100` differs from the one used in Arora et al. (2019). Their model is based on the deeper network `VGG-16` (Simonyan and Zisserman, 2015), which explains why our accuracies are lower than the one reported in Arora et al. (2019).

	supervised		Arora et al. (2019)		PAC-Bayes based methods					
					s-valid		det-valid		PB	
	μ	μ-5	μ	μ-5	μ	μ-5	μ	μ-5	μ	μ-5
CIFAR-100										
AVG-2	91.4	87.6	89.4	85.5	87.5	83.4	**90.0**	**87.1**	75.2	70.9
TOP-1	25.3	17.2	**22.5**	15.0	16.9	12.2	21.4	**16.2**	6.8	5.5
TOP-5	57.8	46.1	52.9	42.7	46.2	37.0	**54.0**	**45.4**	23.0	19.6
AUSLAN										
AVG-2	79.1	74.0	76.1	68.5	76.2	69.1	**76.4**	**69.7**	74.6	67.5
TOP-1	12.2	7.5	23.2	11.4	24.9	11.6	**26.0**	**12.4**	19.6	9.9
TOP-5	35.5	24.8	45.4	26.0	45.5	26.9	**46.6**	**27.8**	40.8	24.3

Table 1: Supervised tasks results. `supervised` was trained on the labelled training data, the others were trained on the contrastive training data. For `supervised`, Arora et al. (2019), `s-valid`, and `det-valid`, hyper-parameters were selected by using the validation loss. `PB` hyper-parameters were selected by the PAC-Bayes bound. The best scores are in bold among contrastive learning algorithms.

highest inner products $\boldsymbol{\mu}_y \cdot \mathbf{f}(\mathbf{x}_i)$. Then,

$$\texttt{TOP-5}(\mathbf{f}_\mathbf{w}) := \frac{1}{|T|}\sum_{i=1}^{|T|} \mathbf{1}[y_i \in \hat{Y}_i]\,.$$

Note that the `TOP-1` and `TOP-5` metrics are not supported by theoretical results, in the present paper or the work of Arora et al. (2019). Nevertheless, we report those as an empirical hint of how representations are learned by our contrastive unsupervised representation learning algorithm.

We observe that `det-valid` algorithm achieves the highest scores of μ and μ-5 classifiers among contrastive learning algorithms except for `TOP-1` of `CIFAR-100`.

PAC-Bayesian generalisation bounds. Table 2 shows the PAC-Bayes bound values obtained from Eq. (14). The bounds were calculated by using the same models used in Table 1. We also reported a training risk $\widehat{R}_{\text{un}}(\mathbf{f}^*)$ and test risk $R_{\text{un}}(\mathbf{f}^*)$ that we calculated by using only the mean parameter of the posterior as for neural network's weight. The rows of $\widehat{\lambda}$ indicated the optimised λ values that minimised Eq. (14), and thus that correspond to the reported PAC-Bayes bounds. Let us stress that all reported bounds values are non-vacuous.

The generalisation bounds obtained with the `PB` model selection criteria are naturally the tightest. For this method, the gap between the empirical risk $\widehat{R}_{\text{un}}(\mathcal{Q})$ and the test risk $\widehat{R}_{\text{un}}(\mathcal{Q})$ is remarkably consistently small. This highlights that the PAC-Bayesian bound minimisation is not prone to overfitting. On the downside, this behaviour seems to promote "conservative" solutions, which in turns gives lower supervised accuracy compared to methods relying on a validation set (as reported in Table 1).

	s-valid	det-valid	PB
CIFAR-100			
$\widehat{R}_{\text{un}}(\mathbf{f}^*)$	0.154	0.131	0.308
$R_{\text{un}}(\mathbf{f}^*)$	0.189	0.167	0.315
$\widehat{R}_{\text{un}}(\mathcal{Q})$	0.179	0.170	0.324
$R_{\text{un}}(\mathcal{Q})$	0.207	0.197	0.330
Bound	0.735	0.721	0.455
KL	31 707	30 894	1 342
$\lambda \times m$	10^5	10^5	10^4
$\widehat{\lambda} \times m$	120 915	120 470	27 954
AUSLAN			
$\widehat{R}_{\text{un}}(\mathbf{f}^*)$	0.001	0.000	0.032
$R_{\text{un}}(\mathbf{f}^*)$	0.006	0.005	0.091
$\widehat{R}_{\text{un}}(\mathcal{Q})$	0.001	0.001	0.005
$R_{\text{un}}(\mathcal{Q})$	0.007	0.008	0.020
Bound	0.069	0.068	0.034
KL	5 064	4 996	1 369
$\lambda \times m$	10^5	10^5	10^4
$\widehat{\lambda} \times m$	366 842	368 916	196 729

Table 2: Contrastive unsupervised PAC-Bayes bounds of the models used in Table 1.

6 Conclusion

We extended the framework introduced by Arora et al. (2019), by adopting a PAC-Bayes approach to contrastive unsupervised representation learning. This allows in particular to (i) derive a new algorithm, by minimising the bound (ii) remove the iid assumption. While supported by novel generalisation bounds, our approach is also validated on numerical experiments are the bound yields non-trivial (non-vacuous) values.

References

P. Alquier and B. Guedj. Simpler PAC-Bayesian Bounds for Hostile Data. *Machine Learning*, 107 (5):887–902, 2018.

S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *ICML*, pages 5628–5637, 2019.

Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018.

O. Catoni. A PAC-Bayesian Approach to Adaptive Classification. *preprint*, 840, 2003.

O. Catoni. *Statistical Learning Theory and Stochastic Optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001*. Springer, 2004.

O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *IMS Lecture Notes - Monograph Series*. Institute of Mathematical Statistics, 2007.

I. Csiszár and P. C. Shields. Information Theory and Statistics: A Tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4): 417–528, 2004.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

G. K. Dziugaite and D. M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *UAI*, 2017.

B. Guedj. A Primer on PAC-Bayesian Learning. *arXiv preprint arXiv:1901.05353v3*, 2019.

M. Higgs and J. Shawe-Taylor. A PAC-Bayes Bound for Tailored Density Estimation. In *ALT*, 2010.

M. W. Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, University of New South Wales, 2002. URL https://archive.ics.uci.edu/ml/datasets/Australian+Sign+Language+signs+(High+Quality).

D. P. Kingma and J. Lei Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

G. Letarte, P. Germain, B. Guedj, and F. Laviolette. Dichotomize and generalize: Pac-bayesian binary activated deep neural networks. *arXiv preprint arXiv:1905.10259*, 2019.

L. Logeswaran and H. Lee. An Efficient Framework for Learning Sentence Representations. In *ICLR*, 2018.

D. A. McAllester. Some PAC-Bayesian Theorems. In *COLT*, pages 230–234, 1998.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, 2013.

M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, O. Srl, and A. Lerer. Automatic Differentiation in PyTorch. In *NeurIPS Workshop*, 2017.

J. Shawe-Taylor and R. C. Williamson. A PAC Analysis of a Bayesian Estimator. In *COLT*, pages 2–9, 1997.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012. URL http://www.cs.toronto.edu/~tijmen/csc321/slides/lectu

L. G. Valiant. A theory of the learnable. In *STOC*, pages 436–445. ACM, 1984.

R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *ECCV*, 2016.

A Extended PAC-Bayes Bounds

Arora et al. (2019) show two extended generalisation error bounds based on Theorem 4. We also show each PAC-Bayesian counterpart of their extended bounds for Theorem 7.

A.1 Block bound

The first extension is to use block pairs for positive and negative samples to make the bound tighter. We also derive a tighter PAC-Bayes bound on the same setting.

Let b be the size of blocks. We change the data generation process; Given $(c^+, c^-) \sim \rho^2$, we sample $(\mathbf{x}, \{\mathbf{x}_j^+\}_{j=1}^b) \sim \mathcal{D}_{c^+}^{b+1}$ and $\{\mathbf{x}_j^-\}_{j=1}^b \sim \mathcal{D}_{c^-}^b$. Given block pairs, unsupervised block loss is defined as

$$L_{\mathrm{un}}^{block}(\mathbf{f}) = \mathbb{E}\left\{\ell\left[f(\mathbf{x}) \cdot \left(\frac{\sum_{i=1}^b \mathbf{f}(\mathbf{x}_i^+)}{b} - \frac{\sum_{i=1}^b \mathbf{f}(\mathbf{x}_i^-)}{b}\right)\right]\right\}. \tag{16}$$

This block loss $L_{\mathrm{un}}^{block}(\mathbf{f})$ lower bounds $L_{\mathrm{un}}(\mathbf{f})$ (Arora et al., 2019, Proposition 6.2): $\forall \mathbf{f} \in \mathcal{F}, L_{\mathrm{un}}^{block}(\mathbf{f}) \leq L_{\mathrm{un}}(\mathbf{f})$. Based on this lower bound, when we define $L_{\mathrm{un}}^{block}(\mathcal{Q}) = \mathbb{E}_{\mathbf{f}\sim\mathcal{Q}} L_{\mathrm{un}}^{block}(\mathbf{f})$, we obtain the following lower bound of the unsupervised risk $L_{\mathrm{un}}(\mathcal{Q})$ for all $\mathcal{Q}$ over $\mathcal{F}$ by taking the expected value according to $\mathcal{Q}$,

$$L_{\mathrm{un}}^{block}(\mathcal{Q}) \leq L_{\mathrm{un}}(\mathcal{Q}).$$

Therefore we derive the tighter block bound by combining the previous lower bound and Theorem 7.

Proposition 9. *$\forall \mathcal{Q}$ over $\mathcal{F}$,*

$$L_{\mathrm{sup}}^{\mu}(\mathcal{Q}) \leq \frac{1}{1-\tau}\left(B_\ell \frac{1 - \exp\left(-\frac{\lambda}{B_\ell}\widehat{L}_{\mathrm{un}}^{block}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp(-\lambda)} - \tau\right). \tag{17}$$

A.2 k-negative samples bound

The second extension is to use k negative samples in their framework as a general setting. Following Arora et al. (2019), we consider the data generation process with k negative samples per each pair. Let $\mathcal{U}$ be the process that generates an unlabelled sample $\mathbf{z} = (\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^k)$ according to the following scheme:

1. Draw $k+1$ latent classes $(c^+, \{c_i^-\}_{i=1}^k) \sim \rho^{k+1}$;
2. Draw two similar samples $(\mathbf{x}, \mathbf{x}^+) \sim (\mathcal{D}_{c^+})^2$;
3. Draw k negative samples $\{\mathbf{x}_i^- \sim \mathcal{D}_{c_i^-} \mid i = 1, \ldots, k\}$.

We extend loss functions for a vector of size k. We use two convex loss functions:

$$\ell_{\log}(\mathbf{v}) \coloneqq \log_2(1 + \sum_{i=1}^k e^{-v_i}), \qquad \textit{(logistic loss)} \tag{18}$$

$$\ell_{\mathrm{hinge}}(\mathbf{v}) \coloneqq \max[0, 1 + \max_i(-v_i)], \qquad \textit{(hinge loss)} \tag{19}$$

Then we define unsupervised contrastive loss and empirical contrastive loss with k negative samples;

$$L_{\mathrm{un}}(\mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{U}} \ell\left(\left\{\mathbf{f}(\mathbf{x}) \cdot \left[\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}_i^-)\right]\right\}_{i=1}^k\right), \tag{20}$$

$$\widehat{L}_{\mathrm{un}}(\mathbf{f}) \coloneqq \frac{1}{m}\sum_{i=1}^m \ell\left(\left\{\mathbf{f}(\mathbf{x}_i) \cdot \left[\mathbf{f}(\mathbf{x}_i^+) - \mathbf{f}(\mathbf{x}_{ij}^-)\right]\right\}_{j=1}^k\right). \tag{21}$$

We analyse a mean classifier as with $k = 1$ scenario. Let $\mathcal{T}$ be the set of supervised classes whose size is $k+1$, let $\mathcal{D}$ be the distribution over $\mathcal{T}$, and let $\mathcal{D}_{\mathcal{T}}$ be the distribution over class in $\mathcal{T}$. The supervised average loss of mean classifier with k negative samples is defined as

$$L_{\mathrm{sup}}^{\mu}(\mathbf{f}) = \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}} L_{\mathrm{sup}}^{\mu}(\mathcal{T}, \mathbf{f}) = \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\mathcal{D}_{\mathcal{T}}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_c} [\ell(\{\mathbf{f}(\mathbf{x}) \cdot (\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c'})\}_{c'\neq c})]. \tag{22}$$

To introduce the counterpart of Lemma 3 for k negative samples, we introduce notations related to the extended class collision. Let $I^+(c_1^-, \dots, c_k^-) = \{i \in [1, \dots, k] \mid c_j^- = c^+\}$ be a set of negative sample indices such that c_j is the same to c^+. Let $\tau_k = P(I^+ \neq \phi)$ be the class collision probability, and let Q be a distinct latent class set of $c^+, c_1^-, \dots, c_k^-$ sampled from ρ^{k+1}.

The following Lemma 10 shows the upper bound of supervised average loss with $k \geq 1$ by the unsupervised contrastive loss.

Lemma 10. *(Arora et al., 2019, Eq. 26),* $\forall \mathbf{f} \in \mathcal{F}$,[4]

$$(1 - \tau_k) \mathop{\mathbb{E}}_{\mathcal{T} \sim \mathcal{D}} \frac{p^+_{\min}(\mathcal{T})}{p_{\max}(\mathcal{T})} L^{\mu}_{\sup}(\mathcal{T}, \mathbf{f}) \leq L_{\mathrm{un}}(\mathbf{f}) - \tau_k \mathop{\mathbb{E}}_{c^+, \{c_i^-\}_{i=1}^k \sim \rho^{k+1}} [\ell(\mathbf{0}_{|I^+|}) \mid I^+ \neq \phi]. \tag{23}$$

where $\mathbf{0}_{|I^+|}$ *is zero vector of size* $|I^+|$, $p_{\max}(\mathcal{T}) = \max_c \mathcal{D}_{\mathcal{T}}(c)$, *and* $p^+_{\min}(\mathcal{T}) = \min_{c \in \mathcal{T}} p_{c^+, \{c_i^-\}_{i=1}^k \sim \rho^{k+1}}(c^+ = c \mid \mathcal{T} = Q, I = \phi)$.

Let us denote Q-weighted loss functions of contrastive learning with k negative samples:

$$L_{\mathrm{un}}(\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{\mathbf{f} \sim \mathcal{Q}} L_{\mathrm{un}}(\mathbf{f}), \tag{24}$$

$$\widehat{L}_{\mathrm{un}}(\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{\mathbf{f} \sim \mathcal{Q}} \widehat{L}_{\mathrm{un}}(\mathbf{f}), \tag{25}$$

$$L^{\mu}_{\sup}(\mathcal{T}, \mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{\mathbf{f} \sim \mathcal{Q}} L^{\mu}_{\sup}(\mathcal{T}, \mathbf{f}). \tag{26}$$

We derive the following Theorem 11 based on Lemma 10 to extend Theorem 7 for $k \geq 1$.

Theorem 11. *Let* $B \in \mathbb{R}_+$ *such that* $\|\mathbf{f}(\cdot)\| \leq B$ *for all* $\mathbf{f} \in \mathcal{F}$. *Given* $k \in \mathbb{N}, \lambda > 0$ *and a prior* $\mathcal{P}$ *over* $\mathcal{F}$, *with probability at least* $1 - \delta$ *over training samples* $U \sim \mathcal{U}^m$, $\forall \mathcal{Q}$ *over* $\mathcal{F}$,

$$(1 - \tau_k) \mathop{\mathbb{E}}_{\mathcal{T} \sim \mathcal{D}} \frac{p^+_{\min}(\mathcal{T})}{p_{\max}(\mathcal{T})} L^{\mu}_{\sup}(\mathcal{T}, \mathcal{Q}) \leq B_\ell \frac{1 - \exp\left(-\frac{\lambda}{B_\ell} \widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q} \| \mathcal{P}) + \ln \frac{1}{\delta}}{m}\right)}{1 - \exp(-\lambda)} - \tau_k \mathop{\mathbb{E}}_{c^+, \{c_i^-\}_{i=1}^k \sim \rho^{k+1}} [\ell(\{0\}_{|I^+|}) \mid I^+ \neq \phi].$$

with $B_\ell \coloneqq \log_2(1 + ke^{2B})$ *for the logistic loss, or* $B_\ell \coloneqq 1 + 2B$ *for the hinge loss.*

Proof. We follow the similar steps to the proof of Theorem 7. Since $\|\mathbf{f}(\cdot)\| \leq B$, we have $\forall \mathbf{x}, \mathbf{x}^+, \mathbf{x}^- \in X$[3]:

$$-2B \leq \mathbf{f}(\mathbf{x}) \cdot [\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)] \leq 2B\,.$$

Given the number of negative samples k, from the loss functions' definition, we can obtain the lower bound and upper bound explicitly.

$$\log_2(1 + ke^{-2B}) \leq \ell_{\log}(\mathbf{v}) \leq \log_2(1 + ke^{2B})\,, \tag{27}$$

$$0 \leq \ell_{\mathrm{hinge}}(\mathbf{v}) \leq 1 + 2B\,, \tag{28}$$

Thus $B_{\ell_{\log}} \coloneqq \log_2(1 + ke^{2B})$ and $B_{\ell_{\mathrm{hinge}}} \coloneqq 1 + 2B$. Therefore we can bound the $L_{\mathrm{un}}(\mathcal{Q})$ by using the same inequality in the proof of Theorem 7: With probability at least $1 - \delta$,

$$\frac{1}{B_\ell} L_{\mathrm{un}}(\mathcal{Q}) \leq \frac{1 - \exp\left(-\lambda \frac{1}{B_\ell} \widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q} \| \mathcal{P}) + \ln \frac{1}{\delta}}{m}\right)}{1 - \exp(-\lambda)}\,.$$

Also since Lemma 10 is true for all $\mathbf{f} \in \mathcal{F}$, we take expected value according to $\mathcal{Q}$;

$$(1 - \tau_k) \mathop{\mathbb{E}}_{\mathcal{T} \sim \mathcal{D}} \frac{p^+_{\min}(\mathcal{T})}{p_{\max}(\mathcal{T})} L^{\mu}_{\sup}(\mathcal{T}, \mathcal{Q}) \leq L_{\mathrm{un}}(\mathcal{Q}) - \tau_k \mathop{\mathbb{E}}_{c^+, \{c_i^-\}_{i=1}^k \sim \rho^{k+1}} [\ell(\{0\}_{|I^+|}) \mid I^+ \neq \phi]. \tag{29}$$

The result is obtained by replacing $L_{\mathrm{un}}(\mathcal{Q})$ in above inequality by its bound in terms of $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$. ∎

[4] In the original paper, it is shown for $\widehat{\mathbf{f}}$, but actually it is valid $\forall \mathbf{f} \in \mathcal{F}$.

B Contrastive Zero-one Risk with k-Negative Samples

We extend the zero-one risk to k negative sampling setting; Let $\mathbf{z} = (\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \ldots, \mathbf{x}_k^-)$, then

$$r_k(\mathbf{z}) = \frac{1}{k}\sum_{i=1}^{k} r(\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}_i^-), \mathbf{f}(\mathbf{x})). \tag{30}$$

We use this zero-one risk to compute $\widehat{R}(\mathcal{Q})$ used in Eq. (14).

C Experiment details

C.1 `AUSLAN` dataset

We used `AUSLAN` time-series dataset instead of `Wiki-3029` used in Arora et al. (2019), which contains 3 029 classes' sentences sampled from Wikipedia. This is because Arora et al. (2019) used recurrent neural networks on this dataset, but PAC-Bayes theory with recurrent neural networks on word sequences dataset is not trivial due to its time-dependent predictor and data sparsity, so it is not out of scope in this paper. Therefore we selected this dataset as a simpler and similar dataset.

`AUSLAN` originally contains 27 time-series samples per class. Each sample has different lengths, whose the maximum is 136 and the minimum is 45, and each time step is represented by a feature vector whose dimensionality is 22. We treated each feature vector as each input sample in our experiment. In addition, we only used the first 45 time steps to unify the number of samples per class. We separated original 27 times-series into 21/3/3 training/validation/test sets. As a result, we obtained 89 775/12 825/12 825/ training/validation/test datasets. We used these datasets as supervised datasets.

From the previous supervised dataset, we created contrastive data by the following ways:

1. Create positive pair $(\mathbf{x}_t, \{\mathbf{x}_j\}_{j=t+1}^{t+2})$, $t = 1, \ldots 43$ per original sample.

2. Sample c^- randomly, then use a random time t' to create a negative block $\{\mathbf{x}_j^-\}_{j=t'}^{t'+1}$,

where the block size is 2. As a result, we obtained 85 785/12 255/12 255 training/validation/test contrastive datasets.

C.2 Network architectures and initialisation parameters

`CIFAR-100` experiments. For all convolution layers the number of channels was 64, the kernel size was 5, the stride of the convolution was 1, zero-padding was 1, and the dilation was 1. The convolutional layers' parameters were initialised as zero-mean truncated Gaussian distribution whose σ was 0.1. For all max-pooling layers, the kernel size was 3, the stride of the window was 2, and the dilation was 1. For the linear layer, the number of units was 100. The linear layers' parameters were also initialised as zero-mean truncated Gaussian distribution whose σ was $1/800$. For all convolutional layers and linear layers biases were initialised as 0.

`AUSLAN` experiments. we used a fully connected one hidden layer's network with ReLU activation function. Both hidden and last layer have 50 neurons. The hidden layers' parameters were initialised as zero-mean truncated Gaussian distribution whose σ was $1/11$, and the output layer's parameters were initialised as zero-mean truncated Gaussian distribution whose σ was $1/25$

C.3 Benchmark methods

Comparison with Arora et al. (2019). We optimised the model by using a stochastic gradient descent algorithm with 100 mini-batches and 500 epochs. We searched the best learning rate in $\{10^{-3}, 10^{-4}\}$ and optimiser algorithm in stochastic gradient descent (SGD) with momentum 0.9, *RMSProp*, and *Adam*. We also performed early-stopping and updated the learning rate by the same as the PAC-Bayes setting.

Supervised learning. The additional linear layers' parameters were initialised as zero-mean truncated Gaussian distribution with $\sigma = 1/50$ and a bias was initialised as 0. The loss function was the multi-class logistic loss. We did the same way to find the best hyper-parameters, learning rate and optimiser, and to perform early-stopping. Optimisation methods and procedures were also same to the non-PAC-Bayesian contrastive learning setting.