Noise Contrastive Learning

Oscar Clivio
Department of Statistics
University of Oxford
oscar.clivio@spc.ox.ac.uk

Michael Komodromos
Department of Mathematics
Imperial College London
michael.komodromos19@ic.ac.uk

Thomas Matcham
Department of Mathematics
Imperial College London
thomas.matcham14@ic.ac.uk

Jake Topping
Department of Statistics
University of Oxford
james.topping@keble.ox.ac.uk

Abstract

We present an overview of Noise Contrastive Learning and outline the use cases in both a Bayesian and Frequentist setting. We demonstrate the utility of Noise Contrastive Learning through a series of experiments where we use it to estimate model parameters, retrieve posterior distributions, estimate the pdf of mixture models and evaluate Bayes factors for model comparison.

1 Introduction

Noise Contrastive Learning (NCL) is originally framed as a technique to estimate model parameters by learning to discriminate between data and some artificially generated noise (Gutmann and Hyvärinen [2010]). For example, in many settings, we are bound to estimate a normalizing constant of some unnormalized distribution, often jointly with the parameters of this distribution. It can be difficult to estimate, particularly when the data is high-dimensional (Gutmann and Hyvärinen [2010]). Often, we resort to Monte Carlo methods. However, classical Monte Carlo methods such as Importance Sampling (IS) and "Weighted Harmonic Mean" (Gelfand and Dey [1994]) poorly estimate the normalizing constant and exhibit high variance (Liu et al. [2015]). The indirect approach of NCL circumvents these difficulties.

Since the introduction of NCL, the concept has also been extended to more general tasks where classical methods are not applicable. For example, in a Bayesian setting where the marginal likelihood is intractable, Markov Chain Monte Carlo (MCMC) methods can be used to draw samples from the posterior distribution. If however, the likelihood is intractable, a setting referred to as *doubly intractable* (Murray et al. [2006]), likelihood-free inference can be used to estimate the posterior distribution. Some likelihood-free methods (Thomas et al. [2020], Hermans et al. [2020]) draw on NCL and use it to estimate a ratio between the likelihood and the marginal likelihood even if these two quantities are cannot be evaluated.

The remainder of our report is organised as follows. In Section 2, we introduce different approaches related to NCL. In Section 3, we reproduce some results from Gutmann and Hyvärinen [2010] and Hermans et al. [2020], and we demonstrate the power of NCL on other applications. In section 4, we conclude our report and highlight other use cases.

2 Noise Contrastive Learning

NCL is amenable to problems in both Frequenstist and Bayesian settings. In a Frequenist setting, NCL can be used for parameter estimation, as well as estimation of a distribution's normalising

constant. In a Bayesian setting, NCL can be used to estimate the posterior density in the absence of a tractable likelihood function.

2.1 Frequentist setting

Consider, p_d, the true distribution of some data X, modelled by a parameterised family of functions $\{p_m(\,\cdot\,;\beta)\}_\beta$ such that $p_d = p_m(\,\cdot\,;\beta^*)$. If p_d is known up to a normalising constant then we can express p_d as

$$p_d = c\, p_m^o(\,\cdot\,;\alpha^*) = p_m(\,\cdot\,;c,a^*) \tag{1}$$

where c is the normalising constant and $p_m^o(\,\cdot\,;\alpha)$ is an unnormalised density that can be evaluated.

Consider a dataset $X = (x_1, \ldots, x_N)$ and let p_n denote a distribution which we can evaluate and sample from. Using p_n we generate data $X' = (x'_1, ..., x'_N)$, thereby obtaining the joint dataset $Z = (z_1, ..., z_{2N})$ where every z_i is sampled from either p_m or from p_n. Let y_i be an auxiliary variable denoting whether z_i has been sampled from p_m or p_n, giving

$$p(z|y=1;c,\alpha) = p_m(z;c,\alpha), \qquad p(z|y=0;c,\alpha) = p_n(z). \tag{2}$$

Assuming the class labels are independent and supported by an equal number of observations, we have

$$p(y=1|z;c,\alpha) = d(z;c,\alpha), \qquad p(y=0|z;c,\alpha) = 1 - d(z;c,\alpha), \tag{3}$$

where

$$d(z;c,\alpha) = \frac{p_m(z;c,\alpha)}{p_m(z;c,\alpha) + p_n(z)}. \tag{4}$$

Using eq. (3) we can obtain an estimate for c and α by optimising the log-likelihood of $Y = (y_1, ..., y_{2N})$,

$$\begin{aligned}\ell(Y|Z;c,\alpha) &= \sum_{i=1,\ldots,2N} \Big[y_i \log p(y_i=1|z_i;c,\alpha) + (1-y_i)\log p(y_i=0|z_i;c,\alpha) \Big] \\ &= \sum_{i=1,\ldots,N} \Big[\log d(x_i;c,\alpha) + \log(1 - d(x'_i;c,\alpha)) \Big].\end{aligned} \tag{5}$$

The choice of p_n is crucial for estimating parameters. Notably if the support of p_n includes the support of p_d (and if other conditions inspired from maximum likelihood estimation are held) then the estimated parameters $(\hat{c}, \hat{\alpha})$ converge in probability to the optimal parameters (c^*, α^*) such that $p_d(.) = p_m(.; c^*, \alpha^*)$. From a practical perspective, p_n should be easy to sample from and compute (Gutmann and Hyvärinen [2010]). We note that if α has already been estimated then we can use NCL as an alternative to Monte-Carlo methods to estimate the normalizing constant (Liu et al. [2015]).

2.2 Bayesian setting

Under a Bayesian framework we are interested in inferring the posterior distribution

$$p(\theta|x) = p(\theta) r(x,\theta), \qquad r(x,\theta) = \frac{p(x|\theta)}{p(x)} \tag{6}$$

where the marginal likelihood, $p(x)$, is given by

$$p(x) = \int_\Theta p(\theta) p(x|\theta) d\theta \tag{7}$$

for some $x \in \mathcal{X}$ and $\theta \in \Theta$. We recall that neither $p(x|\theta)$ nor $p(x)$ can be evaluated.

2.2.1 Inferring the posterior distribution

Thomas et al. and Hermans et al. estimate the posterior distribution by first obtaining an estimate for the discriminator

$$d^*(x,\theta) = \frac{p(x|\theta)}{p(x|\theta) + p(x)} = \frac{p(x,\theta)}{p(x,\theta) + p(x)p(\theta)} \tag{8}$$

denoted as $\widehat{d}(x, \theta)$. Using estimate $\widehat{d}$ we have,

$$\hat{r}(x, \theta) = \frac{\hat{d}(x, \theta)}{1 - \hat{d}(x, \theta)}. \tag{9}$$

It follows, an estimate for the posterior distribution is given by

$$\widehat{p}(\theta|x) = p(\theta)\widehat{r}(x, \theta). \tag{10}$$

Hermans et al. [2020] estimate $\widehat{r}(x, \theta)$ by assuming the pair (x, θ) have been sampled from either $p(x, \theta)$ or $p(x)p(\theta)$, where in the former case (x, θ) are sampled dependently, i.e.

$$\theta \sim p(\theta), \qquad x \sim p(x|\theta) \tag{11}$$

whereas in the later case we sample

$$\theta' \sim p(\theta), \qquad x' = x \tag{12}$$

in effect making x' independent of θ' as x' has been sampled from $p(x|\theta)$ using θ and not θ'. This can be justified using Equation (7), as the marginal $p(x)$ is obtained by integrating out θ (Thomas et al. [2020]). As before, we assign labels $y_i \in \{0, 1\}$ to observations (x_i, θ_i) taking value 1 if (x_i, θ_i) were sampled from $p(x, \theta)$ and 0 otherwise. We then optimise the log-likelihood of $Y = (y_1, \ldots, y_{2N})$

$$\ell(Y|X, \Theta; \omega) = \sum_{i=1,\ldots,N} \Big[\log \widehat{d}(x'_i, \theta'_i; \omega) + \log(1 - \widehat{d}(x'_i, \theta'_i; \omega)) \Big]. \tag{13}$$

Notably, eq. (8) can also be thought of as a discriminator distinguishing samples from $p(x|\theta)$ and $p(x)$. However, fitting an estimate $\widehat{d}_\theta(x; \omega)$ that can discriminate between $p(x|\theta)$ and $p(x)$ for every θ requires fitting a separate classifier for every θ (Thomas et al. [2020]). On the other hand, sampling data-parameter pairs (x, θ) allows for amortised computations with respect to θ. Thereby, in Hermans et al. [2020] $\hat{d}(x, \theta; \omega)$ is given by a neural network, whereas Thomas et al. [2020] parameterise and fit $\hat{d}_\theta(x; \omega)$ via logistic regression. Importantly, both Hermans et al. [2020] and Thomas et al. [2020] show that their respective classifiers converge to $d^*(x, \theta)$, the optimal discriminator.

2.2.2 Estimating the acceptance ratio in MCMC

Hermans et al. [2020] use the estimate of $r(x, \theta)$ to circumvent the intractability of likelihoods in MCMC inference. In the Metropolis-Hastings algorithm, the acceptance probability, ρ, of the transition from θ_t to θ' is

$$\rho = \min\left(1, \frac{p(\theta')}{p(\theta_t)} \frac{p(x|\theta')}{p(x|\theta_t)} \frac{q(\theta'|\theta_t)}{q(\theta_t|\theta')}\right) \tag{14}$$

where $q(\cdot|\theta)$ is the Markov Kernel. As

$$\frac{p(x|\theta')}{p(x|\theta_t)} = \frac{r(x, \theta')}{r(x, \theta_t)}, \tag{15}$$

the acceptance probability can be rewritten as

$$\rho = \min\left(1, \frac{p(\theta')}{p(\theta_t)} \frac{r(x, \theta')}{r(x, \theta_t)} \frac{q(\theta'|\theta_t)}{q(\theta_t|\theta')}\right). \tag{16}$$

Hence, we can provide an estimate $\widehat{\rho}$ using $\widehat{r}$.

2.3 Comparison of Bayesian and Frequentist Methods

We note the Bayesian approach differs from the Frequentist approach as the discriminant function $\hat{d}$ in eq. (4) and eq. (8) are not equivalent. For instance, in the Bayesian setting d^* is estimated, whereas, in the Frequentist approach the discriminator is equal to $\frac{p_m}{p_m + p_n}$.

Further, in the Frequentist approach, the unnormalised distribution p_m^o and the noise distribution p_n are assumed to be tractable, whereas in the Bayesian approach neither $p(x|\theta)$ nor $p(x)$ are tractable, but we can be sampled from. Thereby, we cannot directly infer $p(x|\theta)$ from $d^*(x, \theta)$ as the intractability of $p(x)$ remains, and vice-versa.

2.4 Estimating Bayes Factors

We now consider an approach at the intersection of Frequentist and Bayesian approaches and which consists in estimating Bayes factors.

Consider data $x_{obs} = (x^1_{obs}, ..., x^N_{obs}) \in \mathcal{X}^N$ generated from one of

$$\begin{aligned} \mathcal{M}_1 &: \; x|\theta \sim p_1(.|\theta), \qquad \theta \sim p(\theta), \\ \mathcal{M}_2 &: \; x|\theta \sim p_2(.|\theta), \qquad \theta \sim p(\theta). \end{aligned} \tag{17}$$

Noticeably, the likelihoods have different forms but the parameter priors are the same. Typically, we decide which model best represents the data by evaluating the Bayes factor K, defined as

$$K = \frac{p_1(x)}{p_2(x)}. \tag{18}$$

Thereby $K > 1$ indicates evidence supporting $\mathcal{M}_1$ and $K < 1$ evidence supporting $\mathcal{M}_2$.

When using NCL, we assume $p_i(\cdot|\theta), i = 1, 2$ can be evaluated. Hence, we generate N samples $\theta^i_1, ..., \theta^i_N$ from each parameter posterior $p_i(\theta|x_{obs})$ using exact or approximate inference. We assign labels $y_j \in \{0, 1\}$ to observations θ_j taking value 1 if θ_j was sampled from $p_1(\theta|x)$ and 0 otherwise. We note, as parameter priors in $\mathcal{M}_1$ and $\mathcal{M}_2$ are identical,

$$p(y = 1|\theta, x_{obs}) = \frac{p_1(\theta|x_{obs})}{p_1(\theta|x_{obs}) + p_2(\theta|x_{obs})} = \frac{p_1(x_{obs}|\theta)}{p_1(x_{obs}|\theta) + Kp_2(x_{obs}|\theta)}. \tag{19}$$

We therefore estimate K by optimizing the log-likelihood of $Y = (y^1_1, \ldots, y^1_N, y^2_1, \ldots, y^2_N)$,

$$\begin{aligned} \ell(Y|\Theta, x_{obs}; K) &= \sum_{i=1,\ldots,N} \Big[\log p(y^1_i = 1|\theta^1_i, x_{obs}; K) + \log p(y^1_i = 0|\theta^2_i, x_{obs}; K) \Big] \\ &= - \sum_{i=1,\ldots,N} \Big[\log \Big(1 + e^{l_2(\theta^1_i) - l_1(\theta^1_i) + \log K} \Big) + \log \Big(1 + e^{l_1(\theta^2_i) - l_2(\theta^2_i) - \log K} \Big) \Big], \end{aligned} \tag{20}$$

where $l_i(\theta) = \log p_i(x_{obs}|\theta)$ for $i = 1, 2$. Notably, the negative log-likelihood, $-\ell$, is convex with respect to $\log K$, allowing for computationally efficient numerical optimization and estimation of K (Liu et al. [2015]).

3 Experiments

We demonstrate the capabilities of NCL over four experiments. The first is is a replication of the results in Gutmann and Hyvärinen [2010] learning an ICA model. The second shows how NCL-driven MCMC is consistent with a standard MCMC approach. The third is an example of contrastive learning inspired by the project description where we directly use a trained classifier to approximate the probability density function of one of the classes. Finally, the fourth demonstrates how NCL can be used to estimate the Bayes factor over different models.

3.1 Estimation of ICA model

We aim to verify the results from Gutmann and Hyvärinen [2010] about estimating an ICA model.

3.1.1 Data

Under the ICA model we consider data generated by

$$\boldsymbol{x} = A\boldsymbol{s} \tag{21}$$

where $A \in \mathbb{R}^{4\times4}$ is a mixing matrix and $\boldsymbol{s} \in \mathbb{R}^4$ is drawn from a Laplace density with zero mean and unit variance. We note the density of $\boldsymbol{x}$ is given by

$$p_m(\boldsymbol{x}, \theta) = \exp\left(-\sqrt{2} \sum_{i=1}^{4} |\boldsymbol{b}_i \boldsymbol{x}| + c \right) \tag{22}$$

where $\theta = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_4, c)$, $\boldsymbol{b}_i$ is the i^{th} row of $B = A^{-1}$ and $c \in \mathbb{R}$ is the normalising constant. Our contrastive density, $p_n(\boldsymbol{y})$, is chosen to be the Gaussian density with zero mean and the same covariance matrix as $\mathbf{x}$, $\Sigma = A \cdot A^T$. The parameters θ are therefore estimated by learning to discriminate between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Similar to Gutmann and Hyvärinen [2010] we optimise the log-likelihood ℓ using the conjugate gradient algorithm.

3.1.2 Results

We performed the estimation of parameters for sample sizes, N, ranging from 10^2 to 10^3. For each sample size we generated 150 A matrices, giving us our target parameters. For each A we began an optimization at 5 distinct initialisations to prevent finding local optima. The estimates of θ were measured using the MSE $E||\hat{\theta} - \theta^*||^2$. In fig. 1 we report the median MSE for each sample size of the estimates of B and c separately. Our results match those in Gutmann and Hyvärinen [2010] and we expect they would have been even closer had we used 500 A matrices instead of 150 as in their work.

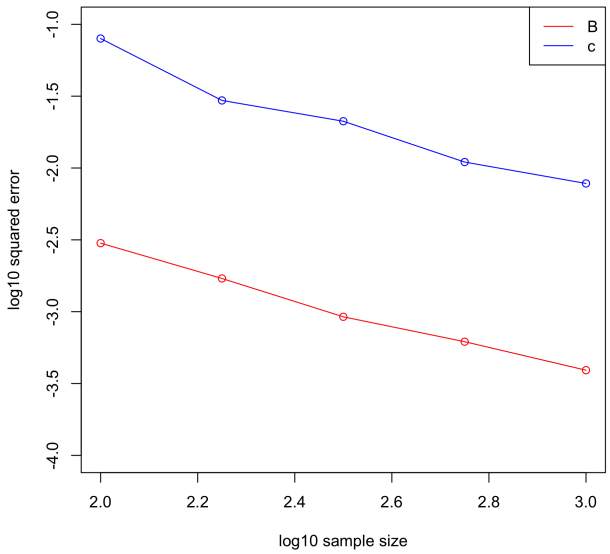


Figure 1: Estimation accuracy

3.2 Fitting a posterior distribution using MCMC and contrastive learning

We demonstrate that the approach from Hermans et al. [2020] can retrieve the same posterior distribution $p(\theta|x)$ as a standard Metropolis-Hastings approach leveraging the tractable likelihood $p(x|\theta)$.

3.2.1 Problem definition

We assume that $\theta \in \mathbb{R}^5$ and that the prior distribution is given by $p(\theta_j) = \mathcal{U}(-3, 3)$ for every $j = 1..5$. The generative model of data x is given by

$$
\begin{aligned}
\mu_\theta &= (\theta_1, \theta_2) \\
\Sigma_\theta &= \begin{bmatrix} \theta_3^4 & \theta_3^2\theta_4^2 \tanh\theta_5 \\ \theta_3^2\theta_4^2 \tanh\theta_5 & \theta_4^4 \end{bmatrix} \\
x_i &\sim \mathcal{N}(\mu_\theta, \Sigma_\theta) \quad \forall i = 1, \ldots, 4 \\
x &= (x_1, ..., x_4)
\end{aligned} \tag{23}
$$

Thereby, the likelihood $p(x|\theta)$ is available in closed form. However, the posterior $p(\theta|x)$ is non-trivial, notably due to the presence of more than one mode induced by the squaring operations. As in Hermans et al. [2020], we generate x_0 from $\theta^* = (0.7, -2.9, -1.0, -0.9, 0.6)$ and estimate the posterior distribution $p(\theta|x_0)$ using Metropolis-Hastings with the standard likelihood and the ratio.

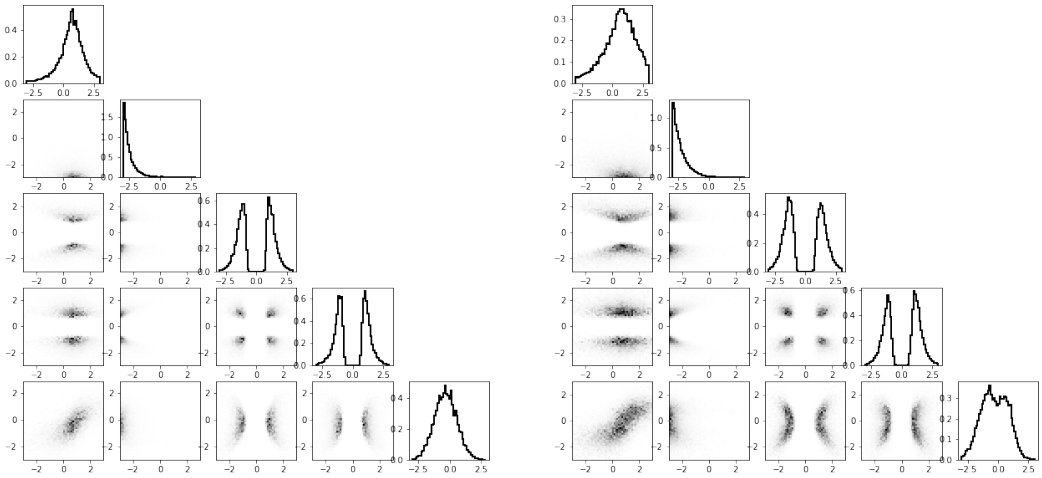
3.2.2 Estimation procedures

During the MCMC step for both the likelihood and the trained ratio, we first generate a burn-in chain with 2×10^4 samples. We then generate a chain with 10^6 samples. We used a standard $\mathcal{N}(\theta'|\theta, I)$ distribution for the proposal.

The ratio was trained using the same hyperparameters as in Hermans et al. [2020] : a SELU activation, Adam with a 0.001 learning rate (without scheduling), AMSGrad and no weight decay, a MLP architecture, no batch normalization or dropout, a 256 batch size, 250 epochs. Only the number of hidden layers was not given, we kept the 64 value used in the repository's MCMC tutorial.

3.2.3 Results

We report MCMC histograms for every component θ_i and pair of components $(\theta_i, \theta_j), i < j$, in fig. 2. We observe similar histograms, except more notably for the last dimension whose posterior marginal derived from the ratio estimator has two narrow modes, whereas that derived from the likelihood only has one mode. In general, this example shows that the ratio estimator can be reliably used for MCMC inference.



(a) MCMC histograms from the likelihood (b) MCMC histograms from the ratio estimator

Figure 2: MCMC histograms derived from the likelihood (left) and the ratio estimator (right).

3.3 Learning a Gaussian mixture using neural network classifiers

We demonstrate noise contrastive learning on artificial data, learning the pdf p_d of a two-dimensional Gaussian mixture that we cannot evaluate but from which we can sample.

3.3.1 Data and classifier models

Our ground-truth model p_d is a mixture of two two-dimensional Gaussian random variables

$$Z_1 \sim N(\mu_1, \Sigma), \quad Z_2 \sim N(\mu_2, \Sigma) \tag{24}$$

$$\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix} \tag{25}$$

with an equal probability of being drawn from Z_1 or Z_2. Our noise distribution p_n is a standard two-dimensional Gaussian $N(\mathbf{0}, I_2)$. We took 100,000 samples from each of p_d and p_n and reserved 20% of samples for validation.

We perform the experiment using two neural network architectures $d(.;\omega)$. The first is a single-layer network with 10 neurons. The second is a multi-layer network with three layers containing 20, 20 and 10 neurons respectively. Both networks use ReLU activations for the hidden layers and a single neuron with a sigmoid activation for the output layer. Training was performed over 10 epochs using Adam for optimization with a binary cross-entropy loss.

3.3.2 Results

The classifiers performed similarly well, with the deeper network converging faster and to a slightly lower loss. The final metrics are shown in Table 1, from which we can see that neither network showed signs of overfitting.

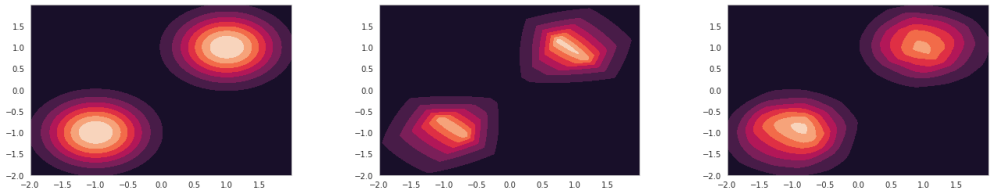
Network	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
Single-layer	0.4666	78.34%	0.4640	78.74%
Multi-layer	0.4595	78.69%	**0.4584**	**78.87%**

Table 1: Final classification losses and accuracies.

We then use our trained networks to derive an approximation p_m to p_d by solving Eq. (4) for p_m, shown in Figure 3. We measure the reconstruction error between p_d and an approximation p_m by

$$\int_{-2}^{2}\int_{-2}^{2} \|p_m(x) - p_d(x)\|^2 dx \tag{26}$$

and the results are shown in Table 2, using noise p_n as a baseline. We can see the the error from our multi-layer network pdf was around 4x lower than from the single-layer network pdf - looking at Figure 3 we see this is likely from the deeper network's expressive power allowing for a better, less jagged fit to the two bell curves. Even the single-layer network however allowed the bimodal distribution to be recovered well, showing that a complex classifier is not needed for an approximation to p_d with this method.



(a) Ground truth p_d

(b) Derived pdf from single-layer network

(c) Derived pdf from multi-layer network

Figure 3: Derived approximations of the Gaussian mixture pdf p_d from two trained classifiers. The deeper network shows a better fit to the shape, but both recover the two peaks.

Approximating pdf	Error
Standard Gaussian noise p_n	0.123711
Derived pdf from single-layer network	0.010402
Derived pdf from multi-layer network	**0.002414**

Table 2: Reconstruction error as described in (26) for approximations to p_d derived from our two classifiers, using the noise p_n as a baseline.

3.4 Estimating Bayes factors using NCL

We demonstrate the ability of NCL as a means of estimating Bayes factors.

3.4.1 Models

Consider $x_0 \in \mathbb{R}$ generated from one of

$$\mathcal{M}_1 : x|\theta \sim \mathcal{N}(\cdot\,; -\theta^2,\, 1), \quad \mathcal{M}_2 : x|\theta \sim \mathcal{N}(\cdot\,; \theta^2,\, 1), \quad \mathcal{M}_3 : x|\theta \sim \mathcal{N}(\cdot\,; \theta^2,\, 2), \tag{27}$$

where θ has a $\mathcal{U}(-30, 30)$ prior distribution. More precisely, we compute the Bayes factor in each of

1. x_0 has been generated from $\mathcal{M}_1$ with $\theta^* = 1$, we select between $\mathcal{M}_1$ and $\mathcal{M}_2$;
2. x_0 has been generated from $\mathcal{M}_1$ with $\theta^* = 1$, we select between $\mathcal{M}_1$ and $\mathcal{M}_3$;
3. x_0 has been generated from $\mathcal{M}_2$ with $\theta^* = 1$, we select between $\mathcal{M}_1$ and $\mathcal{M}_2$.

We generate posterior samples of θ using Metropolis-Hastings with a burning chain of 100 samples and

We expect the Bayes factor to support $\mathcal{M}_1$ in both scenarios 1 and 2, and support $\mathcal{M}_3$ in scenario 3. We also expect the Bayes factor to support $\mathcal{M}_1$ with less confidence in scenario 2 compared to scenario 1 : x_0 will most likely take a negative value, and the posterior of θ in $\mathcal{M}_3$ will have a peak at 0, making the normal distribution closest as possible to negative means (as the parameter posteriors will do for $\mathcal{M}_2$ in scenario 1 or $\mathcal{M}_1$ in scenario 3, this time to approach positive means).

3.4.2 Results

Table 3 summarises our results. We notice that the Bayes Factors are consistent with our expectations. The positive values of $\log K$ in scenarios 1 and 2 indicate support for $\mathcal{M}_1$. The smaller value of $|\log K|$ in scenario 2 compared to scenario 1 suggests we are less confident in $\mathcal{M}_1$. The negative value of $\log K$ in scenario 3 indicates support for $\mathcal{M}_2$. However, we note the Bayes factors might indicate relatively low confidence, as we might consider that we have strong evidence against either model when $|\log K| > 3$ (Kass and Raftery [1995]).

Scenario	x_0 generated from	Models tested	x_0	Definition of K	$\log K$
1	$\mathcal{M}_1$	$\mathcal{M}_1$ vs. $\mathcal{M}_2$	-1.0136	$K = p_1(x)/p_2(x)$	**1.60**
2	$\mathcal{M}_1$	$\mathcal{M}_1$ vs. $\mathcal{M}_3$	-1.0136	$K = p_1(x)/p_3(x)$	**0.97**
3	$\mathcal{M}_2$	$\mathcal{M}_1$ vs. $\mathcal{M}_2$	1.7403	$K = p_1(x)/p_2(x)$	**−2.07**

Table 3: Bayes factors estimated in our model comparisons.

4 Conclusion

We have presented the underlying theory of NCL alongside several experiments demonstrating the practical utility of the method. We have found NCL to be an effective method with strong theoretical support. Although our experiments were limited to relatively small datasets, NCL has seen successful application in Image Processing and Natural Language Processing (Mnih and Kavukcuoglu [2013]).

Acknowledgements

We would like to thank our project supervisor, Geoff Nicholls, for his numerous explanations and ideas, particularly with respect to the application of NCL to Bayes factor estimation.

Code

Code for all experiments can be found at `https://github.com/mkomod/statml-bayes`.

Further detail and figures for Experiment 3.3, as well as the code to replicate it, can be found in `https://github.com/mkomod/statml-bayes/blob/master/python/nn_classification.ipynb`.

References

A. E. Gelfand and D. K. Dey. Bayesian model choice: Asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(3):501–514, 1994.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

J. Hermans, V. Begy, and G. Louppe. Likelihood-free mcmc with amortized approximate ratio estimators, 2020.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430): 773–795, 1995.

Q. Liu, J. Peng, A. Ihler, and J. Fisher III. Estimating the partition function by discriminance sampling. In *Uncertainty in Artificial Intelligence (UAI) 2015*, 2015.

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. *Adv. Neural Inf. Process. Syst.*, pages 1–9, 2013.

I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366. AUAI Press, 2006.

O. Thomas, R. Dutta, J. Corander, S. Kaski, and M. U. Gutmann. Likelihood-Free Inference by Ratio Estimation. *Bayesian Anal.*, (0), 2020.