ECE 595: Homework 3

Manish Nagaraj, PUID: 0029904105
(Spring 2021)

Exercise 1

(a) Consider the vector x and matrix A

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1d} \\ A_{21} & A_{22} & \dots & A_{2d} \\ \vdots & \vdots & \dots & \vdots \\ A_{d1} & A_{d2} & \dots & A_{dd} \end{bmatrix}$$

$$x^T A x = \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1d} \\ A_{21} & A_{22} & \dots & A_{2d} \\ \vdots & \vdots & \dots & \vdots \\ A_{d1} & A_{d2} & \dots & A_{dd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix}$$
$$= \big(x_1(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d) \quad + \quad \dots \quad + \quad x_d(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d)\big)$$

And,

$$tr[Axx^T] = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1d} \\ A_{21} & A_{22} & \dots & A_{2d} \\ \vdots & \vdots & \dots & \vdots \\ A_{d1} & A_{d2} & \dots & A_{dd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}$$
$$= tr \begin{bmatrix} x_1(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d) & \dots \\ \ddots & \vdots \\ \vdots & x_d(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d) \end{bmatrix}$$
$$= \big(x_1(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d) \quad + \quad \dots \quad + \quad x_d(a_{11}x_1 + a_{12}x_2 + \dots a_{1d}x_d)\big)$$

$$\boxed{\therefore x^T A x = tr[Axx^T]}$$

(b)

$$p(\mathcal{D}|\Sigma) = \prod_{n=1}^{N} \Big\{ \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x_n - \mu)^T \Sigma^{-1} (x_n - \mu)\} \Big\}$$

$$\boxed{p(\mathcal{D}|\Sigma) = \frac{1}{(2\pi)^{Nd/2}} |\Sigma^{-1}|^{N/2} \exp\left\{ \frac{1}{2} tr\Big[\Sigma^{-1} \sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T\Big] \right\}}$$

(c) Let,

$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)(x_n-\mu)^T \quad A = \Sigma^{-1}\hat{\Sigma}$$

$$\begin{aligned}p(\mathcal{D}|\Sigma) =& \frac{1}{(2\pi)^{Nd/2}}|\Sigma^{-1}|^{N/2}\exp\Big\{\frac{1}{2}tr\Big[\Sigma^{-1}\sum_{n=1}^{N}(x_n-\mu)(x_n-\mu)^T\Big]\Big\}\\ p(\mathcal{D}|\Sigma) =& \frac{|A|^{N/2}}{(2\pi)^{Nd/2}|\hat{\Sigma}^{N/2}|}\exp\Big\{-\frac{N}{2}tr\Big[\Sigma^{-1}\hat{\Sigma}\Big]\Big\}\end{aligned}$$

The determinant of a matrix is the product of the eigen values and the trace of a matrix is a sum of its eigen values.

$$\boxed{p(\mathcal{D}|\Sigma) = \frac{1}{(2\pi)^{Nd/2}|\hat{\Sigma}^{N/2}|}\Big(\prod_{i=1}^{d}\lambda_i\Big)^{N/2}\exp\Big\{-\frac{N}{2}\sum_{i=1}^{d}\lambda_i\Big\}}$$

(d)

$$\underset{\lambda_1,\ldots,\lambda_d}{\arg\max}\, p(\mathcal{D}|\Sigma) = \underset{\lambda_1,\ldots,\lambda_d}{\arg\max}\, log\Big(p(\mathcal{D}|\Sigma)\Big)$$

can be found by solving

$$\nabla_{\lambda_i} log\Big(p(\mathcal{D}|\Sigma)\Big) = 0, \quad \forall i \in \{1,2,...,d\}$$

$$\begin{aligned}\frac{\delta}{\delta\lambda_i} =& \frac{\delta}{\delta\lambda}\Big(log\Big(\frac{1}{(2\pi)^{Nd/2}|\hat{\Sigma}^{N/2}|}\Big) + \frac{N}{2}\sum_{i=1}^{d}log(\lambda_i) - \frac{N}{2}\sum_{i=1}^{d}\lambda_i\Big) = 0\\ & \frac{1}{\lambda_i} - \lambda_i = 0 \implies \boxed{\lambda_i = 1} \quad \forall i \in \{1,2,...,d\}\end{aligned}$$

(e) Since all eigen values are 1 and we are assuming A is diagonalizable, $A = I$

$$\therefore A = \Sigma_{ML}^{-1}\hat{\Sigma} = I \implies \hat{\Sigma}_{ML} = \hat{\Sigma}$$

$$\boxed{\therefore \hat{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)(x_n-\mu)^T}$$

(f) An alternate way to find $\hat{\Sigma}_{ML}$ is by finding the Σ that maximizes $p(\mathcal{D}|\Sigma)$. i.e.,

$$\hat{\Sigma}_{ML} = \underset{\Sigma}{\arg\max}\, p(\mathcal{D}|\Sigma) = \underset{\Sigma}{\arg\max}\, \log(p(\mathcal{D}|\Sigma))$$

This can be solved by equating the derivative of the log of the likelihood function with respect to Σ.

$$\nabla_{\Sigma}\log(p(\mathcal{D}|\Sigma)) = 0$$

(g) The unbiased estimate $\hat{\Sigma}_{unbias}$ is

$$\boxed{\hat{\Sigma}_{unbias} = \frac{1}{N}\Big(\sum_{n=1}^{N}(x_n - \hat{\mu})(x_n - \hat{\mu})^T\Big) + \Big(\frac{1}{N}\sum_{n=1}^{N} x_n\Big)^2}$$

Exercise 2

(a)

$$p_{Y|X}(C_1|x) \gtrless_{C_0}^{C_1} p_{Y|X}(C_0|x)$$
$$p_{X|Y}(x|C_1)p_Y(C_1) \gtrless_{C_0}^{C_1} p_{X|Y}(x|C_0)p_Y(C_0)$$
$$\log(p_{X|Y}(x|C_1)p_Y(C_1)) \gtrless_{C_0}^{C_1} \log(p_{X|Y}(x|C_1)p_Y(C_1))$$

$$\boxed{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \log\pi_1 - \frac{1}{2}\log|\Sigma_1| \gtrless_{C_0}^{C_1} -\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) + \log\pi_0 - \frac{1}{2}\log|\Sigma_0|}$$

(b)

(i) First two entries of $\mu_1 = (0.4408, 0.4387)$ and $\mu_0 = (0.4824, 0.4864)$

(ii) First 2 x 2 entries of the Σ matrices

$$\Sigma_1 = \begin{bmatrix} 0.0430 & 0.0353 \\ 0.0353 & 0.0424 \end{bmatrix} \quad \Sigma_0 = \begin{bmatrix} 0.0644 & 0.0369 \\ 0.0369 & 0.0662 \end{bmatrix}$$

(iii) Value of $\pi_1 = 0.1713$ and $\pi_0 = 0.8286$

(c) The binary image generated from running the Bayesian classifier

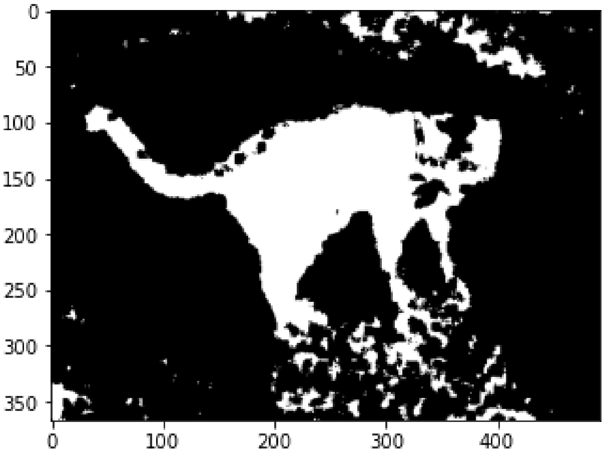


Figure 1: Output of the Bayesian classifier

(d) The Mean Absolute Error (MAE) = 0.09109

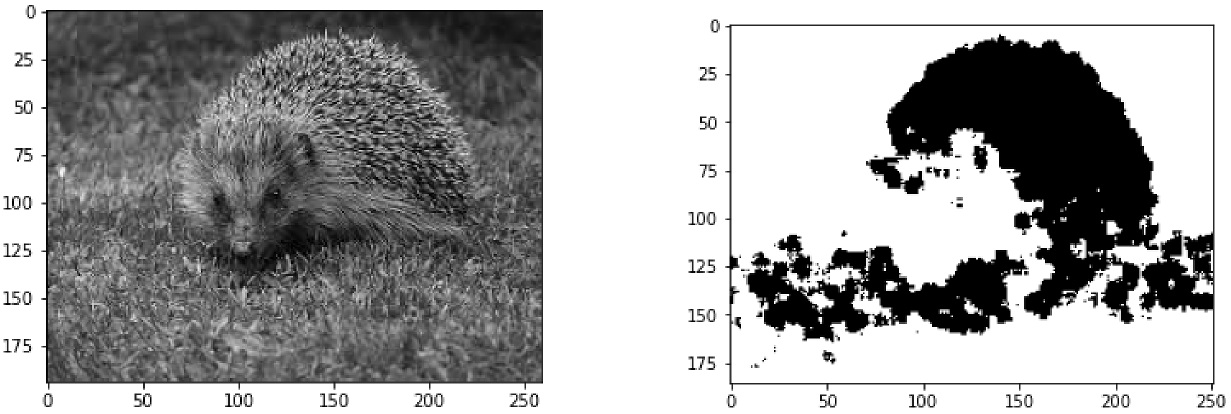


Figure 2: Image of animal on a grass and the mask generated using the classifier

(e) The classifier does not perform very well. This is because there is not enough training data. The classifier has not been trained on images such as the hedgehog image shown in Figure 2

Exercise 3

(a) The Bayesian decision rule is

$$-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)+\log\pi_1-\frac{1}{2}\log|\Sigma_1| \gtrless_{C_0}^{C_1} -\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)+\log\pi_0-\frac{1}{2}\log|\Sigma_0|$$

which is,

$$\log p_{X|Y}(x|C_1)+\log(\pi_1) \gtrless \log p_{X|Y}(x|C_0)+\log(\pi_0)$$

$$\log\left(\frac{p_{X|Y}(x|C_1)}{p_{X|Y}(x|C_0)}\right) \gtrless \log\left(\frac{\pi_0}{\pi_1}\right)$$

$$\frac{p_{X|Y}(x|C_1)}{p_{X|Y}(x|C_0)} \gtrless \frac{\pi_0}{\pi_1}$$

Hence the threshold for the decision rule in Exercise 2 is

$$\boxed{\therefore \tau = \frac{\pi_0}{\pi_1}}$$

(b) The ROC curve of the classifier is plotted in Figure 3

(c) The operating Bayesian decision rule is marked as a red dot in Figure 4

(d) The ROC of a linear regression classifier is shown in Figure 5

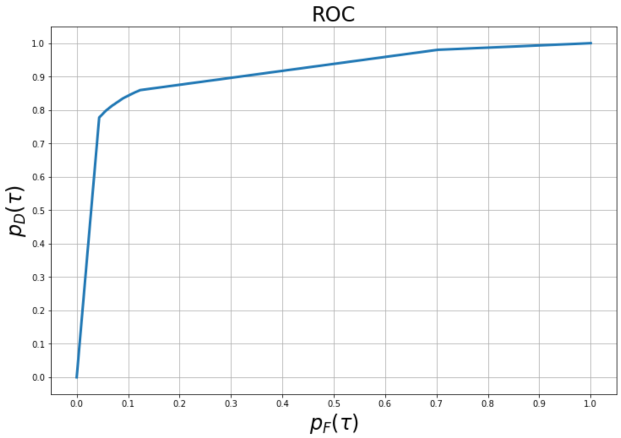


Figure 3: ROC of the Bayesian classifier

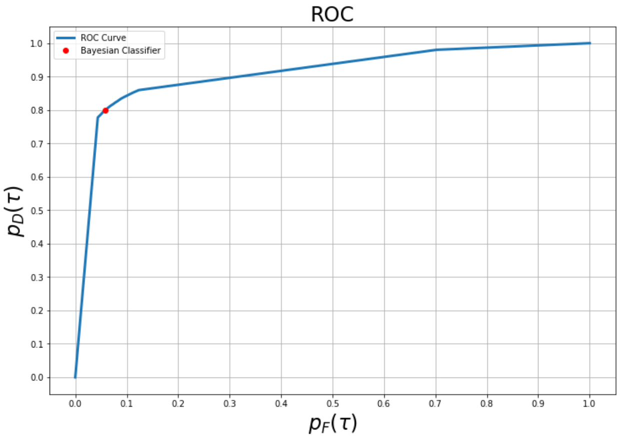


Figure 4: ROC showing the operating point

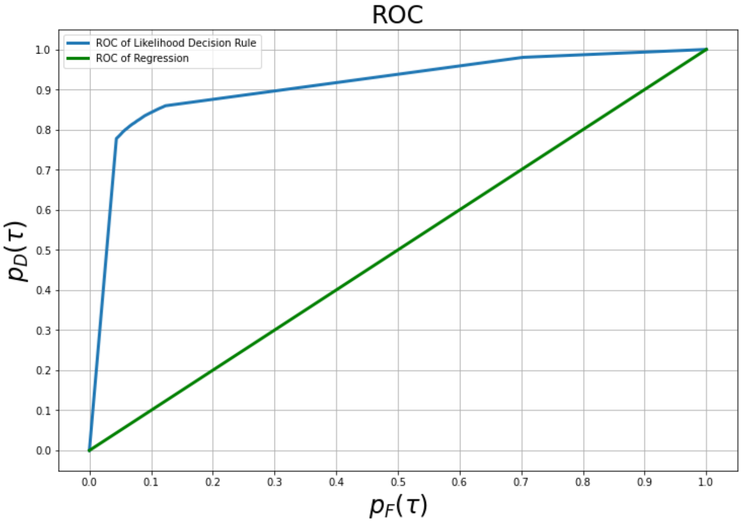


Figure 5: ROC of linear regression classifier overlayed with ROC of Bayesian classifier

Exercise 4: Project: Check Point 3

The project report with check point 3 is attached below.

ECE595 Project - Exploring Learning with Noisy Labels

Manish Nagaraj [1]

Abstract

This is checkpoints 1 and 2 of the project.

1. Checkpoint 1

1.1. Author Details

- **Name:** Manish Nagaraj
- **Major:** Electrical and Computer Engineering
- **Level:** Ph.D.

1.2. Details of Project

This project aims to study the limitations of the methods proposed in the paper referenced in the title (Ren et al., 2018b) using the CleanLab ML framework (Northcutt et al., 2019). Further, it also aims in identifying and implementing any algorithms or techniques that will overcome these limitations.

1.2.1. Datasets

For the purpose of evaluating existing algorithms as well as potential methods proposed, we will be using **CIFAR10**, **MNIST**, **FMNIST** datasets.

1.3. Reference Codes

We will be using the following code sources available as a reference to help implement the project.

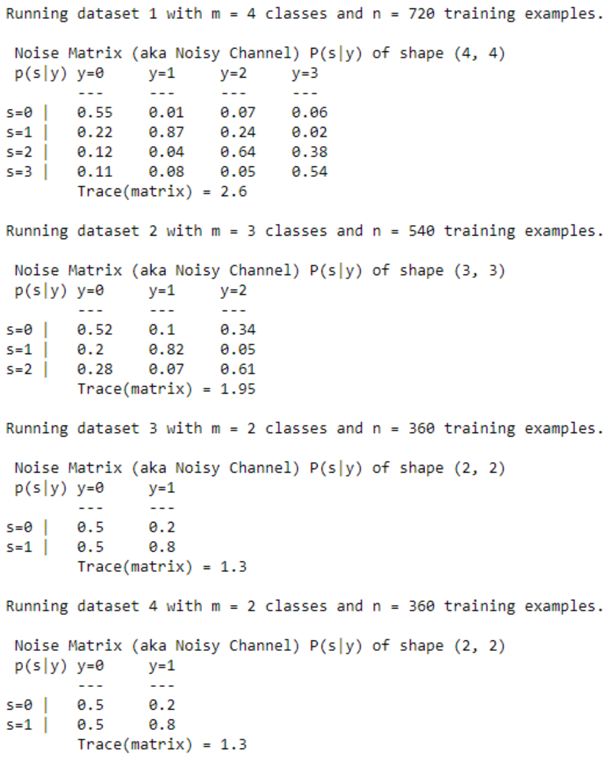
1. CleanLab Framework
2. Learning to Learn by Gradient Descent

2. Checkpoint 2

2.1. Checkpoint 2 Baseline

The baseline shows the performance of standard multi-class and binary classifiers in the presence of noisy labels.

The noise matrices that are used in the datasets are represented in the Figure 1. The s represents the observed noisy labels and the y represents the true labels. If the Trace of the matrix is 4, then there are no noisy values and if the trace is 0, then there all labels are noisy.



```
Running dataset 1 with m = 4 classes and n = 720 training examples.

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (4, 4)
 p(s|y) y=0     y=1     y=2     y=3
        ---     ---     ---     ---
s=0 |   0.55    0.01    0.07    0.06
s=1 |   0.22    0.87    0.24    0.02
s=2 |   0.12    0.04    0.64    0.38
s=3 |   0.11    0.08    0.05    0.54
        Trace(matrix) = 2.6

Running dataset 2 with m = 3 classes and n = 540 training examples.

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (3, 3)
 p(s|y) y=0     y=1     y=2
        ---     ---     ---
s=0 |   0.52    0.1     0.34
s=1 |   0.2     0.82    0.05
s=2 |   0.28    0.07    0.61
        Trace(matrix) = 1.95

Running dataset 3 with m = 2 classes and n = 360 training examples.

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (2, 2)
 p(s|y) y=0     y=1
        ---     ---
s=0 |   0.5     0.2
s=1 |   0.5     0.8
        Trace(matrix) = 1.3

Running dataset 4 with m = 2 classes and n = 360 training examples.

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (2, 2)
 p(s|y) y=0     y=1
        ---     ---
s=0 |   0.5     0.2
s=1 |   0.5     0.8
        Trace(matrix) = 1.3
```

Figure 1. $\mathbf{P}(s|y)$ Noise Matrices of the datasets used.

Figure 2 shows the baseline operation of standard binary and multi class classifiers on datasets that are generated. Figure 3 shows the operation of the same classifiers on the datasets with noisy labels. The label errors are circled in green. Figure 4 shows the performance of the classifiers when the inbuilt Learning with Noisy Labels function of CLEANLAB package is used. In each subfigure, the accuracy scores on a test set are shown as decimal values. The leftmost values in black are the test accuracies of the classifiers trained with perfect labels. The middle values in value are the accuracies are the test accuracies of the classifiers trained with noisy

[1] Department of Electrical and Computer Engineering, Purdue University, USA. Correspondence to: Manish Nagaraj <mnagara@purdue.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

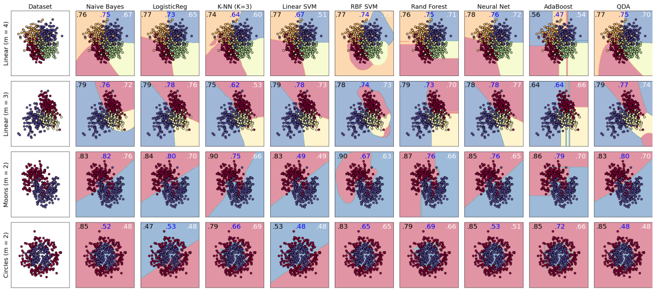


Figure 2. Baseline performance of classifiers with no noisy labels

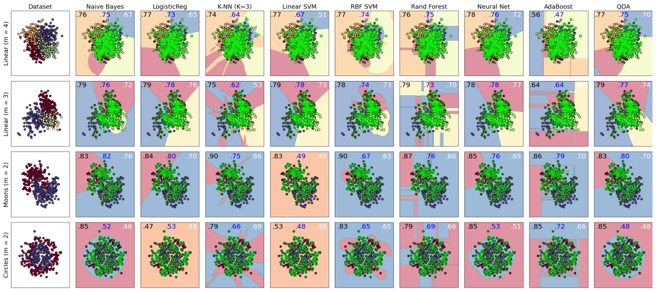


Figure 3. Baseline performance of classifiers with noisy labels (regular training)

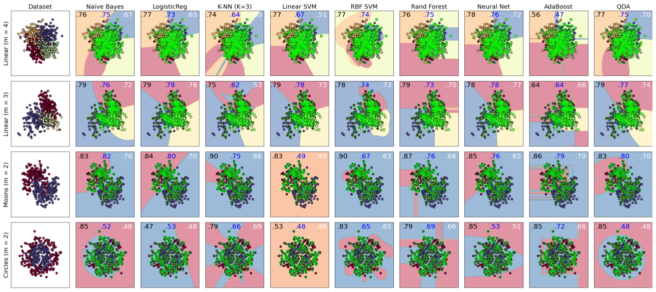


Figure 4. Performance of classifiers with noisy labels with CLEANLAB's inbuilt function

labels using CLEANLAB's inbuilt function. The rightmost values in white are the accuracies of the classifiers trained with noisy labels and no algorithm to rectify the noise.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Northcutt, C. G., Wu, T., and Chuang, I. L. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'17. AUAI Press, 2017. URL `http://auai.org/uai2017/proceedings/papers/35.pdf`.

Northcutt, C. G., Jiang, L., and Chuang, I. L. Confident learning: Estimating uncertainty in dataset labels, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. 2016.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018a.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018b.