Assignment 2

Tutor: Rafiul Nakib
Group members:
Tiannan Chen (SID: 500230978, Unikey: Tche5904)
Haochen Zhang(SID: 490169971, Unikey: hzha9205)

Abstract

This report mainly studies how to reduce the negative impact of label noise in the classifier. We have adopted two methods: reweighting and unbiased estimator to process robustness. In order to ensure that these two methods are effective for different classifiers, we also use two classifiers, FCNet and CNN. This article will be introduced in the order of introduction, core ideas, related work, methods, experiments, and conclusions. Among them, methods will introduce data preprocessing and two robust methods. The reweighting part will be described separately from Image Classification with Known Flip Rates and Image Classification with Unknown Flip Rates, and the unbiased estimator will be described in a separate part.Our final conclusion is that CNN+reweighting has the highest prediction accuracy.

1 Introduction

Label is an important factor that affects the prediction accuracy of the machine learning model. If there is noise in the label of the training set, the prediction results will be much reduced, and it may even be impossible to predict effectively. Therefore, we need to reduce the negative impact of label noise to improve model prediction accuracy. The processing of label noise can be applied in many scenarios. For example, it is more common in image recognition. Due to image blur or bias of workers during annotation, these reasons will lead to label noise. Effective processing of label noise can improve the image quality and recognition accuracy.

We use two methods: reweighting and unbiased estimator(customizes loss function). Reweighting uses the transition matrix to correct the prediction results to improve the accuracy. This method mainly focuses on revising the prediction results.The unbiased estimator customizes the loss function to change the loss every epoch to correct it. Both methods can effectively improve prediction accuracy. This method mainly focuses on revising the prediction process.

2 Related work

Label noise has been a concern for many years, and learning from label noise-affected datasets has been studied in a number of methods across disciplines. We focus on three types of label noise in this paper: Random Classification Noise (RCN), Class-Dependent Noise (CCN), and Instance- and Label-Dependent Noise (ILN). Integration methods such as Bagging or Boosting can be used to lessen the influence of the first form of label noise, however the problem of this strategy is that such enhancement algorithms are particularly vulnerable to random classification noise [3]. Tao and Liu offered a very excellent solution for the second Class-dependent Label Noise solution, which is that any alternative loss function may be utilised for classification and noisy labels by employing

significance reweighting [2]. By comparing the empirical risk minimisers obtained from clean and noisy samples, a new excess risk bound proportional to the noise level is produced for the third kind of label noise [1]. The downside of this strategy is that learning with only noisy samples is impossible without clean samples or strong assumptions about the distribution of the data.

3 Methods

In the method section, we use two different classifiers, one is Convolutional neural network (CNN) and the other is fully connected neural network (FCNet).And we apply two different methods to make them robust, one of them is using transition matrix and the other is using unbiased-estimator, which are applied to two different classifiers. Before starting the experiments, each dataset was subjected to data preprocessing, where each pixel point was normalised and each pixel point was divided by 255 to ensure that the value of each pixel point was between 0-1, thus improving the efficiency and accuracy of each noise robustness classifiers.

3.1 Classification models

3.1.1 Fully connected neural network (FCNet)

In FCNet, this model includes an input layer, two hidden layers and an output layer. For the MNIST dataset, the size of the input layer is 728 because the input image data of MNIST has a shape of 28x28 while the image shape of the CIFAR dataset is 32x32x3, so the input layer of FCNet on the CIFAR dataset is 3072. For the hidden layer, we have chosen the size of the hidden layer on the MNIST dataset to be 3 , 3. And on the CIFAR dataset, the size of the hidden layer is 100, 3. For the size of the output layer, FCNet is 3 on both datasets because the variety of images in both datasets is 3. In addition, the relu activation function is used between layers.

3.1.2 Convolutional neural network (CNN)

In CNN classifier, this model contains two convolutional layers, the first convolutional layer maps the input image from 1 channel to 6 channels and the second convolutional layer maps 6 channels to 10 channels. These convolutional layers use 5x5 convolutional kernel for feature extraction. After each convolutional layer, a ReLU activation function is used and two maximum pooling layers are used to compress the amount of data and parameters and reduce overfitting. The model also includes two fully connected layers after the maximum pooling layer, the first with an input size of 160 and an output size of 22, and the second with an input size of 22 and an output size of 3, and is finally mapped into 3 categories. Finally softmax is used to calculate the probability of each class. In addition, this CNN classifier also uses drop out layer to prevent overfitting by using 0.5 drop out rate.

3.2 Label noise methods

3.2.1 Reweighting with known flip rates

The principle of the transition matrix is to use a small portion of "clean" (noise-free) data to estimate the transition probability from real labels to noise labels, and then convert the noise labels to a more clean labels. This process can reduce the probability of noise. In order to achieve the effect of training with clean data.

$$\text{Transition Matrix} = \begin{bmatrix} P(0,0) & P(1,0) & P(2,0) \\ P(0,1) & P(1,1) & P(2,1) \\ P(0,2) & P(1,2) & P(2,2) \end{bmatrix} \tag{1}$$

The transition matrix is a square matrix whose size is equal to the number of categories. Because the three data sets all have three categories, the transition matrix T is a 3x3 matrix. P(0,2) represents the probability that the point is converted to 2 if it is 0. Since both data sets FashionMINIST0.5 and FashionMINIST0.6 provide transition matrices, the each prediction result is multiplied by the known transition matrix before training the prediction model.

$$ProcessedResult = PredicitonResult \times TransitionMatrix \tag{2}$$

$$\begin{bmatrix} P(\tilde{Y}=1|X) & \cdots & P(\tilde{Y}=C|X) \end{bmatrix}^T = \mathbf{T} \begin{bmatrix} P(Y=1|X) & \cdots & P(Y=C|X) \end{bmatrix}^T \tag{3}$$

3.2.2 Reweighting with unknown flip rates

The data set CIFAR does not provide a known transition matrix, so the difference is that the transition matrix must be found first. The idea of finding the transition matrix is to train in noise data, then predict and compare the difference with the correct answer in clean data, and find the transition matrix through the difference. If the predicted result is x, but the true result is y, then num(x, y) is increased by 1.

$$Matrix = \begin{bmatrix} num(0,0) & num(1,0) & num(2,0) \\ num(1,0) & num(1,1) & num(2,1) \\ num(2,0) & num(1,2) & num(2,2) \end{bmatrix} \tag{4}$$

Then each point is divided by the number of predicted data, and each num(x,y) will become P(x,y). The generated matrix is the transition matrix.

$$TransitionMatrix = Matrix \div SizeOfData \tag{5}$$

We obtained the transition matrix of CIFAR through the above method:

$$CIFAR\_TransitionMatrix = \begin{bmatrix} 0.6850 & 0.1500 & 0.1650 \\ 0.1620 & 0.4260 & 0.4120 \\ 0.2220 & 0.0360 & 0.7420 \end{bmatrix} \tag{6}$$

Then just repeat the steps in reweighting with known flip rates.

3.2.3 Unbiased estimator

The principle is to customize the loss function through the noise model. First, we need to guess or measure the probability that a certain type of label is incorrectly labeled as another type. Then we can compensate for this loss by modifying the loss function. For example, we think we were wrong The probability of marking is noise_rate, then our loss function calculation formula is:

$$corrected\_loss = \frac{loss}{1 - noise\_rate} \tag{7}$$

Then we use this new loss to train the model, which will eliminate some of the errors caused by noise. We tried a variety of different probabilities, and the effect was better when the probability was 10%, so we let noise_rate=0.1 to train our model.

4 Experiments

The datasets used in this experiment are FashionMNIST and CIFAR, where the MNIST data is divided into two with distinct label noise types and the associated transition matrix is provided.The MNIST dataset comprises 18000 training and validation samples, with a total of 3000 test samples. The MNIST dataset comprises 18000 training and validation samples, 3000 test samples, and each sample picture is 28x28 pixels in size. The CIFAR dataset comprises 30000 training and validation samples, as well as 3000 test samples, however each sample picture shape is 32x32. Convolutional neural network (CNN) and Fully connected neural network (FCNet) classifiers with reweighting and unbiased-estimation were utilised. unbiased-estimator to make them more robust. In terms of assessment metrics, this experiment employs four distinct evaluation criteria: recall, precision, F1 score, and Top 1 accuracy.

4.1 Noise robustness classifier on FashionMINIST0.5

Table 1: Reweighting (FashionMINIST0.5)

	CNN	FCNet
Top 1 Accuracy	”Mean = 93.56, Std Dev = 0.84”	”Mean = 83.84, Std Dev = 10.30”
Precision	”Mean = 0.94, Std Dev = 0.00726”	”Mean = 0.86, Std Dev = 0.08106”
Recall	”Mean = 0.94, Std Dev = 0.00835”	”Mean = 0.84, Std Dev = 0.10273”
F1	”Mean = 0.94, Std Dev = 0.00817”	”Mean = 0.82, Std Dev = 0.13640”

Table 2: Unbiased-estimation (FashionMINIST0.5)

	CNN	FCNet
Top 1 Accuracy	”Mean = 90.86, Std Dev = 1.70”	”Mean = 51.40, Std Dev = 19.79”
Precision	”Mean = 0.91, Std Dev = 0.01600”	”Mean = 0.76, Std Dev = 0.09592”
Recall	”Mean = 0.91, Std Dev = 0.01683”	”Mean = 0.51, Std Dev = 0.19800”
F1	”Mean = 0.91, Std Dev = 0.01736”	”Mean = 0.40, Std Dev = 0.25979”

For the metrics table above, the table 1 and table 2 shows the two classifiers with reweighting method. It can be seen that all the metrics of Convolutional neural network (CNN) is much higehr than that in Fully connected neural network (FCNet) model. Moreover, the standard deviation of each metric of the CNN model is much lower than that of FCNet, which shows that the robustness of the CNN model with reweighting is much stronger than that of FCNet with reweighting. At the same time, comparing reweighting and unbiased-estimation from table 1 and table 2, reweighting has higher top 1 accuracy, recall, precision and F1 than unbiased-estimation, and the standard deviation is also lower. It shows that using reweighting can make the classifier more robust.

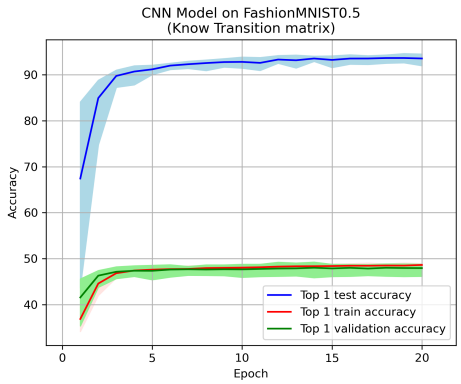

Figure 1: CNN reweighting

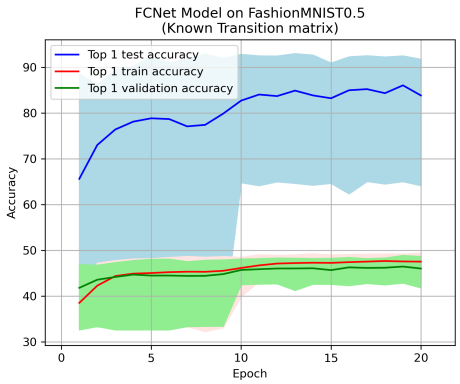

Figure 2: FCNet reweighting

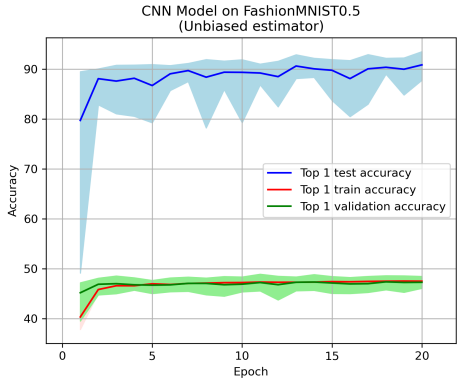

Figure 3: CNN unbiased-estimation

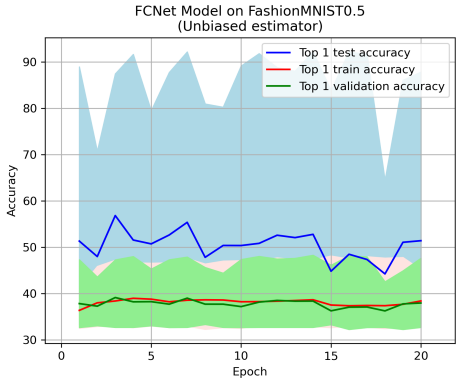

Figure 4: FCNet unbiased-estimation

The above Four figures show the effect of CNN and FCNet using reweighting and unbiased-estimation on the FashionMINIST0.5 data set. It can be found from the figure that the top 1 accuracy of the CNN model combined with reweighting has the best effect on the training set, verification set and test set. The fluctuation of each epoch is very small, indicating strong robustness. The top 1 accuracy of FCNet with unbiased estimation changes greatly in each epoch, indicating that the robustness is weak.

4.2 Noise robustness classifier on FashionMINIST0.6

Table 3: Reweighting (FashionMINIST0.6)

	CNN	**FCNet**
Top 1 Accuracy	”Mean = 88.18, Std Dev = 1.44”	”Mean = 74.56, Std Dev = 15.69”
Precision	”Mean = 0.89, Std Dev = 0.01338”	”Mean = 0.82, Std Dev = 0.05212”
Recall	”Mean = 0.88, Std Dev = 0.01479”	”Mean = 0.75, Std Dev = 0.15706”
F1	”Mean = 0.88, Std Dev = 0.01549”	”Mean = 0.71, Std Dev = 0.20907”

Table 4: Unbiased-estimation (FashionMINIST0.6)

	CNN	**FCNet**
Top 1 Accuracy	”Mean = 73.96, Std Dev = 8.91”	”Mean = 37.35, Std Dev = 8.29”
Precision	”Mean = 0.82, Std Dev = 0.04099”	”Mean = 0.78, Std Dev = 0.00218”
Recall	”Mean = 0.74, Std Dev = 0.08893”	”Mean = 0.37, Std Dev = 0.08313”
F1	”Mean = 0.71, Std Dev = 0.11863”	”Mean = 0.22, Std Dev = 0.11155”

When the dataset is switched to FashionMINIST0.6, from the metrics table 3 and table 4, it can be seen that comparing different classifiers, the metrics of Convolutional neural network (CNN) are still higher than those of Fully connected neural network (FCNet), while the standard deviation of CNN is still lower than that of FCNet, which proves that the robustness of CNN on FashionMINIST0.6 is still stronger than that of FCNet. while comparing different techniques applied on our classifiers, we can see that the unbiased-estimation on the FashionMINIST0.6 dataset has higher average Top 1 accuracy, precision, recall and F1 score than reweighting, which indicates that the technique of unbiased-estimation is more robust than FCNet in classifying the FashionMINIST0.6 dataset. This indicates that the technique of unbiased-estimation has higher accuracy in classifying the FashionMINIST0.6 dataset, but the standard deviation of all the metrics of unbiased-estimation is lower than that of reweighting, which indicates that the robustness of unbiased-estimation in the FashionMINIST0.6 dataset is still weaker than that of reweighting.

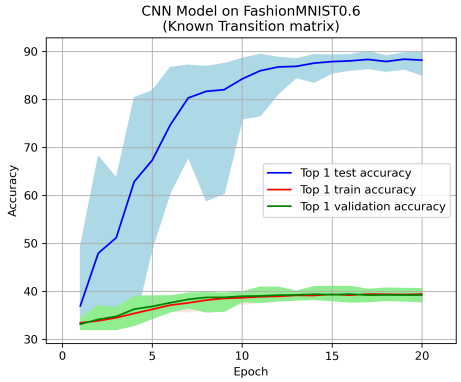

Figure 5: CNN reweighting

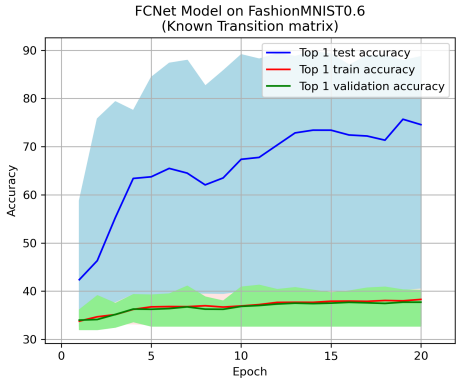

Figure 6: FCNet reweighting

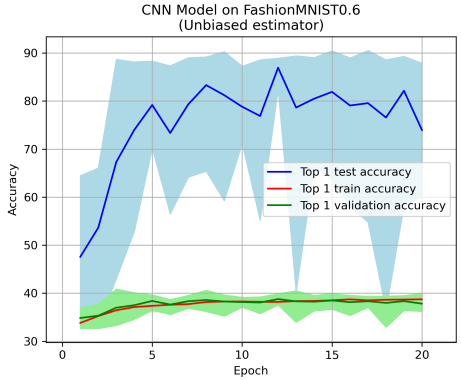

Figure 7: CNN unbiased-estimation

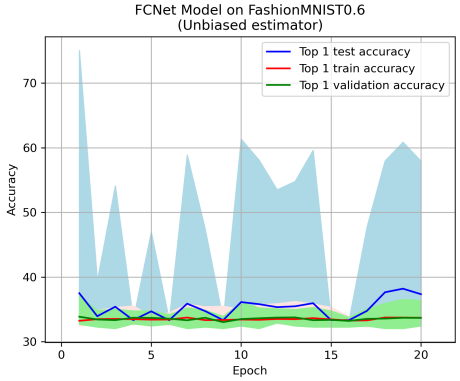

Figure 8: FCNet unbiased-estimation

The above four figures show the effect of CNN and FCNet using reweighting and unbiased estimation on FashionMINIST0.6 dataset. From the graphs, it can be noticed that the top 1 accuracy of the CNN model combined with reweighting works best on the training, validation and test sets. The fluctuation of each epoch compared to FCNet indicates strong robustness. The top 1 accuracy of the unbiased estimation of FCNet varies exceptionally high per epoch, indicating weak robustness. But comparing CNN on FashionMINIST0.5 using reweighting and unbiased estimation the results are worse, the reason may be that on FashionMINIST0.5 Random Classification Noise (RCN) is solved, while on FashionMINIST0.6 it needs to be solved. Class-Dependent Noise (CCN), and CCN requires more complex noise reduction methods than RCN because the label noise of CCN is not random, but also related to the real nature of the data distribution.

4.3 Noise robustness classifier on CIFAR

Table 5: Reweighting (CIFAR)

	CNN	**FCNet**
Top 1 Accuracy	”Mean = 52.45, Std Dev = 4.75”	”Mean = 55.10, Std Dev = 5.51”
Precision	”Mean = 0.56, Std Dev = 0.03640”	”Mean = 0.58, Std Dev = 0.04462”
Recall	”Mean = 0.52, Std Dev = 0.04727”	”Mean = 0.55, Std Dev = 0.05497”
F1	”Mean = 0.50, Std Dev = 0.06306”	”Mean = 0.53, Std Dev = 0.07317”

When the dataset is switched to CIFAR, it can be seen from the above tables **??** and **??** that the metrics of all the metrics decrease significantly regardless of which classifier uses different tech-

Table 6: Unbiased-estimation (CIFAR)

	CNN	FCNet
Top 1 Accuracy	”Mean = 53.55, Std Dev = 5.14”	”Mean = 50.83, Std Dev = 6.39”
Precision	”Mean = 0.59, Std Dev = 0.03630”	”Mean = 0.60, Std Dev = 0.05028”
Recall	”Mean = 0.54, Std Dev = 0.05139”	”Mean = 0.51, Std Dev = 0.06385”
F1	”Mean = 0.50, Std Dev = 0.07194”	”Mean = 0.46, Std Dev = 0.09119”

niques, but the effect of the reweighted FCNet is better than the reweighted CNN, and the standard deviation of each metric of the FCNet is better with the reweighted FCNet than with the reweighted CNN, but the standard deviation of the reweighted CNN is lower than that of FCNet, which indicates that although FCNet with reweighting is better in each metric, CNN with reweighting has stronger stability. However, when the technique uses unbiased estimation, the CNN is better than FCNet in all metrics and has a lower standard deviation, which suggests that the unbiased estimated CNN is more robust than the unbiased estimated FCNet.

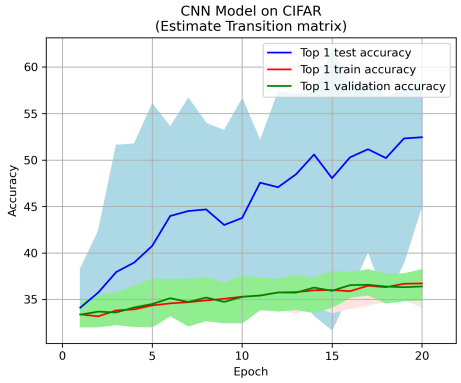

Figure 9: CNN reweighting

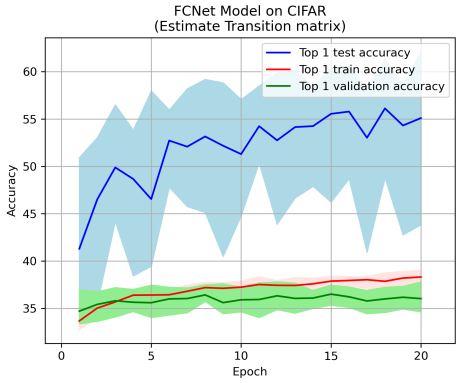

Figure 10: FCNet reweighting

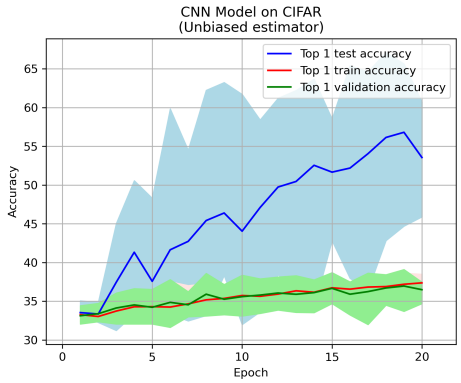

Figure 11: CNN unbiased-estimation

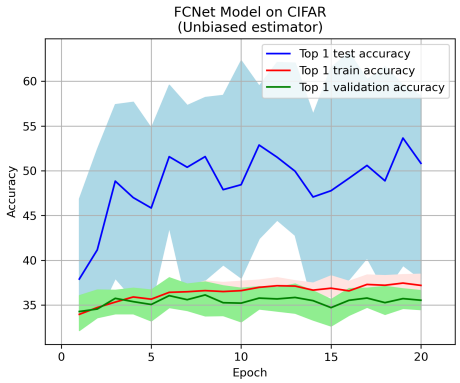

Figure 12: FCNet unbiased-estimation

The above four figures show the effect of using reweighting and unbiased estimation by CNN and FCNet on the CIFAR dataset. As can be seen from the figures, the Top 1 accuracy of FCNet with reweighing is more stable from epoch to epoch. And when unbiased-estimation is used, CNN becomes more stable. However, the accuracy is significantly lower than FashionMINIST in both the training set test set and validation set, which may be caused by the fact that the CIFAR dataset uses Label-dependent Label Noise, which is not affected by the attributes or features of the samples, and the CIFAR dataset is a coloured rather than a grey scale image.

5 Conclusion

This experiment was conducted on two datasets and two noise robustness classifiers for different label noise, and the results show that CNN with reweighting works best on FashionMINIST0.5 and FashionMINIST0.6, as well as being the most robust FCNet with reweighting works best on CIFAR dataset and has more robustness. The robustness of reweighting is stronger than unbiased-estimation on both classifiers, because unbiased-estimation can only work on most of the categories, and if the number of categories is small, the effect may not be so good, while reweighting can change the overall label weight of noise to be more in line with the labels of the test set, improving the generalisation of the model.

For the future work part, we can adopt more different kinds of techniques to make our classifiers more robust, such as performing data augmentation, rotating or flipping the original image, and using the integration methods of bagging or boosting to improve the generalisation ability of the model. We can also add different kinds of label noise in the same dataset to better compare different label noise, and we can introduce grid search and other optimal hyperparameter selection methods when choosing the hyperparameters of the classifiers to make our model more effective.

6 Appendix

The code is divided into two parts: main and algorithm. Algorithm has 8 files including CNN+reweighting, FCNet+reweighting, CNN+unbiased, FCNet+unbiased and the algorithm to calculate the CIFAR transition matrix. Main code includes data processing, training and test functions, configuration setting and visualization.

References

[1] Hyungki Im and Paul Grigas. Binary classification with instance and label dependent label noise. *arXiv.org*, 2023.

[2] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

[3] Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine learning*, 78(3):287–304, 2010.