International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-021-01423-4

ORIGINAL ARTICLE

Towards an improved label noise proportion estimation in small data: a Bayesian approach

Jakramate Bootkrajang[1] · **Jeerayut Chaijaruwanich**[1]

Received: 3 March 2021 / Accepted: 31 August 2021

Abstract
Today's classification task is getting more and more complex. This inevitably renders unanticipated compromises on the quality of data labels. In this paper, we consider learning label noise robust classifiers with focus on the tasks with limited training examples relative to the number of data classes and data dimensionality. In such cases, the existing label noise models tend to inaccurately estimate the noise proportions leading to suboptimal performance. To alleviate the problem, we formulated a regularised label noise model capable of expressing preference on the noise parameters. In addition, we treated the regularisation from a Bayesian perspective so that the regularisation parameters can be inferred from the data through the noise model, thereby facilitating model selection in the presence of label noise. This results in a more data and computationally efficient Bayesian label noise model which could be incorporated into any probabilistic classifier, including those that are known to be data intensive such as deep neural networks. We demonstrated the generality of the proposed method through its integrations with logistic regression, multinomial logistic regression and convolutional neural networks. Extensive empirical evaluations demonstrate that the proposed regularised label noise model can significantly improve, in terms of both the quality of noise parameters estimation and the classification accuracy, upon the existing ones when data is scarce, and is no worse than the existing approaches in the abundance of training data.

Keywords Bayesian regularisation · Label noise estimation · Classification · Small data

1 Introduction

It is increasingly common that today's classification tasks involve learning from inaccurate label information. This might be due to either the errors in sensing or measuring instruments which give rise to ambiguous examples that are difficult to label, the lack of knowledge to determine the correct label or even the disagreement between data annotators [12]. The quality of training labels plays a crucial role in the training of classification models. The error in the training label is known to perturb the decision boundary of supervised learning machine [18, 19, 26, 27], and hence its generalisation ability could be compromised. The discrepancy between the observed label of an example and its true label assignment is referred to as *label noise* in the literature [12]. It is an active research topic both empirically [3, 4, 13, 38, 44] and theoretically [14, 26, 28, 31, 32]. Currently, label noise robust methods are integral part of variety of real-world applications ranging from learning from crowdsourced data [34], label annotator modelling [40], classification of microarray data [6, 29] to analysis of remote-sensing data [43] to name just a few. Moreover, the uncertainty of label information is also being taken into account within the context of semi-supervised learning [23].

Previous approaches for dealing with label noise can generally be categorised into three streams. The first one focuses on the pre-processing of the data which attempts to filter out or make a correction to the suspect label [2, 8, 22]. This is seemingly a good approach but one must be aware that some important information could be lost if the algorithm mistakenly removes correctly labelled data that happens to be difficult to classify. Removing data is also undesirable when the sample size is already small. Further, some of the approaches e.g., [22] even require seeded correct labels from each class in order to perform label correction. Arguably, the requirement could affect real-world applicability of the method when the ground-truths are still debatable. The

✉ Jakramate Bootkrajang
jakramate.b@cmu.ac.th

[1] The Department of Computer Science, Chiang Mai University, Chiang Mai 50200, Thailand

Published online: 14 September 2021

second approach, which is based on the premise that noisy labels incur overfitting, concerns with adjusting the learning algorithm in the spirit of reducing overfitting. Traditional approaches for avoiding overfitting, including early stopping [24, 38] and various kind of model regularisations can be adopted to mitigate the problem [18, 36, 41]. The drawback of this approach is that, in the presence of label noise, overfitting can still occur at model selection level [7, 10] and thus the model selected based on the commonly used validating criteria, such as the validation accuracy, might not necessarily be the one that generalises the best. Effectively choosing an optimal model in noisy settings often relies on validating candidate models on a trusted validation set which is unfortunately not always available in practice [21]. Besides, performing model selection by means of the commonly used cross validation technique is also challenging in small data; partly because of the high variance of the validation scores as the validation set gets proportionally smaller in size. The third approach attempts to directly model the label noise generation process. The resulting label noise model can then be incorporated into the standard classification models. For example, a latent variable label noise model was proposed in [21] for kernel fisher discriminant analysis. The work in [5, 34] extended the label noise model for discriminative classifiers i.e., logistic regression while a kernel version was studied in [7]. The work in [13, 15, 33, 39] also employed models based on the latent variable label noise model in Convolutional Neural Network (CNN) to capture the label noise mechanism.

Among the three major paradigms of label noise handling methods, the model-based approach seems to enjoy superior robustness towards label noise as compared to the former two approaches, (see [5, 15, 21] for empirical evidence), while also being more transparent and highly extensible.

Despite the fact that the classifiers employing the latent variable noise model showed improved robustness towards label noise, there still remains a problem of estimating the label noise proportions when there are not enough training examples. This is particularly true when the classification problem involves either a large number of data classes or high dimensional data or both. It is legitimate to ask why label noise is problematic in small sample size problems where we can simply make an effort to verify all of the labels? Possibly, due to either the lack of knowledge to determine the correct labels or the disagreement between the labelling experts, there indeed exist small sample size classification problems that contain label noise, notably those from the biomedical domain [1, 42]. The difficulty in estimating noise proportion is also partly acknowledged in noise-tolerant deep learning algorithm [39] where they found that estimating the noise parameters from small dataset is still a challenging issue and therefore proposed a regularisation capable of controlling the parameters indicating the proportion of correct labels. Recently, a similar approach akin to that proposed in [39] was employed to constrain the annotator confusion matrix in the context of annotator modelling [40]. However, the added regularisation still relies on hyperparameter tuning which is not straightforward via cross validation. In our experience, the approach tends to overestimate the noise proportions in noisy and small sample size problems.

To improve the accuracy of label noise proportion estimates and to facilitate model selection in the presence of label noise, we propose to impose a sparsity constraint on the label noise parameters. The rationale is to encourage the noise model to focus on classes with greater amount of label confusions while classes with natural overlapping and minor label noise will be given less attention. With the additional constraint, it is expected that the estimation of noise parameters will be more accurate, leading to improved robustness in the cases where training examples are inadequate. Further, we shall treat the sparsity regularisation from a Bayesian perspective where the regularisation parameters are integrated out analytically and can be inferred from the data through the noise model, without the need to perform the time-consuming cross validation. To the best of our knowledge, we are the first to consider employing Bayesian regularisation for expressing preference on the label noise parameters.

The resulting Bayesian label noise model can then be seamlessly incorporated into any probabilistic classifier. To demonstrate its generality, we integrated the resulting noise model into three common probabilistic classification models namely, the logistic regression, the multinomial logistic regression and the neural networks. The empirical results based on both synthetic and real world datasets showed that the classifiers with the Bayesian label noise model better learns the relationship between input features and their true labels when compared to traditional baselines as reflected by the accuracy of noise parameters estimation as well as the classification accuracies. To summarise, we consider the followings to be our main contributions

- We proposed a Bayesian label noise model which is both data and computationally efficient.
- We demonstrated how to integrate the proposed Bayesian label noise model into various probabilistic classifiers.
- We extensively evaluated the resulting label noise robust classifiers on both synthetic datasets as well as various image classification datasets.

The rest of the paper is organised as follows. Section 2 presents the background on the latent variable label noise model. The proposed Bayesian label noise model is formulated in Sect. 3. The integrations of the proposed model with the logistic regression, the multinomial logistic regression and the neural

networks are described in Sect. 4. Experimental studies and discussion of the results are presented in Sect. 5, while Sect. 6 concludes the study and outlines future research directions.

2 Background and problem settings

Let X be an m-dimensional vector space, and $Y = \{1, 2, \ldots, K\}$ be a set of labels. Learning a classification model is the task of inferring a real-valued function $f : X \to Y$, where X and Y represent a set of feature vectors and class labels, using a set of training examples $S = \{\mathbf{x}_n, y_n\}_{n=1}^N$ drawn i.i.d. from some joint distribution $\mathcal{D}_{X\times Y}$. In general, we wish the resulting f to generalise well to unseen examples from the same distribution. However, it is not uncommon that the true label may be corrupted by some external factors. Under this circumstance, the learning algorithm instead has access to a training data with possibly noisy labels $\tilde{S} = \{\mathbf{x}_n, \tilde{y}_n\}_{n=1}^N$, where we use $\tilde{y}$ to denote the observed (and possibly noisy) class label. Clearly the discrepancy between the true label y and the observed label $\tilde{y}$ is problematic in the sense that we are training a classifier based on the training data from $\tilde{\mathcal{D}}_{X\times Y}$ but are expecting it to do well on $\mathcal{D}_{X\times Y}$. The question is how can we learn f using $\tilde{S}$ so that it still performs well on the original noise-free distribution?

One of the principled approaches for dealing with the discrepancy between the true labels and the observed labels is to model the label flipping mechanism explicitly. Following [5], in order to take into account the possibility that the observed label $\tilde{y}$ could be corrupted and that it may differ from the true label, one introduces a latent variable y to represent the true label and rewrites the observed class posterior probability of $\mathbf{x}_n$ as

$$\tilde{P}_n^k := p(\tilde{y}_n = k|\mathbf{x}_n) = \sum_{j=1}^{K} p(\tilde{y}_n = k|y_n = j)p(y_n = j|\mathbf{x}_n) = \sum_{j=1}^{K} \omega_{jk} P_n^j \tag{1}$$

In this way, the observed posterior probability is expressed in terms of a weighted sum of $P_n^j := p(y = j|\mathbf{x}_n)$, the true posterior probabilities. The conditional probability $\omega_{jk} := p(\tilde{y} = k|y = j)$ in Eq. (1) represents the probability that the true label y from class j will flip into the observed class label k, and is used to quantify class-conditional label noise. Hereinafter, we shall refer to this particular label noise model as the *latent variable noise model*. An objective function, e.g., the data log-likelihood, of a probabilistic classifier utilising the latent variable noise model and parameterised by model parameter θ often takes the following form

$$\mathcal{L}(\theta, \Omega) = \sum_{n=1}^{N}\sum_{k=1}^{K} \delta(\tilde{y}_n = k) \log\Big(\sum_{j=1}^{K} p(\tilde{y}_n = k|y_n = j)p(y_n = j|\mathbf{x}_n;\theta)\Big) \tag{2}$$

For a K-class classification task, ω_{jk} form a $K \times K$ label flipping probability matrix, Ω, which needs to be estimated from the data. To incorporate the latent variable noise model into a probabilistic classification model, one replaces the true posterior probabilities with the observed class posterior probabilities defined by the noise model and optimises for the model's parameters and the label noise parameters. The quality of Ω estimate plays a role in the robustness of a probabilistic classifier employing the latent variable noise model. The following lemma, due to [35], quantifies the importance of having an accurate Ω estimate for binary classification problems.

Lemma 1 (Reeves and Kabán [35]) *Let* $\hat{\tilde{P}}^1 : X \to [0, 1]$ *be an estimate of* $\tilde{P}^1$ *and define* $\hat{P}^1 : X \to [0, 1]$ *by* $\hat{P}_n^1 := (\hat{\tilde{P}}_n^1 - \hat{\omega}_{01})/(1 - \hat{\omega}_{01} - \hat{\omega}_{10})$. *Suppose that* $\omega_{01} + \omega_{10} < 1$, *and* $\hat{\omega}_{01}, \hat{\omega}_{10} \in [0, 1)$ *with* $\hat{\omega}_{01} + \hat{\omega}_{10} < 1$. *Suppose further that* $\max\{|\hat{\omega}_{01} - \omega_{01}|, |\hat{\omega}_{10} - \omega_{10}|\} \leq (1 - \omega_{01} - \omega_{10})/4$, *we have*

$$|\hat{P}_n^1 - P_n^1| \leq 8 \cdot \frac{\max\{|\hat{\tilde{P}}_n^1 - \tilde{P}_n^1|, |\hat{\omega}_{01} - \omega_{01}|, |\hat{\omega}_{10} - \omega_{10}|\}}{1 - \omega_{01} - \omega_{10}} \tag{3}$$

The above lemma emphasises the fact that the quality of the true posterior estimate depends on not only the ability of model to fit to the observed labels, i.e., $\hat{\tilde{P}}_n^1 \approx \tilde{P}_n^1$, but also the quality of the noise parameter estimation, as quantified by the deviation of $\hat{\omega}_{jk}$, the estimate, from ω_{jk}, the true one. Therefore it is of great importance to get the estimates as accurately as possible.

Even though the goal of label noise learning is theoretically emphasised but to achieve that might not be straightforward in reality. It is inevitable that without enough training data the resulting parameters could be inaccurate. In such a case, some inductive bias is needed in order to successfully learn the noise model. The Bayesian framework seems to provide an elegant mechanism which makes any assumption on the learning explicit [9, 25]. From the Bayesian viewpoint, the learning problem can be seen as

$$\text{posterior} \propto \text{likelihood} \cdot \text{prior}$$

Here, the prior reflects our belief on the model's parameters and explicitly incorporates such belief in the form of regularisation. In the context of average-case analysis, it also provides optimality guarantee [9]. Motivated by the transparency of the Bayesian framework, we will tackle the noise parameter estimation problem from the Bayesian perspective.

3 Proposed Bayesian label noise model

As mentioned in the previous section, one of the major challenges for the latent variable model is the problem of how to estimate the label flipping matrix accurately, particularly when the training data is scarce. Often, the label flipping matrix is under-estimated, e.g., the estimate is close to the identity matrix, especially when the classification problem also involves large numbers of data classes and high data dimensionality. It should be clear from Eq. (1) that when Ω converges to the identity matrix, $\tilde{P}_n^k$ reduces to P_n^k, and hence a classifier employing the latent variable noise model would behave similarly to its non-robust counterpart. To alleviate the problem, we propose to separately regularise the elements of the label flipping matrix. We would like to encourage the label flipping matrix to be sparse using the L_1 regularisation such that few important elements of the matrix remain non-zeros, while also preventing the label noise flipping matrix to prematurely converge to the identity matrix. We expect the model to remain concentrated on classes with relatively higher percentage of label flipping while minor label confusions as well as natural overlapping of the data classes which manifests as small elements in the matrix should be muted by the regularisers. Without loss of generality, let us consider a problem of learning a robustified probabilistic classification model parameterised by θ and Ω, for which we construct a regularised objective function as

$$\mathcal{R}(\theta, \Omega) = \mathcal{L}(\theta, \Omega) - \sum_{j=1}^{K} \sum_{k=1}^{K} \alpha_{jk} |\omega_{jk}| \tag{4}$$

Here, $\mathcal{L}(\theta, \Omega)$ represents the objective function of the classification model which needs to be maximised, and $\sum_{j=1}^{K} \sum_{k=1}^{K} \alpha_{jk} |\omega_{jk}|$ is the proposed regularisation term. The optimisation criterion in Eq. (4) depends on a set of regularisation parameters α_{jk}, which are traditionally chosen by means of cross validation. However, we argue that choosing the regularisation parameters using standard cross validation is not straight-forward in the presence of labelling errors because the validation set may also contain unreliable validation labels. Furthermore, performing cross validation is also computationally prohibitive as we need to cross validate for K^2 regularisation parameters. Therefore, we propose to employ a Bayesian approach to determine the good value of α_{jk} automatically. We believe that by determining the regularisation parameters simultaneously with learning the model's parameters, through the noise model, we should be able to avoid selecting sub-optimal models while reducing the additional computational burden incurred by the cross validation process.

From the Bayesian perspective, the regularised objective function in Eq. (4) has the following Bayesian interpretation.

$$p(\Omega | S, \theta, \alpha_{j=1:K,k=1:K}) \propto p(S, \theta | \Omega) \prod_{j=1}^{K} \prod_{k=1}^{K} p(\omega_{jk} | \alpha_{jk}) \tag{5}$$

The first term on the r.h.s corresponds to the objective function when the training data S and the model's parameter θ are fixed. Note that we can drop the absolute function from ω_{jk} because we will posit a distribution with positive real support on it. Now, taking the logarithm of Eq. (5) yields,

$$\begin{aligned} \log p(\Omega | S, \theta, \alpha_{j=1:K,k=1:K}) &= \log p(S, \theta | \Omega) \\ &+ \sum_{j=1}^{K} \sum_{k=1}^{K} \log p(\omega_{jk} | \alpha_{jk}) \\ &+ const. \end{aligned} \tag{6}$$

From the above, we see that ω_{jk} is conditioned on the regularisation parameter α_{jk}. Bayesian regularisation attempts to eliminate such dependency and to find the marginal prior probability of ω_{jk} by integrating out α_{jk}.

$$p(\omega_{jk}) = \int_0^{\infty} p(\omega_{jk} | \alpha_{jk}) p(\alpha_{jk}) d\alpha_{jk} \tag{7}$$

To ensure the positivity of ω_{jk}, we posited an exponential distribution on the conditional probability $p(\omega_{jk}|\alpha_{jk})$,

$$p(\omega_{jk} | \alpha_{jk}) = \alpha_{jk} e^{-\alpha_{jk} \omega_{jk}} \tag{8}$$

The non-negative prior probability for α_{jk} can also be modelled using the exponential distribution,

$$p(\alpha_{jk}) = \beta e^{-\beta \alpha_{jk}} \tag{9}$$

The value of β reflects our prior belief on the value of the regularisation parameters. Usually, a small value of β is employed for the non-informative nature of the hyper-prior. To this end, we completed the integration using the Gamma integral, $\int_0^{\infty} x^{\nu-1} e^{-\mu x} dx = \frac{\Gamma(\nu)}{\mu^{\nu}}$ and obtained the marginal prior distribution on ω_{jk} as

$$p(\omega_{jk}) = \beta \frac{\Gamma(2)}{(\omega_{jk} + \beta)^2} \tag{10}$$

where $\Gamma(\cdot)$ is the Gamma function. The regularisation term in Eq. (4) is then equivalent to the negative log of the resulting marginal prior up to some constant. By taking the derivative of the negative log of the marginal prior in Eq. (10) w.r.t ω_{jk}, which gives

$$\frac{\partial - \log p(\omega_{jk})}{\partial \omega_{jk}} = \frac{2}{(\omega_{jk} + \beta)} \frac{\partial \omega_{jk}}{\partial \omega_{jk}} \tag{11}$$

we find that it is of the same form as the derivative of Eq. (4) w.r.t ω_{jk} except the preceding term, from which we can read-off the estimate of the regularisation term as,

$$\alpha_{jk} = \frac{2}{(\omega_{jk} + \beta)} \tag{12}$$

It can be understood from the above update equation that the penalty α_{jk} will be larger for the smaller ω_{jk}, which will consequently drive ω_{jk} towards an even smaller value.

4 Integrations with major probabilistic classification models

We will now demonstrate how the resulting Bayesian label noise model can be readily incorporated into major probabilistic classification models.

4.1 Logistic regression

Assuming $y \in \{0, 1\}$, the standard Logistic Regression (LR) aims at maximising the data log-likelihood of the form,

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} (1 - y_n) \log P_n^0 + y_n \log P_n^1 \tag{13}$$

where P_n^1 is modelled using the sigmoid function $\frac{1}{1+\exp(-\mathbf{w}^T \mathbf{x}_n)}$, and $P_n^0 = 1 - P_n^1$. The relationship between P_n^0 and $\tilde{P}_n^0$ can be established via the Bayesian label noise model yielding the robustified objective of the LR model.

$$\mathcal{R}(\mathbf{w}, \Omega) = \sum_{n=1}^{N} (1 - \tilde{y}_n) \log \tilde{P}_n^0 + \tilde{y}_n \log \tilde{P}_n^1 - \sum_{j=0}^{1} \sum_{k=0}^{1} \alpha_{jk} \omega_{jk} \tag{14}$$

Hereinafter we shall refer to the resulting LR model as *Bayesian robust logistic regression* (brLR). The last term is again the regularisation on the label noise parameters. Learning brLR can be done using gradient-based method where we find the gradient of the modified objective function w.r.t $\mathbf{w}$ as

$$\nabla_{\mathbf{w}} \mathcal{R}(\mathbf{w}, \Omega) = \sum_{n=1}^{N} \Big(\frac{\tilde{y}_n(\omega_{11} - \omega_{01})}{\tilde{P}_n^1} + \frac{(1 - \tilde{y}_n)(\omega_{10} - \omega_{00})}{\tilde{P}_n^0} \Big) P_n^1 P_n^0 \mathbf{x}_n \tag{15}$$

The updates for noise parameters can be done by the multiplicative fixed point updates. The detailed derivations are given in Appendix 1.

$$\omega_{00} = \frac{\omega_{00} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \Big)}{\omega_{00} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \Big) + \omega_{01} \Big(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \Big)} \tag{16}$$

$$\omega_{01} = \frac{\omega_{01} \Big(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \Big)}{\omega_{00} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \Big) + \omega_{01} \Big(\sum_{n=1}^{N} (\tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \Big)} \tag{17}$$

$$\omega_{10} = \frac{\omega_{10} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \Big)}{\omega_{10} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \Big) + \omega_{11} \Big(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \Big)} \tag{18}$$

$$\omega_{11} = \frac{\omega_{11} \Big(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \Big)}{\omega_{10} \Big(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \Big) + \omega_{11} \Big(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \Big)} \tag{19}$$

To this end, we summarise the steps to learn the Bayesian robust Logistic Regression in Algorithm 1.

Algorithm 1: Bayesian robust Logistic Regression (brLR)

Input: Corrupted training data $(\mathbf{x}_n, \tilde{y}_n)_{n=1}^N$
Initialise $\mathbf{w}$, Ω, α_{jk}
while $iter < maxIter$ **do**
Update $\mathbf{w}$ using the gradient in Eq.(15)
Update Ω using the multiplicative updates Eq.(16)-(19)
Update α_{jk} using Eq.(12)
end
Output: Estimated $\hat{\mathbf{w}}, \hat{\Omega}$

4.2 Multinomial logistic regression

In the case of multinomial Logistic Regression (mLR), the data log-likelihood assuming $y \in \{1, K\}$ is defined as

$$\mathcal{L}(\mathbf{W}) = \sum_{n=1}^{N}\sum_{i=1}^{K} \delta(y_n = i) \log \tilde{P}_n^i \tag{20}$$

Here, $\mathbf{W}$ is a row-major matrix with row $\mathbf{w}_k$ representing the weight vector for the posterior probability $P_n^k := \frac{\exp(\mathbf{w}_k^T \mathbf{x}_n)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x}_n)}$ which is modelled using the softmax function. In similar vein to brLR, we can robustify the mLR to obtain the Bayesian robust multinomial Logistic Regression (brmLR) with the new objective function as,

$$\mathcal{R}(\mathbf{W}, \Omega) = \sum_{n=1}^{N}\sum_{i=1}^{K} \delta(\tilde{y}_n = i) \log \tilde{P}_n^i - \sum_{j=1}^{K}\sum_{k=1}^{K} \alpha_{jk}\omega_{jk} \tag{21}$$

Again, a gradient-based method can be used to optimise the weight vectors. The gradient of the objective w.r.t $\mathbf{w}_k$ is

$$\nabla_{\mathbf{w_k}} \mathcal{R}(\mathbf{W}, \Omega) = \sum_{n=1}^{N}\sum_{i=1}^{K} \delta(\tilde{y}_n = i) \frac{\sum_{j=1}^{K} (\omega_{ki} - \omega_{ji}) P_n^k P_n^j}{\tilde{P}_n^i} \mathbf{x}_n \tag{22}$$

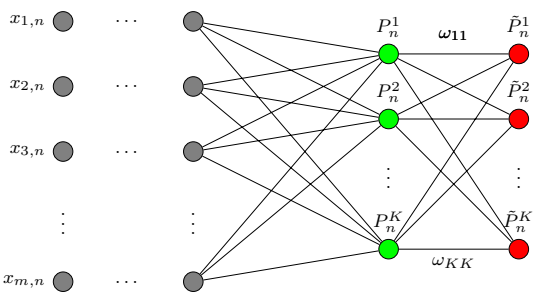


Fig. 1 The neural network with the label noise absorbing layer added on top

The multiplicative update equations for ω_{jk} can also be derived similarly to the case of brLR yielding,

$$\omega_{jk} = \frac{\omega_{jk} \sum_{n=1}^{N} \delta(\tilde{y}_n = k) \frac{P_n^j}{\tilde{P}_n^k} - \alpha_{jk}}{\sum_{j=1}^{K} \left(\omega_{jk} \sum_{n=1}^{N} \delta(\tilde{y}_n = k) \frac{P_n^j}{\tilde{P}_n^k} - \alpha_{jk}\right)} \tag{23}$$

The algorithm for learning the brmLR is summarised in Algorithm 2.

Algorithm 2: Bayesian robust multinomial Logistic Regression (brmLR)

Input: Corrupted training data $(\mathbf{x}_n, \tilde{y}_n)_{n=1}^N$
Initialise $\mathbf{w}_{k=1..K}$, Ω, α_{jk}
while $iter < maxIter$ **do**
Update $\mathbf{w}_k$ using the gradient in Eq.(22)
Update Ω using the multiplicative updates Eq.(23)
Update α_{jk} using Eq.(12)
end
Output: Estimated $\hat{\mathbf{w}}_{k=1..K}, \hat{\Omega}$

4.3 Neural network

The proposed Bayesian label noise model can also be incorporated into neural network classifiers. Recall that a typical multi-layer neural network for classification employs the softmax activation function at the output layer. For a K-classes classification problem there will be K of such nodes, each outputting the class posterior probability. Let us denote the output of the k-th node upon receiving an input vector $\mathbf{x}_n$ as $\tilde{P}_n^k := p(\tilde{y}_n = k|\mathbf{x}_n;\mathbf{W})$, where $\mathbf{W}$ represents the parameters of the network.

Training the neural network is essentially the task of minimising of mismatch between the observed label $\tilde{y}_n$, usually encoded using one-hot format, and the outputs from the output nodes, $\tilde{P}_n^k$. The mismatch is often quantified by the so-called cross-entropy loss, $L(\mathbf{W}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \delta(\tilde{y}_n = k) \log(P_n^k)$. To account for the possibility that the observed labels can be noisy, we can employ the Bayesian label noise model and rewrite the penalised objective function as

$$\mathcal{R}(\mathbf{W}, \Omega) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \delta(\tilde{y}_n = \mathbf{e}^k) \log \sum_{j=1}^{K} \omega_{jk} P_n^j + \sum_{j=1}^{K}\sum_{k=1}^{K} \alpha_{jk}\omega_{jk} \tag{24}$$

Table 1 The characteristics of the image classification datasets employed

Dataset	# instances	Class distribution	Image size
CIFAR-2	10,000	Airplane = 5000, truck = 5000	32 x 32
CIFAR-3	15,000	Cat = 5000, dog = 5000, deer = 5000	32 x 32
CIFAR-10	60,000	Airplane, automobile, bird, cat, deer, dog, frog, Horse, ship, truck , each with 5000 instances	32 x 32
UEC-5	606	Eels on rice = 130, chicken and egg on rice = 121, croissant = 120, beef noodle = 139, Japanese-style pancake = 137	32 x 32
UEC-10	1368	Beef noodle = 139, takoyaki = 134, pizza = 134, Spaghetti = 151, udon noodle = 152, gratin = 115, Fried noodle = 131, tensin noodle = 112, Japanese-style pancake = 137, sandwiches = 163	32 x 32

where $\mathbf{e}^k$ denotes a zero vector of length K with value 1 only at the k-th position. In this manner, the model aims at explaining the observed labels through the current knowledge of the true posterior probabilities and the label flipping probabilities. Structurally, this is equivalent to augmenting the existing network with a linear layer with some constraints on the weights. The outputs from the penultimate layer can be viewed as the *true* class posterior probabilities while the weights of the links from the penultimate nodes to the output nodes are essentially representing the label flipping probabilities. Figure 1 illustrates the resulting architecture.

To learn the model, a gradient-based approach such as the back-propagation algorithm can be employed in minimising the modified cross-entropy loss in Eq. (24), subject to $0 \leq \omega_{ij} \leq 1$ and $\sum_j \omega_{ij} = 1$ constraints. In addition, the value of the regularisation parameters α_{ij} are to be continuously updated according to Eq. (12). To classify a new data point $\mathbf{x}_q$, we read off the outputs from the penultimate layer and decide $\hat{y}_q = \arg\max_k P_q^k$.

5 Empirical evaluations

Comprehensive experiments are designed to investigate the benefits of having the regularisation on the label noise parameters for the robust probabilistic classifiers introduced above.

5.1 Experimental protocol

The experiment is divided into two parts. The first part (Sect. 5.2) concerns the behaviour of the regularised noise model in a controlled environment where the data is synthetically generated with varying number of training examples, data dimensionality and the number of data classes. For this purpose, we shall employed the proposed multi-class brmLR and shall compare its performance with those from the standard multinomial logistic regression (mLR) and robust multinomial logistic regression (rmLR) i.e., a mLR employing the latent variable noise model but without the regularisation on the noise parameters. The results from this set of experiments should enable us to conclude whether or not the added Bayesian regularisation helps the model to better estimate the noise proportions and when would the improvement be more apparent.

The second part of the experiment first concerns the performance of the proposed noise model on real datasets with artificial label noises. We shall be testing various model-based label noise approaches which act as the classification layer on top of 1) a custom convolutional neural network (Sect. 5.3), and 2) the state-of-the-art deep neural networks (Sect. 5.4) on some of the well-known image classification tasks. Specifically, our testbeds are based on real world image datasets including a 5-class and a 10-class subsets of food image recognition data [30] named UEC-5 and UEC-10, CIFAR-10 [20] and its 2-class and a 3-class subsets named CIFAR-2 and CIFAR-3. The detailed characteristics of the datasets are summarised in Table 1. Additionally, we also employed two datasets from the biomedical domain which genuinely contain label noise, namely colon cancer [1] and breast cancer [42] dataset. The experiment with real label noise is presented in Sect. 5.5.

The custom convolutional neural network that we employed in Sect. 5.3 is composed of 6 convolutional layers divided into 3 sets. The first set is made up of two convolutional layers each employing 32 kernels of size (3,3). The second set consists of two layers with 64 (3,3)-kernels. Two convolutional layers with 128 kernels of size (3,3) terminate the feature learning part of the network. Each set of convolutional layers is followed by a max pooling layer. All activation functions at the convolutional layers are the Rectified Linear Unit (ReLU). The fully connected layers which is responsible for classifying the extracted features is a two-layer fully-connected neural network employing 64 hidden

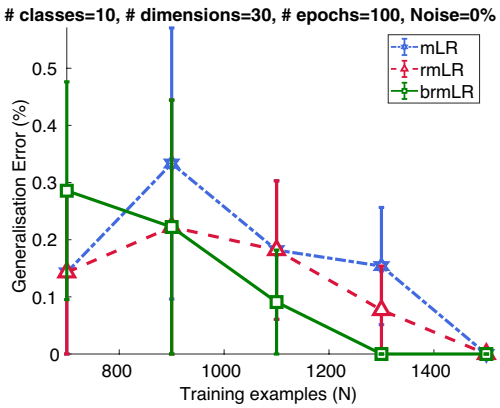
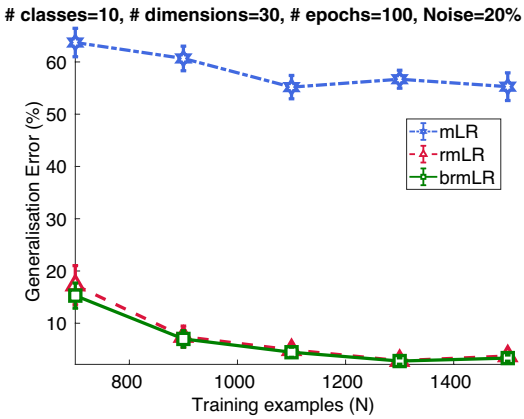
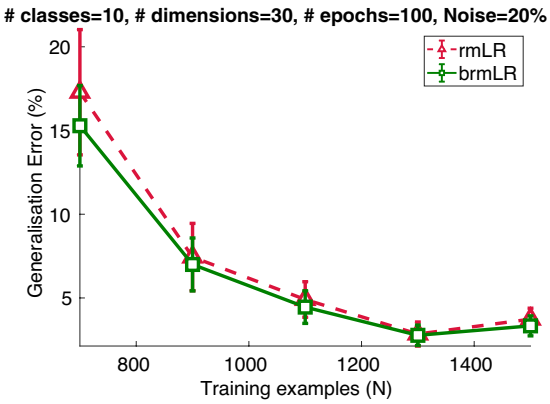


Fig. 2 The generalisation errors (percent) of the three multinomial Logistic Regression models on label-noise-free data as the number of training examples vary from $N = 700$ to 1500. Data dimensionality and data classes were fixed to $m = 30$ and $K = 10$, respectively

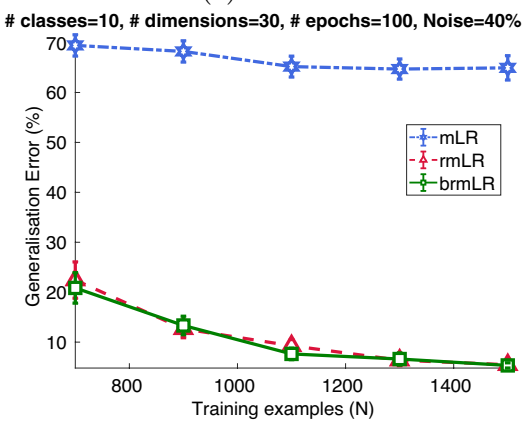
nodes with sigmoid activation functions in the first layer, while the number of nodes in the ultimate layer is set to K, the number of data classes. The activation function of the output layer is the softmax function. For the CNN employing the proposed Bayesian label noise model, a noise absorbing layer with K nodes is added on top of the existing architecture. The activation functions for the nodes in the extra layer are linear functions. All hyperparameters of the custom CNN were chosen based on clean data. We performed a grid search for batch size in $\{16, 32, 64, 128\}$ and for learning rate in $\{0.1, 0.01, 0.001\}$, from which the best settings were found to be 32 and 0.01, respectively. The learning rate decay, however, was empirically set to 3×10^{-4}. Since all the methods to be compared share the same convolutional layers and only differ in the classification layers, we applied the selected hyperparameters for all of them. In all experiments, we trained the networks for 100 epochs.



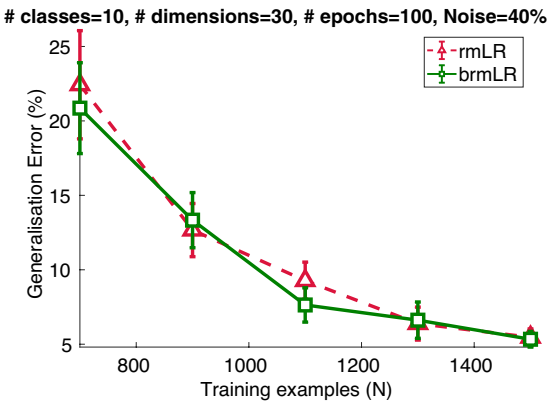
(a) 20% noise



(b) 20% noise (zoomed in)



(c) 40% noise



(d) 40% noise (zoomed in)

Fig. 3 The generalisation errors (percent) of the three multinomial Logistic Regression models as the number of training examples vary from $N = 700$ to 1500. Data dimensionality and data classes were fixed to $m = 30$ and $K = 10$, respectively

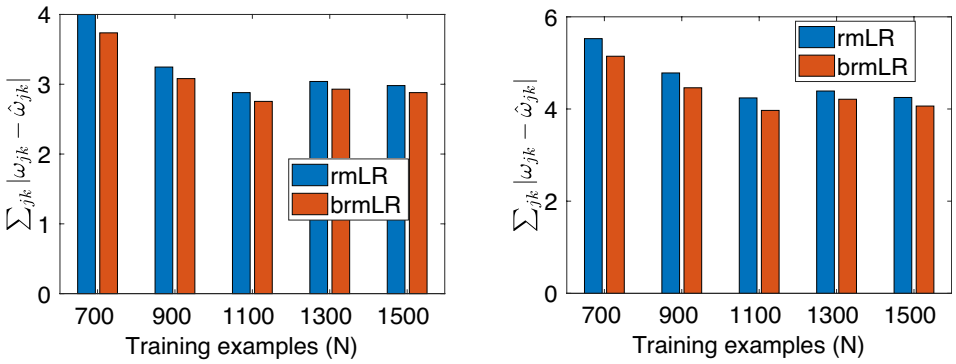


Fig. 4 Estimation errors for the label noise matrix Ω as the number of training examples varied

(a) 20% noise (b) 40% noise

5.2 Results: synthetic data

We start by presenting the classification performance on synthetics datasets in which we varied the number of classes, K, the dimensionality of the data, m and the number of training examples, N. We followed the holdout method and split the data using 80/20 train/test ratio. In all cases, we artificially injected label noise specified by simulated Ω into the training set while labels in the test set remain intact. The label flipping matrix is generated using the following steps. First, we sampled its entries from the standard normal distribution, i.e., $\omega_{jk} \sim \mathcal{N}(0, 1)$. Next, we disregarded the negative elements, and added some positive integer to ω_{ii} for all $i \in [1, K]$ so that the diagonal of Ω remains the largest entries and that the noise rate, taken to be ratio of the sum of the off-diagonal elements and the number of classes, matches the desired rate. Finally, we normalise each row such that it sums to unity. We consider three cases of label noise settings: no noise, 20% noise rate and 40% noise rate. We then reported the test errors averaged over 10 random repetitions of the above procedure.

We will first inspect the behaviour of the three variations of multinomial Logistic Regression models as the number of available training examples vary. Figure 2 illustrates the relative performances of the three classifiers trained on clean labels. The three models achieved very low error rates of about 0.3% and the error rates improved as more training data became available. However, when label noises were injected into the dataset, we can see that the noises affect the standard mLR as can be seen from Fig. 3a, c, and that the benefit of explicitly modelling label noise is evident. We also present close-up plots with mLR removed in Fig. 3b, d to further illustrate the advantage of the proposed Bayesian regularisation on the unregularised rmLR. The results suggest that brmLR tends to be more robust than rmLR when there's a limited number of data available for training, e.g., $N = 700$. Nonetheless, when more data is available the gaps start to diminish, which is somewhat expected. The reason as to why brmLR showed better generalisation performance could be due to the fact that its noise proportion estimates are more accurate than those by rmLR.

To further elucidate our point, we measured the errors of the Ω estimates given by brmLR and rmLR and plotted the sum of element-wise differences between the true label flipping matrix and its estimate, i.e., $\sum_{j=1,k=1}^{K} |\omega_{jk} - \hat{\omega}_{jk}|$, in Fig. 4. Interestingly, the estimates produced by brmLR were generally better than those by rmLR in all cases. We also see that when more examples are available, the Ω estimates become more accurate as can be observed from Fig. 4. This provides evidence supporting our speculation on the data efficiency of the proposed Bayesian label noise model as compared to the unregularised noise model.

Let us further investigate the effect of data dimension on the generalisation performances of the three models. Figure 5 demonstrates first the performances in a noise-free setting. The results match our general understanding in the sense that data becomes more linearly separable as the dimensionality increases, and we see that the errors of all models consequently decreased. As label noises were

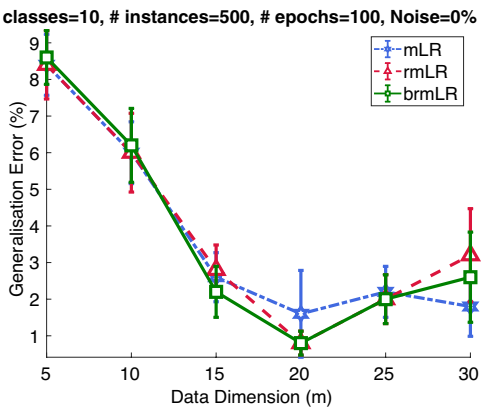
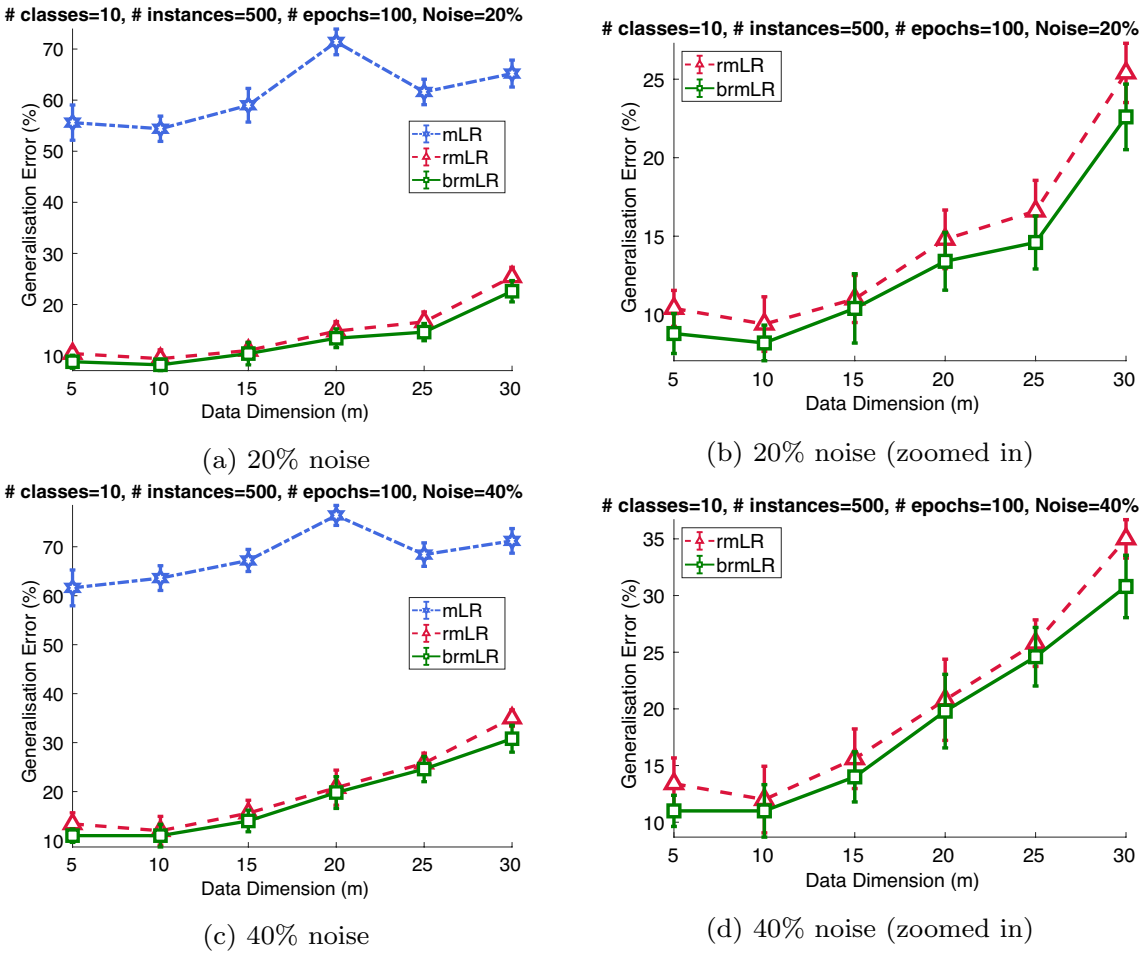


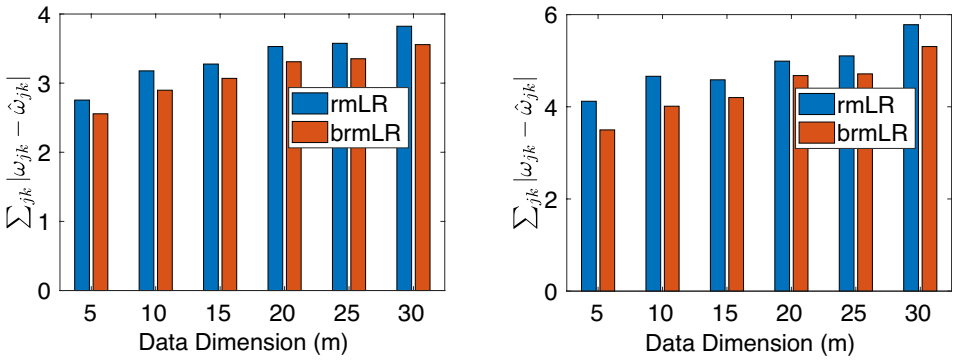
Fig. 5 The generalisation errors (percent) of the three multinomial Logistic Regression models on noise-free data as the number of data dimensions vary from $m = 5$ to 30. Training examples and data classes were fixed to $N = 500$ and $K = 10$, respectively



(a) 20% noise (b) 20% noise (zoomed in)

(c) 40% noise (d) 40% noise (zoomed in)

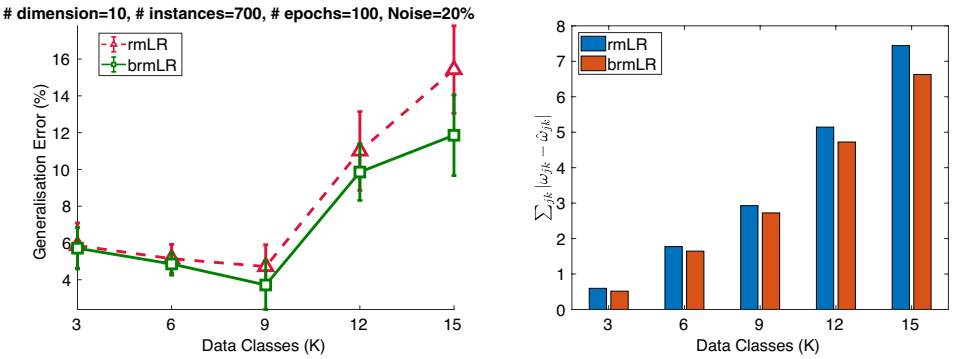
Fig. 6 The generalisation errors (percent) of the three multinomial Logistic Regression models as the number of data dimensions vary from $m = 5$ to 30. Training examples and data classes were fixed to $N = 500$ and $K = 10$, respectively



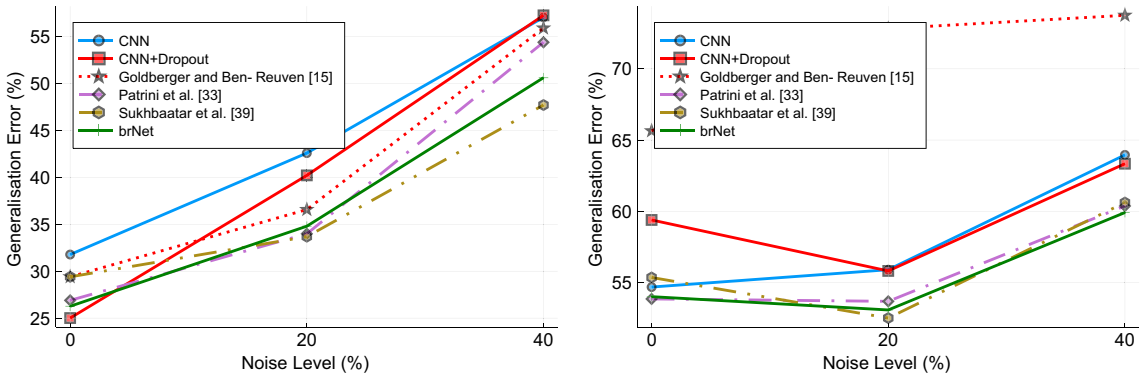
(a) 20% noise (b) 40% noise

Fig. 7 Estimation errors for the label noise matrix Ω as data dimension varied

introduced into the data, we witnessed quite the opposite behaviours from mLR, rmLR and brmLR as m increases. We see from Fig. 6 that the errors also increased with data dimensions. It could be understood that the data becomes more linearly separable in high dimensional space with respect to more than one label configurations including the noisy ones. Therefore, it is easier to get fooled by the noisy labels. Despite the detrimental effect of the data dimensionality on the classification performance, we noticed that brmLR is still more robust than rmLR, and especially so in higher dimensional spaces.



(a) 20% noise (b) 20% noise



(a) UEC-5 (b) UEC-10

Fig. 8 The generalisation errors (percent) of the three multinomial Logistic Regression models as the number of data classes varied from $k = 3$ to 15. Training examples and data classes were fixed to $N = 700$ and $K = 10$, respectively

Fig. 9 Averaged classification errors from brNet over 5 random repetitions compared to its peers on data with label noise at 0%, 20% and 40%

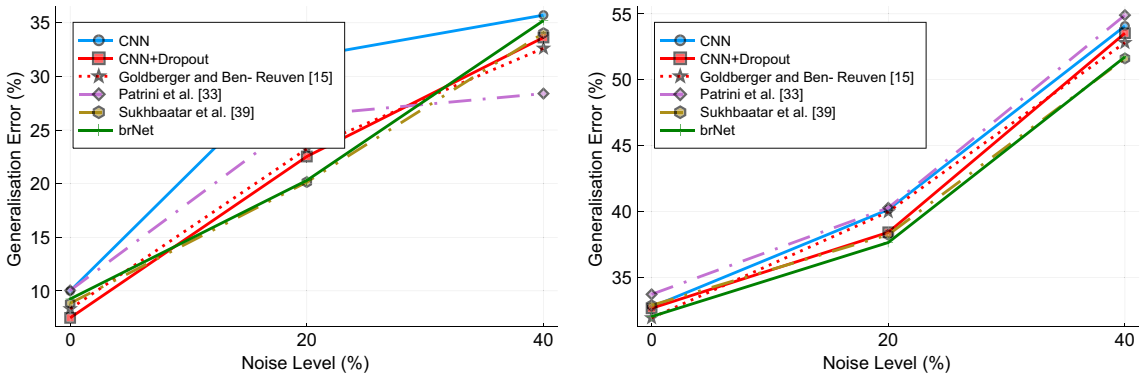
The observed empirical superiority of brmLR stemmed from the fact that brmLR better estimated the noise proportion leading to reduced classification errors. Figure 7 summarised the quality of Ω estimates from brmLR and rmLR.

The last factor which could affect the quality of Ω estimate is the number of data classes. This is because, inevitably, there are more free parameters to be estimated. To investigate this factor, we fixed the number of training examples to $N = 700$, data dimensionality to $m = 30$ and varied the number of data classes. We present our findings in Fig. 8a, b (omitting the 40% noise case for brevity). We observe that as the classification problem involves more data classes it becomes more and more difficult to classify, as can be seen from the increased error rates. Likewise, the estimation of Ω also becomes more difficult. It can be seen that brmLR and rmLR were indistinguishable when there's seemingly enough training data relative to the number of data classes, e.g., 5-class problem. However, as the number of data classes increases while the number of training examples remains fixed, the advantage of brmLR becomes more apparent as reflected by the lower generalisation errors as well as the higher quality of Ω estimate.

From the above set of experiments, we can certainly see that having the Bayesian noise parameter regularisation is indeed beneficial for the robust classifier employing the latent label noise model. With the regularisation, we can expect more accurate noise proportion estimation, which will eventually lead to improved classification performance, especially in the cases where training examples are limited.

5.3 Results: real-world data

Having seen that the proposed Bayesian noise model outperformed the unregularised latent variable noise model in the simulated problems, we shall now switch to more challenging real-world classification testbeds.



(a) CIFAR-2 (b) CIFAR-3

Fig. 10 Averaged classification errors of brNet over 5 random repetitions compared to its peers on data with label noise at 0%, 20% and 40%

Table 2 Average ranks of the six variants of the custom CNN models

CNN	CNN+Dropout	Goldberger et al. [15]	Patrini et al.[33]	Sukhbaatar et al. [39]	brNet
5.08	3.50	3.91	3.66	2.50	**2.33**

Boldface indicates the best ranking averaged over 12 noise settings

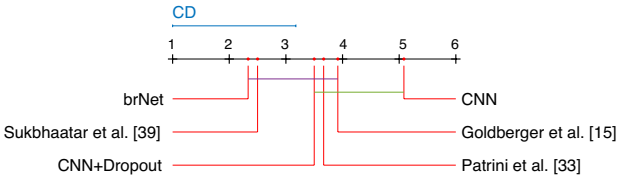


Fig. 11 Graphical representation of the Nemenyi post-hoc test. The critical value of the Nemenyi post-hoc test was found to be 2.18. The x-axis corresponds to the ranking scale with the best rank starts from the left hand side

We will study comparative performances of the custom CNN employing the Bayesian noise model named *brNet* and three robust methods for CNN, all utilising the latent variable noise modelling. They are 1) the one without the Ω regularisation by Patrini et al. [33], 2) the one with Ω's trace regularisation by Sukhbaatar et al. [39] and 3) the one using non-linear noise adaptation layer and without Ω regularisation by Goldberger and Ben-Reuven [15]. Included for reference are the performances of the custom CNN without label noise modelling, and the custom CNN utilising dropout regularisation as per [36] at the classification layer. For CNN+Dropout, we set aside a subset of the training data for validation purpose (which possibly contains noisy labels) and perform 5-fold cross validation on $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ for the dropout parameter in every experiment repetition.

We first present the generalisation performances of the six models on UEC-5 and UEC-10 datasets with varying degree of label contaminations from noise-free setting to 20% and 40% noises. The label noises were generated using the same procedure as described above. The challenge for these two datasets is that there are a relatively limited number of training examples in each class. As such, a noise model which employs some kind of noise parameter regularisation is expected to exhibit superior performance.

Figure 9 summarises the averaged test errors over 5 random train/test splits at 0%, 20% and 40% noise rates. Firstly, we observe that brNet and [39]'s approach both of which utilise some kind of Ω regularisation, were more robust to label noise than the rest of the group on both UEC-5 and UEC-10 datasets. Rather surprisingly, brNet also ranked among the first in the noise-free case. We speculate that the original dataset might contain some small amount of label noise and therefore all of the robust models achieved lower error rates as compared to the vanilla CNN. As a higher ratio of label noise was introduced into the training data, the generalisation performances of all models deteriorated. All robust approaches including CNN+Dropout tended to perform relatively well compared to the vanilla CNN at 20% noise on UEC-5, with the exception of the non-linear noise model [15] on UEC-10. We conjecture that its poor performance could be due to the slightly higher sample complexity of the model, and that more training data is required in order to reach its full potential in this 10-class dataset. Nonetheless, looking at the error rates of all methods, we can tell that the UEC-10 dataset is quite a challenging dataset. The method by [33], which does not employ noise parameter regularisation, did not perform as well as the one which imposes some kind of constraint on the label noise matrix. e.g., brNet and [39]. Indeed, they also noted in [33] that the quality of noise estimation for their approach is subjected to the availability of training data. The brNet seemed

Table 3 Classification accuracies of VGG, DenseNet and MobileNet with and without the proposed noise modelling on CIFAR-10

Base CNN model	Classification layer Without noise model	Classification layer With noise model	p-value
VGG	83.64 ± 0.51	83.63 ± 0.31	1.000
DenseNet	75.15 ± 1.27	**82.29 ± 0.89**	0.008
MobileNet	75.44 ± 0.76	75.28 ± 0.88	0.690

Boldface entry indicates that the difference is statistically significant at 0.01 level

No artificial label noise was injected into the dataset. Included for reference were the p value from the Wilcoxon rank-sum test

to compare favourably to [39]. The difference between the two approaches lies in the prior assumption on the label noise parameters. The proposed brNet aims for sparse label noise matrix whereas [39] tries to prevent the noise matrix to undesirably converge to the identity matrix (which translates to believing that there is no noise in the dataset). In our experience, constraining the trace of Ω by the method in [39] often results in overestimated noise proportions. For this reason, we can see that the method in [39] was generally better than brNet when the noise percentage is high, e.g., 40% (Fig. 10).

Let us next analyse the results from CIFAR-2 and CIFAR-3 datasets where there are plenty of training examples available, i.e., 5000 instances per class. First, it is quite clear that the vanilla CNN is still vulnerable to label noise at all levels. Dropout regularisation seems to help counteracting label noise effectively in these cases. We postulate that in the abundance of validation examples, even when some of them are mislabelled, overfitting might be less severe at the model selection level. And that the cross validated dropout parameter reflects the true one in a noise-free setting. Interestingly, when there are only a small number of classes and plenty of training data, [15] showed improved performance as compared to its peers. The unregularised model by [33] was not much more robust than the vanilla CNN on CIFAR-3, and only slightly better on CIFAR-2, except at 40% noise rate. The proposed brNet seemed stable on all cases except at 40% on CIFAR-2.

To summarise, we ranked all six models ($k = 6$) according to the test errors in twelve test cases ($N = 12$) (4 datasets, each with 3 noise configurations) and presented their mean ranks in Table 2. The Friedman test [11] provided the evidence that the differences between the mean ranks of the algorithms are statistically significant with p-value = 0.0038. Therefore, we can first reject the null-hypothesis which assumes the performances of all algorithms are the same. The Nemenyi post-hoc test was then employed to analyse the differences between the brNet and the existing algorithms. The critical difference (CD) at $\alpha = 0.05$ is found to be $2.85 \cdot \sqrt{(6 \cdot (6+1))/(6 \cdot 12)} = 2.18$. The graphical representation of CD together with the mean ranks of all algorithms is depicted in Fig. 11. Firstly, we observe that the brNet is significantly better than the vanilla CNN since their ranks differ more than the CD, $(2.33 + 2.18 < 5.08)$. Compared to the existing robust approaches, the brNet reveals its tendency to perform better than its peers across the board. Though, the post-hoc test might not be powerful enough to detect the statistical significance in their differences. Nonetheless, by considering the mean ranks alone the model which constrains the trace of Ω [39] is still worse than brNet. The dropout regularisation method also performed surprisingly well and ranked third among all algorithms. Meanwhile, although the method by [33] and [15] were effective against label noise as compared to the standard non-robust CNN, they were still lagging behind the methods which employ label noise parameters regularisation. From this evidence, it is rather convincing that being able to estimate label noise proportions accurately is crucial, and that employing a regularisation on noise parameters can effectively improve noise proportion estimation and hence improving model's robustness towards labelling errors, especially in the case where training examples are limited.

5.4 Results: idealised setting with deeper networks

Label noise modelling can be seen as a form of regularisation that prevents the classification model from overfitting to noisy labels. One may wonder how the proposed noise model would behave in an ideal learning scenario where there are an abundance of training examples and none of the labels is noisy. A good label noise model should be able to distinguish the natural class overlapping from label noise. To demonstrate this particular aspect of the proposed method, we compared the predictive performance of CNN with and without the proposed noise model on the CIFAR-10 dataset. Since the dataset is quite complex, we switched from the generic CNN to more advanced forms of CNN, namely VGG [37], DenseNet [17] and MobileNet [16].[1] The classification layers configuration is identical to that in the previous experiments. All the models also share the same hyperparameters which were empirically tuned based on the VGG model. Specifically, we used learning rate $= 10^{-3}$, learning rate decay $= 10^{-6}$, momentum $= 0.9$ and batch size $= 256$ and trained the networks for 100 epochs. We also adopted data augmentation for simulating the unlimited number of training examples. We note that the selected hyperparameters might not be optimal for DenseNet and MobileNet. However, our goal is not about comparing the base CNNs

[1] We used the default implementation of VGG16, DenseNet121 and MobileNet from Keras library.

Table 4 Detail of the two datasets and the classification accuracies of rNDA, depuration and brLR on Colon cancer and Breast cancer datasets

Dataset	# pos.	# neg.	# genes	Model		
				rNDA	Depuration	brLR
Colon cancer	40 (5)	22 (4)	2000	59.68	64.52	**96.77**
Breast cancer	25 (4)	24 (5)	7129	69.39	71.43	**85.71**

The numbers in the parentheses indicate the number of mislabelled instances in the respective classes. Boldface entries indicate the best performances

but rather to observe the difference between the CNNs with and without noise modelling. As such, we believe that the effects of hyperparameters are minimal. The average test accuracies and their standard deviations over 5 random train (80%)/test (20%) splits are summarised in Table 3

The majority of the results did not deviate much from our expectations. We observed that the CNNs with noise model achieved marginally lower accuracy, albeit not statistically significant, with a p-value larger than what normally accepted. The drop in accuracy could be due to the added regularisation. It is also understandable that MobileNet, which is a rather scaled down network for mobile devices, obtained slightly lower predictive performance as compared to the VGG. Surprisingly, in the case of DenseNet we see that the label noise model helps improve the accuracy from 75.15 to 82.29. This is quite unexpected. We speculate that this might be due to two reasons: (1) we did not employ any further regularisation e.g., dropout, and so DenseNet could suffer from overfitting, and (2) the inherent regularisation effect of the noise model may help reduce such overfitting. In fact, when analysing the training accuracies, we found that the training accuracy for DenseNet with noise model was found to be lower than that of the vanilla DenseNet. Overall, it seems that the proposed label noise model did not harm the predictive performance in the idealised setting where there is enough training data and all the labels are supposedly perfect.

5.5 Results: biomedical datasets with genuine label noise

So far, we conducted the experiment with artificial label noise. In this section, we shall demonstrate the superiority of the proposed method in the case where the data genuinely contains label noise. We employed two datasets from the biomedical domain which are not only small in size but also high dimensional, namely, colon cancer [1] and breast cancer [42] dataset. According to the literature some of the examples in both datasets were mislabelled with biological evidence. Since these are binary classification problems, we will employ brLR in this experiment. We compared brLR with two classical methods namely the model-based robust normal discriminant analysis (rNDA) [21] and the nearest-neighbour based data imputation method named *Depuration* [2]. We followed the leave-one-out cross validation method, well used in the literature for small sample size problems, for evaluating the three models. Hyperparameters for depuration, namely the number of neighbours and the relabelling threshold, were tuned using 5-fold cross validation from a set $\{3, 5, 7, 9, 11\}$, which is sensible for a dataset of this size. We summarised the average test accuracies in Table 4.

It is quite clear from the results that brLR was superior to its counterparts in both cases. The method of depuration and rNDA did poorly on colon cancer and lagged behind brLR in breast cancer. The improved classification performance could be due to the more accurate noise proportion estimation. The proposed brNet was able to better estimate the noise rates, for example in breast cancer the correct noise rate for the positive class and the negative class were $4/25 = 0.1600$ and $5/24 = 0.2083$, respectively. We found that the estimates from brNet were 0.1979 and 0.2058, as opposed to 0.000 and 0.0800 by rNDA. The differences in the noise estimates as well as the rigidity of the generative model did not seem to work well in this case. The similar behaviour can be observed in the case of colon cancer dataset. Unfortunately, depuration was unable to provide the noise estimates and this could be seen as a limitation, in terms of transparency and interpretability, of the imputation approach as compared to the model-based methods.

6 Conclusion

In this work, we pointed out the necessity and the challenge of accurately estimating the label noise matrix for the latent variable label noise model in small sample size problems. To improve the noise proportion estimation, we proposed a simple yet effective approach which employs a sparsity promoting regularisation to constrain the values of the elements of the label noise matrix. Although our modelling approach inevitably resulted in a large number of regularisation parameters which required further tuning, we facilitated the matter by adopting a Bayesian approach to infer the regularisation parameters without the need to perform the costly cross validation process. We then showed how the

resulting approach can be seamlessly incorporated into existing robust classifiers and showed that the estimation of the noise matrix learned by the proposed method is more accurate than that obtained from the existing approaches. Empirical results demonstrated that the unregularised label noise model indeed struggled in the cases where the training data is inadequate relative to the number of data classes and data dimensionality, and that the proposed Bayesian regularisation technique helps improving the noise proportion estimation leading to improved classification performance in these challenging scenarios. The proposed regularisation was also found to be superior to existing regularisation approaches.

There are a number of directions for which the current research can be advanced. On one hand, in this work we restrict ourselves to the study of class-conditional label noise (also known as random noise in the literature), and whether or not the proposed technique is applicable to the instance-dependent label noise is still unclear. Further investigation is necessary in order to tackle the arguably more realistic form of label noise. On the other hand, extending the regularisation to a more advanced form of sparsity assumption such as group sparsity is also worth studying in the subsequent work. Finally, to apply the proposed technique in real-time setting requires the model to learn from incoming data and to adapt its concept accordingly. Combining label noise problems with incremental online learning is definitely an interesting research direction.

Appendix

In this section we show how to derive the multiplicative fixed point update equations for label noise parameters. For ω_{00} and ω_{01}, we first construct a Lagrangian of Eq. (14) imposing a constraint that $\omega_{00} + \omega_{01} = 1$,

$$\mathcal{R}(\mathbf{w}, \Omega) = \sum_{n=1}^{N} (1 - \tilde{y}_n) \log \tilde{P}_n^0 + \tilde{y}_n \log \tilde{P}_n^1 - \sum_{j=0}^{1} \sum_{k=0}^{1} \alpha_{jk} \omega_{jk} + \lambda (1 - \sum_{l=0}^{1} \omega_{0l}) \tag{25}$$

Taking the derivative of the above w.r.t. ω_{00} we can arrive at,

$$\lambda = \sum_{n=1}^{N} \frac{(1 - \tilde{y}_n)}{\tilde{P}_n^0} P_n^0 - \alpha_{00} \tag{26}$$

Multiplying the above by ω_{00} yields,

$$\omega_{00} \lambda = \omega_{00} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \right) \tag{27}$$

We can derive a similar expression for ω_{01} which turns out to be,

$$\omega_{01} \lambda = \omega_{01} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \right) \tag{28}$$

Summing Eqs. (27) and (28) and using the fact that $\omega_{00} + \omega_{01} = 1$ we can work out the expression for the Lagrange multiplier.

$$\lambda = \omega_{00} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \right) + \omega_{01} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \right) \tag{29}$$

Substitute λ in Eq. (29) back into Eqs. (27) and (28) gives us the multiplicative update equations for ω_{00} and ω_{01}

$$\omega_{00} = \frac{\omega_{00} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \right)}{\omega_{00} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \right) + \omega_{01} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \right)} \tag{30}$$

$$\omega_{01} = \frac{\omega_{01} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \right)}{\omega_{00} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^0} - \alpha_{00} \right) + \omega_{01} \left(\sum_{n=1}^{N} (\tilde{y}_n) \frac{P_n^0}{\tilde{P}_n^1} - \alpha_{01} \right)} \tag{31}$$

The update equations for ω_{10} and ω_{11}, which can be derived similarly, turned out to be.

$$\omega_{10} = \frac{\omega_{10} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \right)}{\omega_{10} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \right) + \omega_{11} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \right)} \tag{32}$$

$$\omega_{11} = \frac{\omega_{11} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \right)}{\omega_{10} \left(\sum_{n=1}^{N} (1 - \tilde{y}_n) \frac{P_n^1}{\tilde{P}_n^0} - \alpha_{10} \right) + \omega_{11} \left(\sum_{n=1}^{N} \tilde{y}_n \frac{P_n^1}{\tilde{P}_n^1} - \alpha_{11} \right)} \tag{33}$$

Acknowledgements This research is supported by Thailand Research Fund (TRF) and Office of the Higher Education Commission (grant number MRG6280252). Data Science Research Centre, the Department of Computer Science, Chiang Mai University provides research and computing facilities. Finally, the authors would like to express their gratitude for the constructive feedback from the editor and anonymous reviewers.

Funding This research is supported by Thailand Research Fund (TRF) and Office of the Higher Education Commission (Grant number MRG6280252).

Data Availibility Statement Code which generates the synthetic data is available at https://github.com/jakramate/brNoiseModel.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability Codes are available at https://github.com/jakramate/brNoiseModel.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci USA 96(12):6745–6750
2. Barandela R, Gasca E (2000) Decontamination of training samples for supervised pattern recognition methods. In: Proceedings of the joint IAPR international workshops on statistical techniques in pattern recognition and structural and syntactic pattern recognition, pp 621–630 (2000)
3. Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. In: Proceedings of the Asian conference on machine learning, pp 97–112 (2011)
4. Bootkrajang J, Chaijaruwanich J (2018) Towards instance-dependent label noise-tolerant classification: a probabilistic approach. Pattern Anal Appl 1–17 (2018)
5. Bootkrajang J, Kabán A (2012) Label-noise robust logistic regression and its applications. In: Proceedings of the Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 143–158
6. Bootkrajang J, Kabán A (2013) Classification of mislabelled microarrays using robust sparse logistic regression. Bioinformatics 29(7):870–877
7. Bootkrajang J, Chaijaruwanich J (2020) Towards instance-dependent label noise-tolerant classification: a probabilistic approach. Pattern Anal Appl 23:95–111. https://doi.org/10.1007/s10044-018-0750-z
8. Brodley CE, Friedl MA (1996) Identifying and eliminating mislabeled training instances. In: Proceedings of the thirteenth national conference on artificial intelligence, vol 1, pp 799–805
9. Buntine WL (1991) Bayesian backpropagation. Complex Syst 5:603–643
10. Cawley GC, Talbot NL (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 11:2079–2107
11. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
12. Frénay B, Kabán A (2014) A comprehensive introduction to label noise. In: Proceedings of the European symposium on artificial neural networks, computational intelligence and machine learning
13. Ghosh A, Kumar H, Sastry P (2017) Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence
14. Ghosh A, Manwani N, Sastry PS (2015) Making risk minimization tolerant to label noise. Neurocomputing 160:93–107
15. Goldberger J, Ben-Reuven E (2017) Training deep neural-networks using a noise adaptation layer. In: Proceedings of the 5th international conference on learning representation
16. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications
17. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition, pp 2261–2269
18. Jindal I, Nokleby M, Chen X (2016) Learning deep networks from noisy labels with dropout regularization. In: Proceedings of the 16th international conference on data mining, pp 967–972
19. Karimi D, Dou H, Warfield SK, Gholipour A (2019) Deep learning with noisy labels: exploring techniques and remedies in medical image analysis. arXiv:1912.02911
20. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto
21. Lawrence ND, Schölkopf B (2001) Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of the international conference on machine learning, pp 306–313
22. Lee KH, He X, Zhang L, Yang L (2018) Cleannet: transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5447–5456
23. Li D, Liu Y, Huang D (2020) Development of semi-supervised multiple-output soft-sensors with co-training and tri-training MPLS and MRVM. Chemom Intell Lab Syst 199:103970
24. Li M, Soltanolkotabi M, Oymak S (2020) Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: Proceedings of the international conference on artificial intelligence and statistics. PMLR, pp 4313–4324
25. Liu Y, Pan Y, Huang D (2015) Development of a novel adaptive soft-sensor using variational Bayesian pls with accounting for online identification of key variables. Ind Eng Chem Res 54(1):338–350
26. Long PM, Servedio RA (2010) Random classification noise defeats all convex potential boosters. Mach Learn 78(3):287–304
27. Lugosi G (1992) Learning with an unreliable teacher. Pattern Recognit 25:79–87
28. Manwani N, Sastry PS (2013) Noise tolerance under risk minimization. IEEE Trans Cybern 43(3):1146–1151
29. Martín-Merino M (2013) A kernel SVM algorithm to detect mislabeled microarrays in human cancer samples. In: 13th IEEE international conference on bioinformatics and bioengineering. IEEE, pp 1–4
30. Matsuda Y, Hoashi H, Yanai K (2012) Recognition of multiple-food images by detecting candidate regions. In: Proceedings of the IEEE international conference on multimedia and expo, pp 25–30
31. Menon A, Van Rooyen B, Ong CS, Williamson B (2015) Learning from corrupted binary labels via class-probability estimation. In: Proceedings of the international conference on machine learning, pp 125–134
32. Natarajan N, Dhillon IS, Ravikumar PK, Tewari A (2013) Learning with noisy labels. In: Proceedings of the advances in neural information processing systems, pp 1196–1204
33. Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L (2017) Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1944–1952
34. Raykar VC, Yu S, Zhao LH, Valadez GH, Florin C, Bogoni L, Moy L (2010) Learning from crowds. J Mach Learn Res 11:1297–1322
35. Reeve H, Kabán A (2019) Fast rates for a KNN classifier robust to unknown asymmetric label noise. In: Proceedings of the international conference on machine learning, pp 5401–5409
36. Rusiecki A (2020) Standard dropout as remedy for training deep neural networks with label noise. In: Theory and applications of dependable computer systems, pp 534–542
37. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations

38. Song H, Kim M, Park D, Lee JG (2019) Prestopping: How does early stopping help generalization against label noise? arXiv:1911.08059
39. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, Fergus R (2014) Training convolutional networks with noisy labels. arXiv:1406.2080
40. Tanno R, Saeedi A, Sankaranarayanan S, Alexander DC, Silberman N (2019) Learning from noisy labels by regularized estimation of annotator confusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11244–11253
41. Wager S, Wang S, Liang PS (2013) Dropout training as adaptive regularization. In: Proceedings of the advances in neural information processing systems, pp 351–359
42. West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H Jr, Marks JAO, Nevins JR (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. Proc Natl Acad Sci USA 98(20):11462–11467
43. Zhang R, Chen Z, Zhang S, Song F, Zhang G, Zhou Q, Lei T (2020) Remote sensing image scene classification with noisy label distillation. Remote Sens 12(15). https://doi.org/10.3390/rs12152376
44. Zhang Z, Sabuncu M (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. In: Proceedings of the advances in neural information processing systems, pp 8778–8788

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com