Noname manuscript No.
(will be inserted by the editor)

Toward Instance-dependent Label Noise Tolerant Classification: A Probabilistic Approach

Jakramate Bootkrajang · Jeerayut Chaijaruwanich

Received: date / Accepted: date

Abstract Learning from labelled data is becoming more and more challenging due to inherent imperfection of training labels. Existing label-noise tolerant learning machines were primarily designed to tackle class-conditional noise which occurs at random, independently from input instances. However, relatively less attention was given to a more general type of label noise which is influenced by input features. In this paper, we try to address the problem of learning a classifier in the presence of instance-dependent label noise by developing a novel label noise model which is expected to capture the variation of label noise rate within a class. This is accomplished by adopting a probability density function of a mixture of Gaussians to approximate the label flipping probabilities. Experimental results demonstrate the effectiveness of the proposed method over existing approaches.

Keywords instance-dependent label noise · classification · logistic regression

1 Introduction

Classical machinery of learning a classifier relies on a perfect set of labelled data. Unfortunately, it is becoming more difficult to guarantee the correctness of the training labels. In many classification tasks, labelling errors originate from the lack of knowledge or resources to annotate the input instances [1–3]. In addition, the quality of labels is being compromised in order to acquire large amount of training data quickly and cheaply [4–6].

In general, there are three forms of label noises: class-conditional label noise, instance-dependent label noise and adversarial label noise [7,8]. Class-conditional noise is the noise which occurs randomly and independently from

Department of Computer Science, Chiang Mai University, Chiang Mai, Thailand, 50200
Tel.: +66-053-943-414
fax: +66-053-892-281
E-mail: jakramate.b@cmu.ac.th (corresponding author), jeerayut.c@cmu.ac.th

the input feature. The noise is often referred to as random noise because labels of random instances within a class could be flipped subjected to same label flipping probability. Instance-dependent noise is the noise which is in parts influenced by input features. The dependency can be of many forms. One of the most natural ones is the case where instances that live closer to the decision boundary have higher chance of being mislabelled. Consequently, the flipping probabilities for labels in the same class are then not necessarily equal. Adversarial noise is an extreme noise that is intentionally injected into the dataset to perturb the classification algorithm the most [9]. A good example of this type of noise can be found in spam email classification problem where spams were specially crafted to look like normal email and to mislead the algorithm as much as possible.

Among the three types of noises, random label noise is probably the one that received most attention during the past. Accordingly, methods for dealing with random label noise are rather well developed. Notable model-based approaches for dealing with random label noise include robust linear discriminant analysis [10] robust Kernel Fisher Discriminant [11,12] robust Logistic Regression [13,14], and robust kernel logistic regression [15] to name just a few. Furthermore, theoretical properties and its effects on learning algorithm are relatively well understood [16–20].

Compared to random labelling error, the effects of instance-dependent label noise on learning machine is somewhat less understood and methods for tackling such noise are still lacking. Part of the reason might be that there can be many types of dependencies between input instance and label noise. As such, modelling the occurrence of instance-dependent label is more difficult than the random noise case since any modelling assumption might be bound to specific dependencies only. One of the earlier attempts to address instance-dependent noise is a work in [21] where a model of label noise conditioned on the distance of input from the decision boundary was studied. The work focused on theoretical aspect of the model in the presence of such noise but provided no algorithmic solution for learning the model. Recently, [22] proposed a generalisation of the latent variable label noise model [11] which was originally formulated for random noise to cover the cases where noise does not occur at random. The work assumed label flipping is more likely when input instance lives closer to decision boundary and proposed to use a probability density function of the exponential distribution to model the noise. An approach under similar assumption have also been studied in [23] with one subtle difference that an unnormalised Gaussian distribution and the Laplace distribution were instead chosen to model the distribution of label noise. The authors of [23] coined the term 'Boundary Condition Noise' (BCN) to refer to this particular type of instance-dependent noise.

One of the difficulties in modelling instance-dependent noise is the fact that distribution of label noise varies from case to case. For example, the models found in [23,22] might comfortably accommodate BCN but might not be suitable for noise which occurs further from the decision boundary. This

prompts a new development of more flexible model which is able to capture wider range of instance-dependent label noises.

In this work, we will investigate a new approach to modelling instance-dependent label noise. We will employ a probability density function of a mixture of Gaussians in order to approximate label noise distribution within a class. The new label noise model will later be incorporated into a standard logistic regression classification model. The increased flexibility of the new label noise model should capture the noise rate better and hence improved classification performance can be expected.

We consider the following points to be our contributions.

- We propose a novel instance-dependent label noise model based on a mixture of Gaussians.
- We develop a new logistic regression classifier employing the new noise model and devise an algorithm to learn the classifier jointly with estimating the label noise model.
- We extensively evaluate the proposed method and demonstrate that the approach is superior to existing approaches in various situations.

The rest of the paper is organised as follows. Section 2 briefly reviews the latent variable model and its generalisation. A novel label noise function based on Gaussian mixture model is proposed in Section 3. Section 4 demonstrates the effectiveness of the proposed method using a wide range of artificial and real-world testbeds. Section 5 concludes the study.

2 Problem setting and backgrounds

A classification task is a task of inferring discrete-valued function $h : \mathbf{x} \to y$ which maps input vector $\mathbf{x} \in R^m$ to discrete class label $y \in K$ using a finite set of input-label pairs $(\mathbf{x}_n, y_n)_{n=1}^N$ drawn i.i.d from some joint distribution $\mathcal{D}$. In this study we will focus on binary classification problems where $y \in \{0, 1\}$. In an idealised setting $\mathbf{x}$ is paired with its *true* class assignment. However in many real-world classification problems the correctness of the label cannot be guaranteed and we instead have access to $S = (\mathbf{x}_n, \tilde{y}_n)_{n=1}^N \sim \tilde{\mathcal{D}}$. Here, $\tilde{y}$ denotes an *observed* (and possibly noisy) class label. The question is can we still learn h using this noisy set of labelled data?

One of the early probabilistic approaches to address the problem of learning in the presence of label noise is through the use of a so-called *latent variable model* which initially developed for a generative classifier [11]. A work in [13] later considers the model in discriminative classifier perspective. Essentially, the latent variable model seeks to explain the posterior probability of the observed label using a weighted average of the posterior probabilities of the true class label.

$$p(\tilde{y} = j|\mathbf{x}_n, \phi) = \sum_{k=0}^{1} p(\tilde{y} = j|y = k)p(y = k|\mathbf{x}_n, \phi) \tag{1}$$

Here, ϕ denotes parameters of a classification model, e.g., the weight vector of a logistic regression model. Hereinafter, we shall use $\tilde{P}_n^j$ for the posterior probability of the observed class j, and P_n^k for the posterior probability of the true class k to simplify the notation. The term, $p(\tilde{y} = j|y = k)$ in Eq.(1) represents a probability that a label was flipped from class k into class j. Clearly, the probability is independent from the input vector. Therefore, the model in Eq.(1) is well-suited for class-conditional label noise, i.e., random label noise.

Arguably, the constraint imposed by Eq.(1) is rather too rigid. In reality, noise is expected to occur more in the region of maximum confusion, i.e., near the decision boundary or region with too few examples. Previous study has empirically shown that the argument is indeed observable in real-life [23]. They found that the majority of disagreements between experts and non-expert labellers in crowdsourced labelling task occurred near the decision boundary. The work in [22] also acknowledged the limitation of the latent variable model and pioneered the generalisation of the model by taking possible dependencies between flipping probability and input instance into consideration using:

$$\tilde{P}_n^j = \sum_{k=0}^{1} p(\tilde{y} = j|y = k, \mathbf{x}_n, \phi) P_n^k \tag{2}$$

Observe that in the generalised label noise model the flipping probability term now depends on both the input vector and the parameter of the classification model. A work in [22] considers the use of constrained exponential distribution function to model the flipping probability. Using logistic regression as a classification model the flipping probability of each instance can then be expressed by $p(\tilde{y}|y = k, \mathbf{x}_n, \mathbf{w}) = \frac{1}{\alpha_k} \exp(-\mathbf{w}^T \mathbf{x}_n / \alpha_k)$. In a similar manner, [23] used the Laplace distribution and an unnormalised zero-mean Gaussian distribution to capture the noise's dynamics. They observed that the Laplace model generally yielded superior classification performances.

Although, all of the aforementioned noise models have been shown to work well on BCN problems, it is still questionable if those noise models will still work if nature of noise deviate from the BCN assumption. For example, in the case where noise rate peaks somewhere within the class and not precisely at the decision boundary, or when distribution of noise is not strictly Laplace or Gaussian. To address the possible shortcoming, in this paper we will propose a more flexible alternative. The new noise model will be based on a probability density function of Gaussian Mixture Model. We shall then incorporate the new noise model into a logistic regression to yield a label noise robust classifier.

3 Gaussian Mixture Noise Model

The generalised label noise model allows label flipping probability $p(\tilde{y} = j|y = k, \mathbf{x}, \phi)$ to be defined in different ways. Previous studies had proposed various types of distribution functions to model the noise rate. However, the predictive

Table 1: Summary of the notations used in the manuscript.

Symbol	Description
$\mathbf{x}$	The input variables
y	The true label
$\tilde{y}$	The observed label
P_n^k	The true posterior of class k of $\mathbf{x}_n$
$\tilde{P}_n^k$	The observed posterior of class k of $\mathbf{x}_n$
$\mathcal{G}$	Gaussian mixture model
C	The number of mixing components
μ_i	The mean of the i-th component
σ_i	The standard deviation of the i-th component
ω_i	The mixing weight of the i-th component
τ_n	The normalised distance of $\mathbf{x}_n$ from decision boundary
$\mathbf{w}$	The weight vector of logistic regression
λ_i	The regularisation parameter of σ_i

performance of the classifier could be compromised, when the true distribution of label noise cannot be adequately approximated by the chosen function. In this section we shall explore a new label noise model. The approach makes use of Gaussian Mixture Model (GMM) to approximate the label flipping probability. In general, a univariate GMM is expressed by:

$$\mathcal{G}(\tau) = \sum_{i=1}^{C} \omega_i \mathcal{N}(\tau; \mu_i, \sigma_i^2) \tag{3}$$

with a constraint that $\sum_{i=1}^{C} \omega_i = 1$. Here C represents the number of mixing components. We will use $\mathcal{G}(\tau)$ to model a chance that label will flip from one class to the other. The chance is expressed as a function of a random variable τ which represents a distance of an example $\mathbf{x}$ from decision boundary induced by a classifier. We note that for more sophisticated treatment of label noise, one can model the noise level as a function of instance's position. In that case, multivariate GMM could be employed in place of the univariate GMM. In the current study, however, we chose the distance from decision boundary because the measure adequately reflects our noise assumption while also being easier to work with. In the case of logistic regression parametrised by the weight vector $\mathbf{w}$, the distance is defined as $\tau_n = \mathbf{x}_n^T \mathbf{w} / \|\mathbf{w}\|$. To further account for asymmetric label flipping, i.e., the case where one class flips to the other but not vice versa, we will model the label noise associated with instances classified by the model as positive and negative using separate mixtures. In the case of logistic regression this can be easily differentiated by observing the polarity of τ_n. We denote the distance of instances which fall on the *positive* side of the decision boundary by τ_n^1 and those that fall on the *negative* side by τ_n^0. To this

end, we define the label flipping probabilities to be:

$$p(\tilde{y}_n = 1|y_n = 0, \mathbf{x}_n, \mathbf{w}) = \mathcal{G}^{01}(\tau_n^0) \tag{4}$$

$$p(\tilde{y}_n = 0|y_n = 0, \mathbf{x}_n, \mathbf{w}) = 1 - \mathcal{G}^{01}(\tau_n^0) \tag{5}$$

$$p(\tilde{y}_n = 0|y_n = 1, \mathbf{x}_n, \mathbf{w}) = \mathcal{G}^{10}(\tau_n^1) \tag{6}$$

$$p(\tilde{y}_n = 1|y_n = 1, \mathbf{x}_n, \mathbf{w}) = 1 - \mathcal{G}^{10}(\tau_n^1) \tag{7}$$

where

$$\mathcal{G}^{01}(\tau_n^0) = \sum_{i=1}^{C} \frac{\omega_i^{01}}{\sigma_i^{01}\sqrt{2\pi}} \exp\left(- \frac{(\tau_n^0 - \mu_i^{01})^2}{2(\sigma_i^{01})^2} \right) \tag{8}$$

$$\mathcal{G}^{10}(\tau_n^1) = \sum_{i=1}^{C} \frac{\omega_i^{10}}{\sigma_i^{10}\sqrt{2\pi}} \exp\left(- \frac{(\tau_n^1 - \mu_i^{10})^2}{2(\sigma_i^{10})^2} \right) \tag{9}$$

Here, $\Theta = \{\omega_i^{10}, \omega_i^{01}, \mu_i^{10}, \mu_i^{01}, \sigma_i^{10}, \sigma_i^{01}\}_{i=1}^{C}$ is a set of noise model's parameters that need to be estimated from the data.

3.1 Regularisations on the standard deviations

By definition, the value of label flipping probability must be between zero and one. However, the density function in Eq.(3) which we adopted to model the flipping probability can output any positive real. One way to guarantee that Eqs.(4)-(7) never exceed one is to control the value of σ_i's in the two mixtures. To do that we first derive a bound on the maximum of $\mathcal{G}(\tau)$.

$$\mathcal{G}(\tau) = \sum_{i=1}^{C} \frac{\omega_i}{\sigma_i\sqrt{2\pi}} \exp\Big(- \frac{(\tau_n - \mu_i)^2}{2\sigma_i^2} \Big) \tag{10}$$

$$\leq \sum_{i=1}^{C} \frac{\omega_i}{\sigma_i\sqrt{2\pi}} \tag{11}$$

$$\leq \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{C} \frac{\omega_i}{\sigma_i} \tag{12}$$

$$\leq \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{C} \frac{1}{\sigma_i} \tag{13}$$

$$\leq \frac{1}{\sqrt{2\pi}} \frac{C}{\min_i \sigma_i} \tag{14}$$

Since the exponential function in Eq.(10) attains its maximum of one when $\tau = \mu$, the value of $\mathcal{G}(\tau)$ is bounded by Eq.(11). By observing that $0 \leq \omega_i \leq 1$, we further bound the ratio ω_i/σ_i in the summation by $1/\sigma_i$. It is then not difficult to verify that Eq.(14) follows from Eq.(13). We note that the bound is quite loose especially for large C. This might affect the working of the

model that employs a GMM with too many mixing components in the sense that its σ_i can be unreasonably large. Large σ_i translates to flat PDF and hence indicating that the model is ignoring the label noise.

So, to ensure that label noise function gives out number between zero and one we require that $\mathcal{G}(\tau) \leq \frac{1}{\sqrt{2\pi}} \frac{C}{\min_i \sigma_i} \leq 1$, or

$$\min_i \sigma_i \geq \frac{C}{\sqrt{2\pi}} \tag{15}$$

We will introduce a regularisation to restrict the value of the smallest σ_i to be greater than $C/\sqrt{2\pi}$. Since during the parameters learning process we do not know which σ_i will be the smallest we thus enforce the constraint on all σ_i. For this purpose, we will employ a log-barrier function of the form, $\log(\sigma_i - C/\sqrt{2\pi})$, to enforce such constraint on each of the σ_i's in the two mixtures. Essentially, the log-barrier discourages σ_i from getting too close to the bound.

3.2 Objective function

We will now incorporating the proposed label noise function into the standard logistic regression classifier. To this end, let us first write down the objective function of the Gaussian mixture model based robust logistic regression (GMMLR) as well as all of the regularisations in the form of a penalised log-likelihood:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \Theta) = \sum_{n=1}^{N} \Bigg\{ & \tilde{y}_n \log\left([1 - \mathcal{G}_n^{10}]P_n^1 + \mathcal{G}_n^{01} P_n^0\right) \\ & + (1 - \tilde{y}_n) \log\left(\mathcal{G}_n^{10} P_n^1 + [1 - \mathcal{G}_n^{01}]P_n^0\right) \Bigg\} \\ & + \sum_{j \in \{01,10\}} \sum_{i=1}^{C} \lambda_i^j \log\left(\sigma_i^j - \frac{C}{\sqrt{2\pi}}\right)\end{aligned} \tag{16}$$

where as before $\Theta = \{\omega_i^{10}, \omega_i^{01}, \mu_i^{10}, \mu_i^{01}, \sigma_i^{10}, \sigma_i^{01}\}_{i=1}^{C}$ denotes a set of parameters to the noise model. Here we simplify some notation by writing $\mathcal{G}_n^{jk} = \mathcal{G}^{jk}(\tau_n^j)$. The first term in Eq.(16) is the objective function for the logistic regression. The second term is a regularisation term imposing constraints on the standard deviations.

3.3 Model selection

It is apparent that our model formulation involves a number of regularisation parameters. A traditional way of picking good value of those is by performing cross validation. However, the technique requires a lot of computational efforts.

To facilitate the selection of λ_i, we propose a simple heuristic to select a good value of the regularisation parameters by:

$$\lambda_i = \frac{1}{(\sigma_i - C/\sqrt{2\pi})} \tag{17}$$

The motivation behind this heuristic is to discourage σ_i from taking extreme values. First, notice that Eq.(16) can be maximised by increasing the value of σ_i. But that is not always favourable because large σ_i translates to a flat Gaussian indicating that the model believes there is no label noise in the dataset. In such case, the robust model will behave like a traditional non-robust classifier. By setting λ_i using Eq.(17), we are balancing the gain in the likelihood value as σ_i increases and hence preventing it from being too large. The heuristic also helps enforcing the constraint that σ_i must be greater than $C/\sqrt{2\pi}$. Observe when σ_i is getting close to $C/\sqrt{2\pi}$, e.g., $(\sigma_i - C/\sqrt{2\pi} < 1)$, the log-barrier function will be negative. At the same time λ_i will be large resulting in the undesirable decrement of the objective function value. In effect, this prevents σ_i from getting too small.

3.4 Learning the model

Once the formulation is finalised, we now move on to model optimisation. There are basically two sets of parameters in concern. The first is $\mathbf{w}$, the weight vector of the logistic regression model and the second is Θ, the parameters to the label noise model. Since there is no close form solution, we resort to gradient descend methodology for estimating the parameters. In particular, we adopted a variant of Newton's method called quasi-Newton method by which the calculation of the Hessian can be facilitated. We implemented the machinery of learning using publicly available optimisation package [24]. In the following we will derive the gradients of the objective function w.r.t to each of the parameters.

First, the gradient of the objective function w.r.t $\mathbf{w}$ is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \sum_{n=1}^{N} \left(\frac{\tilde{y}_n}{\tilde{P}_n^1} - \frac{1-\tilde{y}_n}{\tilde{P}_n^0} \right) (1 - \mathcal{G}_n^{10} - \mathcal{G}_n^{01}) P_n^1 P_n^0 \mathbf{x}_n \tag{18}$$

Next, the derivatives of Eq.(16) w.r.t the means of $\mathcal{G}^{10}$ and $\mathcal{G}^{01}$ are calculated,

$$\frac{\partial \mathcal{L}}{\partial \mu_i^{10}} = \sum_{n=1}^{N} \left(\frac{1-\tilde{y}_n}{\tilde{P}_n^0} - \frac{\tilde{y}_n}{\tilde{P}_n^1} \right) \left(\frac{(\tau_n^1 - \mu_i^{10})}{(\sigma_i^{10})^2} \right) \omega_i^{10} P_n^1 \mathcal{N}_i^{10} \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_i^{01}} = \sum_{n=1}^{N} \left(\frac{\tilde{y}_n}{\tilde{P}_n^1} - \frac{1-\tilde{y}_n}{\tilde{P}_n^0} \right) \left(\frac{(\tau_n^1 - \mu_i^{01})}{(\sigma_i^{01})^2} \right) \omega_i^{01} P_n^0 \mathcal{N}_i^{01} \tag{20}$$

where $\mathcal{N}_i^{10}$ and $\mathcal{N}_i^{01}$ denote the i-th Gaussian of $\mathcal{G}^{10}$ and $\mathcal{G}^{01}$, respectively. Likewise, the gradients of the objective w.r.t σ_i are found to be:

$$\frac{\partial \mathcal{L}}{\partial \sigma_i^{10}} = \sum_{n=1}^{N} \left(\frac{1-\tilde{y}_n}{\tilde{P}_n^0} - \frac{\tilde{y}_n}{\tilde{P}_n^1} \right) \left(\frac{(\tau_n^1 - \mu_i^{10})^2}{(\sigma_i^{10})^3} - \frac{1}{\sigma_i^{10}} \right) \omega_i^{10} P_n^1 \mathcal{N}_i^{10} + \frac{\lambda_i}{\sigma_i^{10} - \frac{C}{\sqrt{2\pi}}} \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma_i^{01}} = \sum_{n=1}^{N} \left(\frac{\tilde{y}_n}{\tilde{P}_n^1} - \frac{1-\tilde{y}_n}{\tilde{P}_n^0} \right) \left(\frac{(\tau_n^0 - \mu_i^{01})^2}{(\sigma_i^{01})^3} - \frac{1}{\sigma_i^{01}} \right) \omega_i^{01} P_n^0 \mathcal{N}_i^{01} + \frac{\lambda_i}{\sigma_i^{01} - \frac{C}{\sqrt{2\pi}}} \tag{22}$$

For the mixing weights, we proceed the same way. Using $\omega_i = u_i^2$ parameterisation, we first optimise for u_i by:

$$\frac{\partial \mathcal{L}}{\partial u_i^{10}} = \sum_{n=1}^{N} \left(\frac{1-\tilde{y}_n}{\tilde{P}_n^0} - \frac{\tilde{y}_n}{\tilde{P}_n^1} \right) 2u_i^{10} P_n^1 \mathcal{N}_i^{10} - 2\alpha_i u_i^{10} \tag{23}$$

$$\frac{\partial \mathcal{L}}{\partial u_i^{01}} = \sum_{n=1}^{N} \left(\frac{\tilde{y}_n}{\tilde{P}_n^1} - \frac{1-\tilde{y}_n}{\tilde{P}_n^0} \right) 2u_i^{01} P_n^0 \mathcal{N}_i^{01} - 2\alpha_i u_i^{01} \tag{24}$$

We later recover ω_i by squaring the optimised u_i. Additionally, we adopted the projection on simplex method by [25] to ensure that the mixing weights sum to unity. Algorithm 1 summarises the steps to learn the proposed "Gaussian Mixture Model-based robust Logistic Regression" (GMMLR) model

Algorithm 1 The steps to learn the proposed GMMLR model.

Require: Set of training data $(\mathbf{x}_n, y_n)_{n=1}^N$
  Initialise $\mathbf{w} \sim \texttt{uniform}[-0.1, 0.1]$,
  Initialise $\mu_i^{10} = i/C$ and $\mu_i^{01} = -i/C$ for $i = 1 : C$,
  Initialise σ_i^{10} and σ_i^{01} with value greater than $C/\sqrt{2\pi}$
  while Iteration < MaxIteration **do**
    Calculate regularisation parameter λ_i, using Eq.(17)
    Calculate distance from decision boundary $\tau_n^k = \mathbf{w}^T\mathbf{x}/||\mathbf{w}||$
    Calculate flipping probabilities using Eqs.(4)-(7)
    Update $\mathbf{w}$ using Eq.(18)
    Update μ_i^{jk} using Eq.(19) and Eq.(20)
    Update σ_i^{jk} using Eq.(21) and Eq.(22)
    Update ω_i^{jk} using Eq.(23) and Eq.(24)
  end while
Ensure: Optimised weight vector, $\mathbf{w}$. Optimised $\mu_i^{jk}, \sigma_i^{jk}, \omega_i^{jk}$.

Table 2: Characteristics of the datasets used in this study. Top 11 datasets are from the UCI repository. The bottom 5 datasets are datasets which genuinely contains label noise. The ground truths for all of the datasets except LULC are available. For LULC, the dataset is divided into training and testing sets apriori. The training labels are noisy although the exact mislabelling rate is unknown. The testing labels are clean.

UCI dataset	# Instances	# Features	Remarks
australian	690	14	-
biodegradation	1055	41	-
boston	506	13	-
diabetes	768	8	-
fertility	100	9	-
german	1000	24	-
ionosphere	351	34	-
magic04	19020	10	-
musk	476	166	-
spambase	4601	57	-
usps46	11000	256	class 4 vs class 6
Realworld dataset	**# Instances**	**# Features**	**Remarks**
breast [26]	49	7129	9 mislabellings
colon [27]	62	2000	9 mislabellings
leukaemia [28]	72	7129	1 mislabelling
websearch [15]	1030	1318	183 mislabellings
LULC [3]	10845	28	forest vs grass

4 Experiments

4.1 Experimental protocol

The datasets used in the experiments fall into two categories: data with no report of labelling errors and real-world data that genuinely contains label noise according to the literature. The first group of data contain 11 datasets from UCI repository [29]. For the second group, we used colon cancer dataset [27], breast cancer dataset [26], leukaemia dataset [28], websearch dataset [15] and Land Use/Land Cover (LULC) dataset [3], all of which have been reported to contain label noise. Table 2 summarises the characteristics of the datasets used in this study.

Since only part of the datasets in our test-suite originally contains label noise, we had to simulate the existence of label noise in the remaining datasets. To generate instance-dependent noise we first train a logistic regression on clean training data. The resulting weight vector, together with parameters of respective noise distributions are then used to calculate label flipping probabilities of instances in the training set. We adopt PDFs of Gaussian distribution $\mathcal{N}(\mu = 1, \sigma^2 = 2)$, Gamma distribution $\Gamma(shape = 1, scale = 2)$ and mixture of Guassians $0.5 * \mathcal{N}(\mu = 0, \sigma^2 = 0.5) + 0.5 * \mathcal{N}(\mu = 2, \sigma^2 = 0.5)$ as label noise generating functions. The gamma noise distribution is used to simulate boundary condition noise which can often be observed in practice [23]. Noise generated by Gaussian and mixture of Gaussians are less common but the two

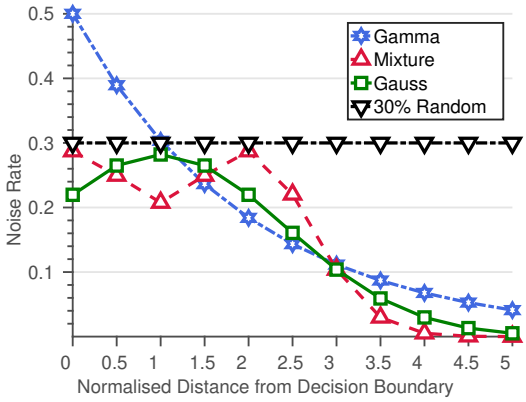

Fig. 1: Illustration of label noise rates studied in this work.

distributions will serve as testbeds for examining the flexibility of the noise model. Still, one could imagine that such complex label noise may exist in crowdsourced datasets, where labellers of different backgrounds are asked to perform the labelling task. Presumably each of the labeller who is responsible for a 'subset' of the dataset might introduce labelling errors to the dataset at different mislabelling rates. Along the same line, such complex label noise has also been found in automate product categories tagging task where labels of instances in specific data space are subjected to different label flipping probabilities [5]. In addition to instance-dependent label noises, we also investigated the model's generality in dealing with random label noise at 30% level. The noise rates corresponding the four noise generating functions are illustrated in Fig 1. These artificial label noises will be referred to as *gaussian noise*, *gamma noise*, *mixture noise* and *random noise* respectively.

In the case where label noise is already presented in the data, we did not inject artificial noise any further. Since the ground truths are available for these set of data we can always evaluate the learned model using noise-free test data. We divided the data using 80/20 percentage of training and testing set split except for the case of breast dataset where we used 90/10 splitting due to high-dimensional and low sample size nature of the data.

4.2 Effect of the number of mixing components

We will begin with a set of experiments which seeks to better understand the working of the GMMLR model. Firstly, we want to see how C, the number of mixing components, affects the model's predictive performance. We will compare GMMLR utilising 1, 3, 5, 7 and 9 mixing components named GMMLR-1, GMMLR-3, GMMLR-5, GMMLR-7 and GMMLR-9 on 11 UCI datasets corrupted by the four types of label noises described above. We performed 20 experiment repetitions using 80/20 training/testing data splitting for each

Table 3: Mean classification errors (%) and standard errors of GMMLR model with varying number of mixing components under *mixture noise* averaged over 20 runs. Boldface entry highlights the best mean rank.

Dataset	GMMLR-1	GMMLR-3	GMMLR-5	GMMLR-7	GMMLR-9
australian	14.49 ± 3.51	14.93 ± 3.38	14.42 ± 3.14	13.95 ± 2.76	14.06 ± 2.83
biodeg	17.18 ± 2.80	16.97 ± 2.66	17.35 ± 3.18	17.46 ± 3.47	17.87 ± 4.90
boston	15.39 ± 4.34	15.59 ± 4.52	15.78 ± 4.49	16.23 ± 4.50	16.37 ± 4.41
diabetes	25.71 ± 2.93	25.00 ± 2.68	25.26 ± 2.66	25.26 ± 2.70	25.26 ± 2.71
fertility	19.75 ± 7.69	19.75 ± 10.06	18.75 ± 9.44	18.75 ± 9.44	18.75 ± 9.44

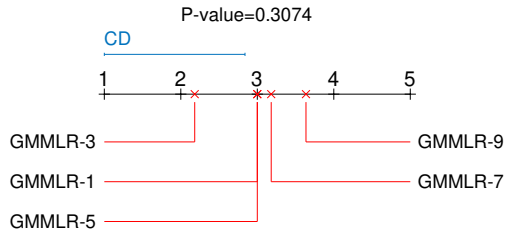

Fig. 2: Nemenyi post-hoc test for comparing five GMMLR models with varying number of mixing components under *mixture noise*. In this case, the predictive performances were not statistically different as tested by Friedman test at 0.05 level.

Table 4: Mean classification errors (%) and standard errors GMMLR model with varying number of mixing components under *gaussian noise* averaged over 20 runs. Boldface entry highlights the best mean rank.

Dataset	GMMLR-1	GMMLR-3	GMMLR-5	GMMLR-7	GMMLR-9
australian	15.62 ± 3.26	15.04 ± 4.10	15.11 ± 4.17	15.25 ± 4.18	15.25 ± 4.21
biodeg	17.27 ± 3.16	17.13 ± 3.15	17.13 ± 3.28	17.54 ± 3.17	17.84 ± 3.29
boston	14.17 ± 3.12	14.66 ± 3.43	14.80 ± 4.08	16.32 ± 4.46	16.96 ± 4.52
diabetes	25.45 ± 4.81	25.32 ± 4.85	25.88 ± 4.60	25.91 ± 4.56	25.88 ± 4.58
fertility	18.75 ± 10.11	18.50 ± 10.01	18.50 ± 9.75	18.75 ± 9.72	18.50 ± 9.75
german	28.00 ± 3.57	28.93 ± 4.59	28.70 ± 4.27	28.70 ± 4.29	28.70 ± 4.29
ionosphere	15.77 ± 3.89	16.13 ± 4.27	15.99 ± 4.07	16.13 ± 4.07	16.06 ± 4.17
magic04	21.27 ± 0.70	21.31 ± 0.73	21.63 ± 1.07	21.84 ± 1.12	21.99 ± 1.13
musk	25.78 ± 5.31	25.94 ± 5.62	25.99 ± 4.88	25.89 ± 5.05	26.15 ± 5.39
spambase	8.12 ± 0.90	8.12 ± 0.90	8.31 ± 0.82	8.52 ± 0.81	8.97 ± 1.01
usps46	1.92 ± 1.20	1.91 ± 1.21	1.80 ± 1.08	1.81 ± 1.14	1.82 ± 1.16
Meanrank	**2.3182**	2.5000	2.5455	3.7273	3.9091

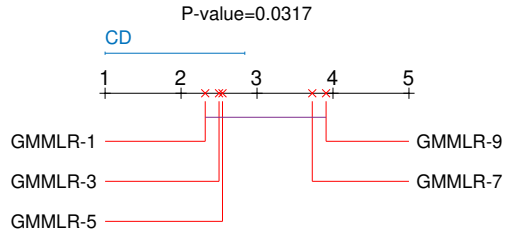

Fig. 3: Nemenyi post-hoc test for comparing GMMLR with varying number of mixing components under *gaussian noise*. Although, p-value from Friedman test indicates significant differences in performance, the post-hoc test shows that the differences are still within critical value.

noise types and computed the average classification errors and their standard errors.

The average classification errors incurred by GMMLR-1 to GMMLR-9 in the presence of *mixture noise*, *gaussian noise*, *random noise* and *gamma noise* are presented in Table 3 - 6, respectively. In each table, we ranked the performance of the five variants of GMMLR on each dataset, and summarised their mean ranks in the bottom row of the respective table. We performed Friedman test at 0.05 level as described in [30] followed by Nemenyi post-hoc test in order to compare the performances of the five classifiers. The graphical representation of the Nemenyi test are placed right after its corresponding table. We decided to draw the graphs for all of the experiments regardless of whether the p-value from the Friedman test is smaller than 0.05 or not.

From the results, there seem to be some significant differences in the performances of the five variants of GMMLR in all noise types except for the case of mixture noise, as first tested by the Friedman test. That is, under *gaussian noise* (Table 4 and Fig.3), *gamma noise* (Table 6 and Fig.5) and *random noise* (Table 5 and Fig.4) the p-values are smaller than 0.05 and so a post-hoc test was necessary. However, Nemenyi post-hoc test showed that groups of classifiers with significantly different performances can be formed only in *random noise* case. From Fig.4, we see that GMMLR-3 and GMMLR-1 is significantly better than GMMLR-9 while there is not enough evidence to tell if GMMLR-3 is better than GMMLR-1, GMMLR-5 or GMMLR-7.

Despite the statistical tests showing that all variants of GMMLR are not statistically different in most of the cases tested, we can still observe some general trends from the graphical representations of Nemenyi test. From Fig.2-4, we notice that GMMLR-1, GMMLR-3, and GMMLR-5, albeit not statistically different, were ranked higher than GMMLR-7 and GMMLR-9. The slightly lower predictive performances exhibited by GMMLR with large C could be due to the bound on the standard deviation in Eq.(14), where the s.d. is excessively constrained for large C. It can be understood that when the s.d. is large, as enforced by the bound, GMMLR becomes less aware of label noise. As such, the performance of GMMLR with larger C might be compromised.

Table 5: Mean classification errors (%) and standard errors of GMMLR models with varying number of mixing components under *random noise* averaged over 20 runs. Boldface entry highlights the best mean rank.

Dataset	GMMLR-1	GMMLR-3	GMMLR-5	GMMLR-7	GMMLR-9
australian	16.45 ± 3.70	16.41 ± 3.89	17.36 ± 4.10	17.68 ± 3.93	17.68 ± 3.93
biodeg	20.47 ± 4.19	20.52 ± 3.43	20.95 ± 3.54	21.14 ± 4.03	21.42 ± 4.10
boston	19.36 ± 6.89	19.17 ± 7.19	19.85 ± 6.92	20.34 ± 6.51	20.78 ± 6.02
diabetes	25.97 ± 4.40	25.94 ± 4.20	25.94 ± 4.18	26.14 ± 3.87	26.10 ± 3.91
fertility	24.25 ± 11.50	22.75 ± 10.45	22.75 ± 10.45	22.75 ± 10.45	22.75 ± 10.45

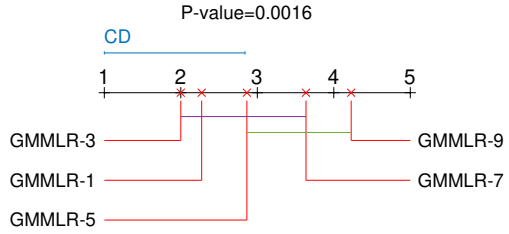

Fig. 4: Nemenyi post-hoc test for comparing GMMLR with varying number of mixing components under *random noise*. The post-hoc test groups the classifiers into two groups where we see that GMMLR-3 is better than GMMLR-9. However, there is not enough evidence to say if any of GMMLR-1,3,5,7 is better than the others.

In terms of the running time, it is apparent as shown in Table 7 that a model with smaller number of mixing components runs faster. We see that the time required to train the model increases, rather linearly, as the number of mixing components increases. So, in light of the predictive performance and the running time, it seems that GMMLR with $C = 3$ is preferable to other variants of GMMLR.

4.3 Effects of standard deviation regularisation

Next, we will investigate the influence and the importance of regularising the value of the standard deviations of the Gaussians in the mixture. The standard deviation indicates the spread of each Gaussian and hence directly controls the value of the flipping probability. By formulation, the flipping probability must be between zero and one. Figure 6 illustrates one of the estimated label flipping probabilities obtained from GMMLR with s.d. regulariser (top) and without s.d. regulariser (bottom). We see that when s.d. is too small the output from the PDF of the GMM is no longer below one. Although, this is perfectly fine in

Table 6: Mean classification errors (%) and standard errors of GMMLR with varying number of mixing components under *gamma noise* averaged over 20 runs. Boldface entry highlights the classifier with the best mean rank.

Dataset	GMMLR-1	GMMLR-3	GMMLR-5	GMMLR-7	GMMLR-9
australian	16.92 ± 4.97	18.22 ± 4.77	19.38 ± 5.50	19.93 ± 5.65	19.93 ± 5.65
biodeg	16.37 ± 4.62	16.23 ± 4.65	17.16 ± 4.17	17.84 ± 3.73	18.27 ± 3.81
boston	17.25 ± 5.18	17.25 ± 5.33	18.73 ± 5.21	19.17 ± 5.09	19.36 ± 4.95
diabetes	23.31 ± 4.10	24.03 ± 4.62	25.36 ± 4.01	25.45 ± 4.06	25.45 ± 4.06
fertility	31.75 ± 13.89	29.75 ± 14.82	29.75 ± 14.82	30.00 ± 14.51	29.75 ± 14.82

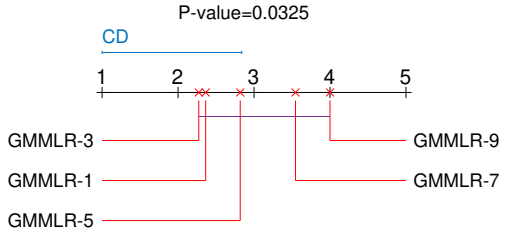

Fig. 5: Nemenyi post-hoc test for comparing GMMLR with varying number of mixing components under *gamma noise*. Although, p-value from Friedman test indicates differences in performance, the post-hoc test shows that the differences are still within critical value.

Table 7: Running times (seconds) of variants of GMMLR learning from Gamma noise on Intel Core-i5 3.2GHz machine averaged over 20 runs.

Dataset	GMMLR-1	GMMLR-3	GMMLR-5	GMMLR-7	GMMLR-9
australian	0.69 ± 0.12	1.54 ± 0.27	2.75 ± 0.20	4.22 ± 0.35	5.83 ± 0.38
biodeg	0.90 ± 0.07	2.06 ± 0.18	3.62 ± 0.26	5.30 ± 0.49	7.29 ± 0.61
boston	0.63 ± 0.08	1.50 ± 0.24	2.71 ± 0.25	3.99 ± 0.27	5.44 ± 0.32
diabetes	0.61 ± 0.11	1.39 ± 0.20	2.68 ± 0.27	4.02 ± 0.25	5.57 ± 0.21
fertility	0.31 ± 0.07	0.79 ± 0.08	1.48 ± 0.03	2.13 ± 0.03	2.86 ± 0.04
german	0.73 ± 0.13	1.64 ± 0.24	3.07 ± 0.26	4.55 ± 0.27	6.31 ± 0.28
ionosphere	0.63 ± 0.04	1.44 ± 0.16	2.55 ± 0.18	3.68 ± 0.29	4.93 ± 0.32
magic04	5.10 ± 0.20	12.35 ± 0.77	22.57 ± 1.65	34.67 ± 2.90	48.18 ± 3.61
musk	0.74 ± 0.08	1.70 ± 0.17	2.94 ± 0.24	4.27 ± 0.32	5.69 ± 0.40
spambase	2.16 ± 0.10	4.54 ± 0.22	7.84 ± 0.58	11.70 ± 0.82	16.11 ± 1.00
usps46	1.91 ± 0.12	4.26 ± 0.20	6.85 ± 0.32	9.64 ± 0.46	12.96 ± 0.52

standard statistical usage where the probability is taken to be the integral of the PDF, it might not always work for our case where the output from the PDF is directly used as label flipping probability. When the flipping probability is not between zero and one, the term on the left hand side of Eq.(2) is no longer a valid posterior probability in which case the classification performance might suffer.

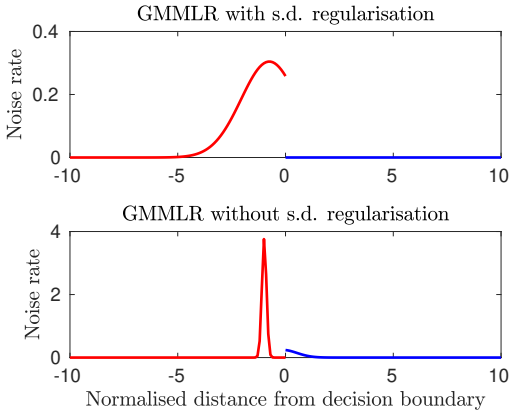

Fig. 6: Illustration of estimated noise rate. Observe that, without regularisation, too small s.d. would produce the flipping probability that violates the probabilistic constraint.

To validate our speculation, we investigated the effect of standard deviation regularisers λ_i, by comparing GMMLR *with* regulariser (GMMLR-reg) and GMMLR *without* regulariser (GMMLR-noreg). We opted for GMMLR with 3 mixtures as it achieved relatively good performance in the previous experiment. In this experiment, GMMLR-reg selects λ_i using the proposed heuristic in Eq.(17). We tested the two variants on the 11 UCI datasets corrupted with the four types of noises mentioned above. Again, we repeated the experiment for 20 repetitions for each label noise configurations and computed the average classification errors and their standard errors. The results are summarised in Table 8.

From the results, we observe that the regularisation on the s.d. is quite important and turning it off results in slightly higher classification errors. To test if such differences are significant, we then employed Wilcoxon ranksum test at 0.05 level to compare the performances of the two GMMLRs. Surprisingly, we only found that the differences were significant, e.g., p-value < 0.05 in two cases namely in the spambase dataset corrupted by *mixture noise* and biodeg dataset corrupted by *gaussian noise*. The reason why the majority of the performance differences of the two GMMLRs are not statistically significant could be that the estimated s.d. of GMMLR-noreg were still sometimes legitimate (the resulting PDF outputs values between zero and one). However, there were also times when the estimated s.d. of GMMLR-noreg was unusable which can be observed through the relatively larger error variances compared to those of GMMLR-reg in most of the datasets.

Despite the fact that we do not have enough evidence to say that GMMLR-reg is statistically better than GMMLR-noreg, we still see that GMMLR-reg achieved lower error rates in general and in real world classification problems there is a good chance that GMMLR with s.d. regulariser will perform better.

Table 8: Mean classification errors (%) and standard errors of GMMLR without standard deviation regulariser (GMMLR-noreg) and GMMLR which employs the regulariser (GMMLR-reg). Boldface entries are the best ones (not necessarily statistically significant). Entry marked with '-' is worse than GMMLR-reg, with '+' is better than GMMLR-reg and with '≈' is equivalent to GMMLR-reg as tested by Wilcoxon rank sum test at 0.05 level. The last row counts the cases where GMMLR-reg is better than, equivalent to, and worse than GMMLR-noreg.

Dataset	Mixture noise		Gaussian noise	
	GMMLR-noreg	GMMLR-reg	GMMLR-noreg	GMMLR-reg
australian	**14.75 ± 2.73** ≈	14.93 ± 3.38	15.58 ± 4.48 ≈	**15.04 ± 4.10**
biodeg	19.53 ± 6.58 ≈	**16.97 ± 2.66**	20.14 ± 4.79 -	**17.13 ± 3.15**
boston	17.11 ± 4.83 ≈	**15.59 ± 4.52**	17.55 ± 5.61 ≈	**14.66 ± 3.43**
diabetes	25.39 ± 2.82 ≈	**25.00 ± 2.68**	26.27 ± 4.30 ≈	**25.32 ± 4.85**
fertility	**19.50 ± 10.37** ≈	19.75 ± 10.06	21.00 ± 9.12 ≈	**18.50 ± 10.01**
german	27.73 ± 3.39 ≈	**26.82 ± 2.72**	**28.80 ± 4.51** ≈	28.93 ± 4.59
ionosphere	17.25 ± 6.93 ≈	**15.42 ± 4.56**	16.76 ± 4.81 ≈	**16.13 ± 4.27**
magic04	22.87 ± 2.64 ≈	**21.83 ± 1.04**	22.32 ± 2.16 ≈	**21.31 ± 0.73**
musk	28.54 ± 7.16 ≈	**26.56 ± 5.09**	28.39 ± 9.85 ≈	**25.94 ± 5.62**
spambase	11.78 ± 4.47 -	**8.50 ± 1.67**	9.19 ± 2.24 ≈	**8.12 ± 0.90**
usps46	4.12 ± 7.08 ≈	**2.19 ± 1.28**	4.17 ± 9.81 ≈	**1.91 ± 1.21**
Win-Tie-Loss	1-10-0	n/a	1-10-0	n/a

Dataset	Random noise		Gamma noise	
	GMMLR-noreg	GMMLR-reg	GMMLR-noreg	GMMLR-reg
australian	17.25 ± 3.77 ≈	**16.41 ± 3.89**	20.00 ± 9.74 ≈	**18.22 ± 4.77**
biodeg	23.08 ± 5.12 ≈	**20.52 ± 3.43**	19.10 ± 4.88 ≈	**16.23 ± 4.65**
boston	**17.94 ± 4.74** ≈	19.17 ± 7.19	18.92 ± 4.71 ≈	**17.25 ± 5.33**
diabetes	26.14 ± 4.18 ≈	**25.94 ± 4.20**	27.11 ± 8.31 ≈	**24.03 ± 4.62**
fertility	**21.25 ± 10.99** ≈	22.75 ± 10.45	31.00 ± 14.56 ≈	**29.75 ± 14.82**
german	29.85 ± 6.18 ≈	**28.27 ± 4.47**	31.18 ± 6.15 ≈	**29.48 ± 4.62**
ionosphere	22.39 ± 5.97 ≈	**20.85 ± 5.04**	**17.25 ± 4.88** ≈	18.24 ± 4.63
magic04	22.62 ± 1.60 ≈	**21.87 ± 1.20**	21.03 ± 1.32 ≈	**20.97 ± 0.71**
musk	29.48 ± 9.07 ≈	**27.60 ± 4.82**	29.48 ± 9.64 ≈	**23.49 ± 7.73**
spambase	11.28 ± 1.65 ≈	**10.36 ± 2.05**	9.32 ± 1.60 ≈	**9.07 ± 1.53**
usps46	11.67 ± 10.83 ≈	**10.65 ± 4.62**	3.98 ± 4.86 ≈	**3.88 ± 0.83**
Win-Tie-Loss	0-11-0	n/a	0-11-0	n/a

4.4 Cross validation versus automatic selection of λ_i

In this work we have proposed a simple heuristic for determining λ_i, the regularisation parameters of the standard deviations, instead of performing the traditional cross validation (CV). We want to study if this heuristic finds a good set of regularisation parameters compared to those found by performing a grid search using the traditional CV. To be of any usefulness, the heuristic should be able to select a "good" GMMLR model as measured by the model's predictive performance, and also within shorter period of learning time compared to CV. To test the ability of the proposed heuristic in selecting good GMMLR model, we will again take GMMLR-3, which seems to work best from the previous experiments and compare it with a GMMLR-3 for which λ_i was selected by the traditional CV. We will refer to the latter as GMMLR-3-CV. In particular, GMMLR-3-CV selected its regularisation parameters (λ_i) from the set $\{2^i\}, i \in \{\pm 1, \pm 3, \pm 5\}$ via 3 folds cross validation.

Table 9: Mean classification errors (%) and standard errors of GMMLR where its regularisers were selected using cross validation (GMMLR-3-CV) and GMMLR-3 which selects regularisers via the proposed heuristic. Boldface entries are the best ones (not necessarily statistically significant). The performance obtained from using the heuristic is similar to what we would get from performing time-consuming cross validation. The differences are not statistically significant as tested by Wilcoxon rank sum test at 0.05 level.

Dataset	Mixture noise		Gaussian noise	
	GMMLR-3-CV	GMMLR-3	GMMLR-3-CV	GMMLR-3
australian	**14.53 ± 3.29** ≈	14.93 ± 3.38	15.36 ± 4.08 ≈	**15.04 ± 4.10**
biodeg	17.06 ± 2.86 ≈	**16.97 ± 2.66**	17.16 ± 3.18 ≈	**17.13 ± 3.15**
boston	15.93 ± 4.46 ≈	**15.59 ± 4.52**	**14.26 ± 4.19** ≈	14.66 ± 3.43
diabetes	25.23 ± 2.62 ≈	**25.00 ± 2.68**	25.91 ± 4.43 ≈	**25.32 ± 4.85**
fertility	**19.00 ± 8.83** ≈	19.75 ± 10.06	19.50 ± 10.50 ≈	**18.50 ± 10.01**
german	27.32 ± 2.40 ≈	**26.82 ± 2.72**	**28.57 ± 4.14** ≈	28.93 ± 4.59
ionosphere	**15.35 ± 4.50** ≈	15.42 ± 4.56	**16.13 ± 3.97** ≈	**16.13 ± 4.27**
magic04	22.18 ± 1.52 ≈	**21.83 ± 1.04**	21.70 ± 1.49 ≈	**21.31 ± 0.73**
musk	**26.35 ± 5.16** ≈	26.56 ± 5.09	**25.62 ± 4.91** ≈	25.94 ± 5.62
spambase	8.68 ± 1.65 ≈	**8.50 ± 1.67**	8.13 ± 0.94 ≈	**8.12 ± 0.90**
usps46	2.28 ± 1.40 ≈	**2.19 ± 1.28**	**1.67 ± 1.00** ≈	1.91 ± 1.21

Dataset	Random noise		Gamma noise	
	GMMLR-3-CV	GMMLR-3	GMMLR-3-CV	GMMLR-3
australian	17.07 ± 3.70 ≈	**16.41 ± 3.89**	**16.63 ± 5.06** ≈	18.22 ± 4.77
biodeg	20.81 ± 3.47 ≈	**20.52 ± 3.43**	16.97 ± 4.04 ≈	**16.23 ± 4.65**
boston	19.31 ± 7.46 ≈	**19.17 ± 7.19**	17.55 ± 5.43 ≈	**17.25 ± 5.33**
diabetes	26.46 ± 3.94 ≈	**25.94 ± 4.20**	25.19 ± 4.07 ≈	**24.03 ± 4.62**
fertility	**22.75 ± 12.40** ≈	**22.75 ± 10.45**	**29.00 ± 14.01** ≈	29.75 ± 14.82
german	28.48 ± 4.95 ≈	**28.27 ± 4.47**	31.15 ± 5.65 ≈	**29.48 ± 4.62**
ionosphere	21.20 ± 4.37 ≈	**20.85 ± 5.04**	**17.96 ± 4.26** ≈	18.24 ± 4.63
magic04	22.43 ± 1.96 ≈	**21.87 ± 1.20**	21.59 ± 2.32 ≈	**20.97 ± 0.71**
musk	**27.19 ± 5.23** ≈	27.60 ± 4.82	**23.33 ± 7.38** ≈	23.49 ± 7.73
spambase	**10.34 ± 2.02** ≈	10.36 ± 2.05	9.15 ± 1.62 ≈	**9.07 ± 1.53**
usps46	10.90 ± 4.24 ≈	**10.65 ± 4.62**	**3.47 ± 1.09** ≈	3.88 ± 0.83

The results from the experiment are summarised in Table 9. Statistically, the differences in the error rates incurred by the two variants of GMMLR are negligible. This is because the p-values from Wilcoxon rank sum test are all greater than 0.05 level. The results can be used as an evidence to validate the usefulness of the proposed heuristic for automatic selection of λ_i. Combining this observation with the fact that CV requires much longer time to perform the model selection, it is natural to conclude that the proposed heuristic is useful for selecting λ_i as it can speed up the learning process without sacrificing the classification performance.

4.5 Comparisons with other classifiers on artificial label noises

We have seen from the previous experiments about the effect of the number of mixing components, the benefits of regularising the standard deviation and the usefulness of the proposed automatic regularisation parameters selection and come to the conclusion that GMMLR with $C = 3$, with s.d. regularisation and with the automated model selection is likely to perform best compared to other variations of GMMLR model. In this section, we will investigate the

comparative performance of the GMMLR with other existing label noise robust classifiers, with traditional classifier which is unaware of label noise as well as with the current state-of-the-art algorithm for classification problems.

Specifically, we compared the proposed method to the generalised Logistic Regression (gLR) [22], Logistic regression with Laplace noise model (LaplaceBCN) [23] which were developed for boundary condition noise and robust Logistic Regression (rLR) [13] which targets random label noise. In the case of rLR, instead of using the EM algorithm as in [13], we employed a gradient-based algorithm [14] which was shown to converge faster for optimising the model. We additionally included a traditional logistic regression (LR) and an SVM with linear kernel (SVM-Linear) [1] as baseline models. The C parameter of the SVM were chosen using 5-fold cross validation from values in $\{2^i\}$ where $i \in \{-10, -8, -6, \ldots, 6, 8, 10\}$. We decided not to include a more common SVM with radial basis function (rbf) kernel in this study since the nonlinearity of the model can yield extreme positive/negative effects on the performance e.g., serious overfitting on noisy linearly separable data and incomparably better accuracy on nonlinear datasets. As such, it is difficult to understand the intrinsic label noise robustness of the model. Still, we believe that the SVMs employing nonlinear kernels could be well compared with the kernelised version of the above robust classifiers in the future study.

The comparison with the traditional classifiers will validate the practical usefulness of label noise modelling. The comparison with gLR and LaplaceBCN will demonstrate whether the added flexibility of the GMMLR is beneficial while the comparison with rLR should highlight the inadequacies of existing latent variable model in the presence of instance-dependent label noises.

We will now present the predictive performances as measured by classification errors on 11 UCI datasets subjected to the four artificial label noises mentioned above. We note that, unlike the previous set of experiments, here we performed 100 experiment repetitions for each label noise configurations in order to obtain more reliable statistics. Table 10 summarises the results for *gamma noise* (top) and *gaussian noise* (bottom) while the results for *mixture noise* and *random noise* are presented in Table 11.

Let us first compare the proposed method to the traditional LR. From the results we see that the proposed GMMLR model exhibited more robustness towards label noise than the traditional classifier, which is unaware of mislabelling, in the majority of the datasets and under all noise types. For each of the noise types, GMMLR was statistically significantly better than LR on at least 4 datasets. Further GMMLR was never worse than LR in all of the cases. These observations well demonstrate the benefit of having the noise model built into the classifier while also highlighting the inadequacy of the traditional LR classifier in the presence of label noise.

Compared to rLR which assumes uniform noise rate, GMMLR tends to perform better in the cases where noise distributions are not uniform, i.e., *mixture noise*, *gaussian noise* and *gamma noise*. This signifies the necessity

[1] We used LIBLINEAR [31] in this study.

of having more flexible noise model. We observe further that in random noise cases, rLR sometime outperformed GMMLR, which is somehow expected. Still, GMMLR did quite well under random noise and this demonstrates the generality of the proposed model. Further, within the group of classifiers adopting instance-dependent noise model, LaplaceBCN and gLR showed impressive performances compared to GMMLR in the case where the noise matches their assumptions, e.g., *gamma noise*. However, when noise distribution deviates from their assumptions, for example *gaussian noise*, the number of times that the two classifiers were better than GMMLR was reduced.

We observed that SVM-Linear is quite robust to *gamma noise* having more winning counts than GMMLR. We think that this might due to the inherent robustness of SVM towards noise near decision boundary. As by formulation, SVM can neglect some hard-to-classify points in the confusion region through the use of slack variables and can instead selects points on the boundary of the confusion region to be the support vectors. So, in the presence of label noises near decision boundary, SVM seems to be able to select meaningful support vectors and thus the orientation of its decision boundary is not much altered. However, SVM-Linear seems to be troubled most by random noise. This is also understandable since if all of the examples are subjected to label flipping, it is then harder to ensure the correct orientation of the decision boundary since the selected support vectors might not be the optimal ones.

The last row of each tables provides a summary of the number of times that GMMLR won, tied or lose to its peers as tested by Wilcoxon ranksum test at 0.05 level. We see that GMMLR is the top performer in the non-random noise scenarios based on the winning counts. In the presence of 30% random label noise, we observe that the performances of GMMLR are comparable to rLR, and are generally better than those of LaplaceBCN and SVM. We also notice that under random noise, gLR seems to do better than LaplaceBCN. We speculate that the modified exponential distribution used by gLR has heavier tails than the noise distribution used by LaplaceBCN. It was then able to approximate uniform label noise better. Still, by formulation, the GMMLR is more flexible and so better classification performances were observed in the random noise case.

Table 10: Mean errors (%) and standard errors of GMMLR compared to existing traditional and robust classifiers. Boldface entries (also marked by '-' sign) are statistically worse than GMMLR, Underline entries ('+' sign) are statistically better than GMMLR while normal entries ('≈' sign) are statistically equivalent to GMMLR as tested by Wilcoxon rank sum test at 0.05 level. The last row counts the cases where GMMLR is better than, equivalent to, and worse than the method in that column.

Dataset	Classification error (%) under *gamma noise*					
	LR	gLR	rLR	LaplaceBCN	SVM-Linear	GMMLR
australian	**19.04 ± 4.77** -	**19.14 ± 4.80** -	18.21 ± 5.61 ≈	18.94 ± 6.49 ≈	15.23 ± 3.33 +	17.51 ± 4.44
biodeg	**18.69 ± 4.12** -	18.53 ± 4.39 ≈	19.56 ± 8.21 ≈	16.01 ± 3.53 +	17.73 ± 3.83 ≈	17.19 ± 3.72
boston	**20.09 ± 4.60** -	**19.67 ± 4.75** -	20.33 ± 8.40 ≈	19.76 ± 8.68 ≈	**20.66 ± 5.11** -	17.85 ± 5.35
diabetes	**27.32 ± 4.69** -	**27.32 ± 4.68** -	**27.42 ± 5.12** -	25.53 ± 4.48 ≈	**28.57 ± 5.46** -	25.66 ± 4.73
fertility	29.65 ± 13.66 ≈	29.95 ± 13.40 ≈	**40.40 ± 20.01** -	**40.15 ± 17.12** -	20.30 ± 15.07 +	30.05 ± 13.49
german	29.64 ± 4.48 ≈	29.66 ± 4.48 ≈	**32.23 ± 8.29** -	27.39 ± 4.06 +	**30.75 ± 3.89** -	29.02 ± 4.60
ionosphere	17.23 ± 4.81 ≈	17.99 ± 5.35 ≈	**18.89 ± 5.55** -	17.23 ± 10.60 ≈	16.55 ± 5.02 ≈	17.21 ± 4.69
magic04	**23.72 ± 3.32** -	21.41 ± 0.97 ≈	21.22 ± 0.76 ≈	20.86 ± 0.70 +	**24.53 ± 6.22** -	21.22 ± 0.88
musk	**30.87 ± 6.24** -	**26.01 ± 6.48** -	25.04 ± 5.97 ≈	**32.36 ± 9.56** -	23.97 ± 5.29 ≈	23.60 ± 5.12
spambase	**10.60 ± 1.40** -	8.19 ± 1.58 +	8.36 ± 1.70 ≈	**9.51 ± 1.66** -	**11.84 ± 4.54** -	8.60 ± 1.33
usps46	**8.12 ± 1.58** -	**6.09 ± 2.39** -	**3.27 ± 1.67** -	**3.96 ± 1.40** -	0.32 ± 0.33 +	2.67 ± 1.24
Win-Tie-Loss	8-3-0	5-5-1	5-6-0	4-4-3	5-3-3	n/a
	Classification error (%) under *gaussian noise*					
australian	15.04 ± 3.00 ≈	14.99 ± 3.09 ≈	15.36 ± 3.15 ≈	14.65 ± 3.09 ≈	14.91 ± 3.07 ≈	14.94 ± 2.88
biodeg	**17.99 ± 3.79** -	17.51 ± 3.75 ≈	17.18 ± 3.89 ≈	16.13 ± 3.30 ≈	17.24 ± 3.68 ≈	16.38 ± 3.04
boston	**17.12 ± 4.39** -	**16.40 ± 3.99** -	15.41 ± 3.65 ≈	14.42 ± 3.43 ≈	**17.07 ± 4.16** -	15.02 ± 3.26
diabetes	25.42 ± 4.10 ≈	25.39 ± 4.11 ≈	25.24 ± 3.62 ≈	24.69 ± 3.70 ≈	24.78 ± 3.91 ≈	24.96 ± 3.84
fertility	18.65 ± 7.97 ≈	18.75 ± 8.11 ≈	**30.45 ± 15.73** -	**27.05 ± 12.91** -	13.60 ± 7.22 +	19.30 ± 8.20
german	26.70 ± 3.39 ≈	26.68 ± 3.40 ≈	**28.14 ± 5.41** -	25.79 ± 3.34 ≈	27.06 ± 3.49 ≈	26.25 ± 3.31
ionosphere	16.70 ± 5.72 ≈	16.28 ± 5.70 ≈	16.69 ± 5.51 ≈	23.94 ± 23.86 ≈	14.75 ± 4.52 ≈	15.96 ± 5.23
magic04	**22.40 ± 1.21** -	**22.74 ± 1.23** -	**21.65 ± 0.73** -	21.48 ± 0.57 ≈	**23.15 ± 2.60** -	21.32 ± 0.55
musk	**30.25 ± 6.75** -	**25.45 ± 6.62** -	**25.02 ± 6.12** -	**31.65 ± 10.78** -	22.94 ± 4.57 ≈	22.27 ± 5.09
spambase	**10.72 ± 1.62** -	8.41 ± 1.58 ≈	8.23 ± 1.35 ≈	8.21 ± 1.00 +	**11.24 ± 2.63** -	8.47 ± 1.00
usps46	**5.55 ± 2.96** -	**2.94 ± 1.83** -	1.49 ± 1.00 ≈	**2.49 ± 1.25** -	0.21 ± 0.21 +	1.25 ± 0.73
Win-Tie-Loss	6-5-0	4-7-0	4-7-0	3-7-1	3-6-2	n/a

Table 11: Mean errors (%) and standard errors of GMMLR compared to existing traditional and robust classifiers. Boldface entries (also marked by '-' sign) are statistically worse than GMMLR, Underline entries ('+' sign) are statistically better than GMMLR while normal entries ('≈' sign) are statistically equivalent to GMMLR as tested by Wilcoxon rank sum test at 0.05 level. The last row counts the cases where GMMLR is better than, equivalent to, and worse than the method in that column.

Dataset	Classification error (%) under *mixture noise*					
	LR	gLR	rLR	LaplaceBCN	SVM-Linear	GMMLR
australian	15.12 ± 3.30 ≈	15.22 ± 3.37 ≈	15.62 ± 3.32 ≈	15.49 ± 6.27 ≈	14.77 ± 3.06 ≈	15.20 ± 3.20
biodeg	**19.28 ± 4.09** -	18.55 ± 4.12 ≈	18.14 ± 4.51 ≈	16.58 ± 2.78 +	18.57 ± 3.81 ≈	17.58 ± 3.18
boston	**16.81 ± 4.15** -	15.83 ± 3.96 ≈	15.86 ± 3.88 ≈	15.44 ± 7.52 ≈	**16.90 ± 3.87** -	15.45 ± 3.46
diabetes	24.32 ± 3.48 ≈	24.33 ± 3.52 ≈	24.43 ± 4.20 ≈	24.05 ± 3.43 ≈	24.27 ± 3.56 ≈	24.05 ± 3.51
fertility	19.25 ± 9.93 ≈	19.30 ± 9.77 ≈	**26.10 ± 12.18** -	**24.65 ± 12.76** -	15.15 ± 9.36 +	19.50 ± 9.76
german	27.11 ± 3.15 ≈	27.11 ± 3.13 ≈	27.43 ± 4.21 ≈	25.82 ± 2.56 ≈	**28.09 ± 3.39** -	26.32 ± 2.82
ionosphere	17.73 ± 5.82 ≈	17.32 ± 5.33 ≈	17.59 ± 5.25 ≈	31.04 ± 27.96 ≈	15.21 ± 4.90 +	16.85 ± 5.48
magic04	**22.23 ± 1.29** -	**23.76 ± 2.30** -	**22.25 ± 1.03** -	**21.95 ± 0.91** -	**23.48 ± 4.45** -	21.51 ± 0.76
musk	**32.08 ± 6.45** -	**28.68 ± 6.24** -	**28.21 ± 6.64** -	**32.57 ± 8.99** -	26.54 ± 5.79 ≈	25.51 ± 5.60
spambase	**11.79 ± 2.31** -	12.65 ± 5.91 ≈	11.57 ± 4.60 ≈	8.89 ± 1.36 +	**12.79 ± 3.10** -	9.37 ± 1.15
usps46	**3.92 ± 2.78** -	**3.09 ± 2.21** -	1.70 ± 1.31 ≈	**2.53 ± 1.48** -	0.23 ± 0.25 +	1.33 ± 1.02
Win-Tie-Loss	6-5-0	3-8-0	3-8-0	4-5-2	4-4-3	n/a
	Classification error (%) under *random noise*					
australian	16.53 ± 3.40 ≈	16.66 ± 3.51 ≈	17.32 ± 5.59 ≈	16.32 ± 4.06 ≈	15.17 ± 3.21 ≈	15.69 ± 3.26
biodeg	21.70 ± 5.04 ≈	21.24 ± 6.49 ≈	21.11 ± 7.56 ≈	18.59 ± 4.19 +	20.79 ± 4.57 ≈	20.39 ± 4.50
boston	**20.73 ± 4.74** -	19.33 ± 4.75 ≈	19.96 ± 7.10 ≈	17.80 ± 5.22 ≈	**20.30 ± 4.79** -	18.24 ± 4.84
diabetes	27.86 ± 4.48 ≈	27.81 ± 4.47 ≈	26.14 ± 4.49 ≈	26.31 ± 3.78 ≈	**29.29 ± 5.09** -	26.89 ± 4.23
fertility	25.60 ± 12.40 ≈	25.60 ± 12.48 ≈	**35.95 ± 19.47** -	**36.50 ± 16.94** -	19.50 ± 15.15 +	25.80 ± 12.14
german	29.40 ± 4.13 ≈	29.38 ± 4.14 ≈	29.35 ± 5.50 ≈	27.32 ± 3.18 ≈	**30.31 ± 4.03** -	28.31 ± 3.72
ionosphere	21.00 ± 5.71 ≈	21.61 ± 7.43 ≈	22.46 ± 8.63 ≈	25.58 ± 19.87 ≈	18.97 ± 5.38 ≈	19.85 ± 5.97
magic04	**23.83 ± 2.30** -	21.01 ± 0.87 +	24.06 ± 4.92 ≈	**22.38 ± 6.39** -	**24.45 ± 3.41** -	22.13 ± 1.24
musk	**33.90 ± 7.66** -	**30.41 ± 7.06** -	**29.60 ± 7.53** -	**34.96 ± 9.29** -	28.86 ± 6.49 ≈	27.16 ± 8.02
spambase	**13.71 ± 1.93** -	11.22 ± 5.21 ≈	10.88 ± 4.42 +	11.14 ± 2.30 ≈	**14.17 ± 4.84** -	11.15 ± 1.42
usps46	**16.08 ± 3.06** -	**13.20 ± 6.91** -	7.61 ± 5.38 ≈	**14.67 ± 9.51** -	0.73 ± 0.49 +	7.57 ± 3.95
Win-Tie-Loss	5-6-0	2-8-1	2-8-1	4-6-1	5-4-2	n/a

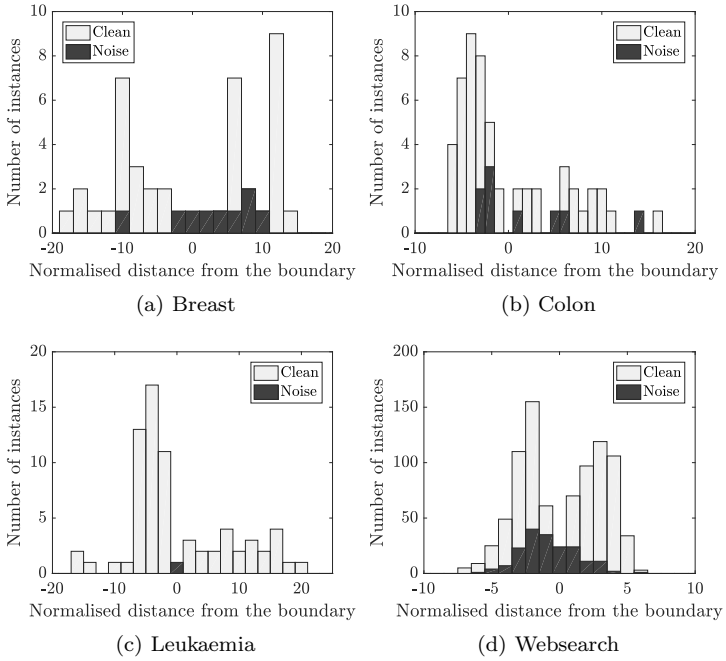

Fig. 7: Characteristics of the label noises in 4 real-world datasets.

4.6 Comparisons with other classifiers on noisy real-world datasets

Real-world dataset can be challenging since we have no control over the characteristics of the noise. In this section we will investigate the algorithms' robustness against real label noise. We use five datasets from various application domains and probably with different sources of labelling errors. For colon cancer dataset the task is to classify the microarray samples as normal tissue (N) or tumour (T). According to the literature [27], the samples T2, T30, T33, T36, T37, N8, N12, N34, N36 may be mislabelled. For breast cancer data, the aim is to distinguish between estrogen positive and estrogen negative observations. Biological evidence [26] suggests that the samples 11, 14, 16, 31, 33, 40, 43, 45, 46 are mislabelled. The objective of leukaemia data analysis is to discriminate between Acute Lymphoblastic Leukaemia (ALL) and Acute Myeloid Leukaemia (AML). Unlike the previous two datasets, the only one suspect mislabelling in this dataset was identified by consensus of previous label noise detection algorithms. The websearch dataset is an image classification datasets where the images were gathered quickly via image search engine. The annotations were also sloppily done by taken the query keyword as the class label directly. Obviously, some of the images returned by the search engine might not relate to the keyword (class label) at all. The websearch dataset

is composed of two classes constructed using keywords 'bike' and 'not bike'. The ground truths for this dataset were manually identified [15]. Lastly, LULC dataset is a land usage classification dataset. The images were obtained from satellite imagery while labels of the land patches were extracted from crowd-sourced OpenStreetMap (OSM) database. We selected two out of six classes, namely 'forest' and 'grass', which are most often mistaken for one another according to the analysis in [3] for our experiment. The LULC dataset was divided into training and testing sets apriori by the creator [3]. The approximate noise rate was found to be around 40% but the exact noisy labels were not identified. The testing labels, however, are all clean. For each of the datasets except breast and LULC, we trained the classifiers on 80% of the data and held out 20% for testing. We used 90/10 train/test splitting for breast dataset due to the small sample size and high dimensional nature of the data. In the case of breast, colon, leukaemia and websearch datasets, mislabelled instances that were split into test set were corrected before testing using the available ground truths. For LULC data, we sampled 80% of the instances in the training set for training, and evaluated the model on the whole test set. Since the test labels of LULC dataset are clean, we did not perform any label correction.

Since the ground truth labels are available (except for LULC dataset), we first summarised the number of correct labels versus noisy labels as a function of their distances from the decision boundary in order to get a better idea of how noise distributions might look like. The plots generated using the weight vector of LR trained in noise-free data are depicted in Fig.7. We observe that the noises spread rather randomly in the case of breast and colon cancer data. Although, according to the literature [26] label noise in breast dataset is more symmetric. That is, the ratio of mislabelling in class ALL and AML are quite similar, and previous theoretical studies seem to suggest that symmetric label noise is relatively harmless [16]. On the other hand, the noise in colon cancer is somewhat asymmetric and we expect that the benefit of label noise modelling, especially the one assuming uniform random noise, will be more obvious in this case. The leukaemia dataset contains only one mislabelling and it should pose little problem to all of the algorithms. Still, it is interesting to see how much label noise aware algorithms would be fooled into believing that some hard-to-classify samples are mislabelled. Lastly, we see that majority of label noise in websearch data occurs near the decision boundary, in which case LaplaceBCN and SVM-linear might be able to do well in this case.

We now present the classification results obtained from 100 experiment repetitions in Table 12. Boldface entries highlight the cases where GMMLR was significantly better, while underlined entries marks the case where GMMLR was worse than its peers as tested by Wilcoxon ranksum test at 0.05 level.

Let us first consider the classifiers' performances in the cases of breast cancer, colon cancer and leukaemia datasets. Statistically, the differences in predictive performances of all models against GMMLR under these three datasets were not significant except for the case of colon cancer data where we spot impressive performance from rLR. We were quite surprised how well the traditional classifiers (LR and SVM) performed on these noisy datasets, and at

Table 12: Mean classification errors (%) and standard errors of the six algorithms on real-world datasets that genuinely contains label noise. Boldface entries highlight the cases where GMMLR was significantly better. Underlined entries mark the cases where GMMLR was worse than its peers as tested by Wilcoxon ranksum test at 0.05 level.

Dataset	Classification model		
	LR	gLR	rLR
breast	15.80 ± 13.42 ≈	15.40 ± 12.98 ≈	16.00 ± 12.71 ≈
colon	10.83 ± 9.29 ≈	10.67 ± 9.41 ≈	7.25 ± 7.55 +
leukaemia	4.07 ± 4.91 ≈	4.20 ± 5.25 ≈	4.13 ± 5.17 ≈
websearch	**20.53 ± 3.12** -	**20.22 ± 3.16** -	18.50 ± 2.94 ≈
LULC	**33.33 ± 1.08** -	**34.27 ± 1.33** -	**34.24 ± 1.36** -
Win-Tie-Loss	2-3-0	2-3-0	1-3-1

Dataset	Classification model		
	LaplaceBCN	SVM-Linear	GMMLR
breast	15.80 ± 13.42 ≈	14.00 ± 13.18 ≈	15.40 ± 13.29
colon	11.08 ± 8.95 ≈	10.08 ± 8.40 ≈	10.25 ± 8.93
leukaemia	4.20 ± 5.16 ≈	4.13 ± 5.08 ≈	4.20 ± 5.16
websearch	18.67 ± 2.95 ≈	16.20 ± 2.72 +	18.91 ± 2.84
LULC	**32.58 ± 1.27** -	**46.12 ± 1.00** -	31.94 ± 0.91
Win-Tie-Loss	1-4-0	1-3-1	n/a

the same time wondered why label noise modelling did not help. We speculate firstly that the sample size and data dimensionality might affect the learning of label noise model such that the estimated noise rates do not reflect the real flipping probabilities when the number of training instances is limited. Indeed, by observing the label flipping probabilities learned by GMMLR in these datasets, we found that they were quite close to zero, making the classifier behaves much like the traditional LR. Meanwhile, rLR which is the simplest label noise model seems to required less training instances compared to other robust models, and hence was able to perform well under colon cancer dataset. Among the three datasets, colon cancer is probably less demanding due to its larger sample size and relatively low dimensionality. This is also where we started to see some slight improvement of GMMLR over LR.

For websearch dataset, we see that SVM-Linear was the top performer. GMMLR was worse than SVM-Linear but not by a large margin. Its performances were comparable to rLR and LaplaceBCN, the models which assume boundary conditioned noise. The improvements of GMMLR over traditional LR was well apparent, and this once again demonstrates the benefit of having the label noise model built into the traditional classifier. Interestingly, SVM-Linear was quite robust to noise around the decision boundary. We think that the slack variables might have helped masking out the negative effect of instances that were hard to classify (which could be the result of flipped labels) to some extent.

In the case where training examples are abundance as in LULC dataset, it turned out that GMMLR can better estimate the noise rates and hence an improved classification accuracy can be expected. We see that GMMLR

outperformed its peers in this particular case. We were surprised by the twisted performance of the SVM which was rather impressive in the previous four datasets. The imbalance of class 'forest' (7431 instances) and class 'grass' (446 instances) might added up to the already severe noise level (estimated to be around 40%). The last row of Table 12 summarises the win-tie-loss counts of GMMLR against other classifiers. We observe that GMMLR was never worse than the traditional LR, gLR and LaplaceBCN, while its winning counts are comparable to SVM-Linear and rLR. Overall, the empirical evidences seem to suggest that it is still a very challenging task to come up with a general label noise model which can do well on all types of label noise. A particular model tends to work best only when the nature of label noise matches its noise assumption. Still, through the series of experiments, we have witnessed that the proposed GMMLR model was quite flexible compared to the existing ones. It was able to counteract the negative effects of various instance-dependent label noises. It also performed reasonable well in the presence of random label noise.

5 Conclusion

In this paper, we have proposed a novel label noise model which is based on a mixture of Gaussians. We incorporated the new noise model into a logistic regression and developed a Gaussian Mixture Model based robust Logistic Regression classifier. Experimental results demonstrated that the added flexibility of the noise function helps counteracting negative effects of wide range of label noises including class-conditional label noise and instance-dependent label noises. Since the current work is still limited to binary classification problems, it is then interesting to investigate the multi-class version of the proposed model in the future.

Acknowledgements

The authors would like to thank anonymous reviewers for constructive comments. This research is financially supported by the Thailand Research Fund (Grant number: MRG59080235). Department of Computer Science, Faculty of Science at Chiang Mai University provides research and computing facilities.

References

1. Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics, 2-7 August 2009, Singapore*, pages 280–287, 2009.
2. Aleksander Kolcz and Gordon V. Cormack. Genre-based decomposition of email class noise. In *SIGKDD'09*, pages 427–436, 2009.
3. Brian A. Johnson and Kotaro Iizuka. Integrating openstreetmap crowdsourced data and landsat time-series imagery for rapid land use/land cover (lulc) mapping: Case study of the laguna de bay area of the philippines. *Applied Geography*, 67:140 – 149, 2016.

4. Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263, 2008.
5. Dan Shen, Jean-David Ruvini, and Badrul Sarwar. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 595–604, New York, NY, USA, 2012. ACM.
6. Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.
7. Benoît Frénay and Michel Verleysen. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.
8. Aditya Krishna Menon, Brendan van Rooyen, and Nagarajan Natarajan. Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751*, 2016.
9. Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *ACML*, volume 20 of *JMLR Proceedings*, pages 97–112. JMLR.org, 2011.
10. Raj S. Chhikara and Jim McKeon. Linear discriminant analysis with misallocation in training samples. *Journal of the American Statistical Association*, 79(388):899–906, 1984.
11. Neil D. Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML'01*, pages 306–313. Morgan Kaufmann, 2001.
12. Yunlei Li, Lodewyk F.A. Wessels, Dick de Ridder, and Marcel J.T. Reinders. Classification in the presence of class noise using a probabilistic kernel Fisher method. *Pattern Recognition*, 40(12):3349–3357, 2007.
13. Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.
14. Jakramate Bootkrajang and Ata Kabán. Label-noise robust logistic regression and its applications. In *ECML-PKDD'12*, pages 143–158, 2012.
15. Jakramate Bootkrajang and Ata Kabán. Learning kernel logistic regression in the presence of class label noise. *Pattern Recognition*, 47(11):3641–3655, 2014.
16. Gábor Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25:79–87, 1992.
17. Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, March 2010.
18. Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS'13*, pages 1196–1204, 2013.
19. Naresh Manwani and P. S. Sastry. Noise tolerance under risk minimization. *IEEE T. Cybernetics*, 43(3):1146–1151, 2013.
20. Aritra Ghosh, Naresh Manwani, and P. S. Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.
21. Peter A. Lachenbruch. Discriminant analysis when the initial samples are misclassified ii: Non-random misclassification models. *Technometrics*, 16(3):pp. 419–424, 1974.
22. Jakramate Bootkrajang. A generalised label noise model for classification in the presence of annotation errors. *Neurocomputing*, 192:61–71, 2016.
23. Jun Du and Zhihua Cai. Modelling class noise with symmetric and asymmetric distributions. In *AAAI*, pages 2589–2595, 2015.
24. Mark Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab, 2005.
25. Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.
26. Mike West, Carrie Blanchette, Holly Dressman, Erich Huang, Seiichi Ishida, Rainer Spang, Harry Zuzan, John A. Olson Jr., Jeffrey R. Marks, and Joseph R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA*, 98(20):pp. 11462–11467, 2001.

27. U. Alon, N. Barkai, D. A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A. J. Levine. 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, 1999.
28. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
29. Andrew Frank, Arthur Asuncion, et al. Uci machine learning repository, 2010.
30. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, December 2006.
31. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.