Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Modelling Class Noise with Symmetric and Asymmetric Distributions

Jun Du
School of Computer Science
China University of Geosciences
Wuhan, P. R. China, 430074
dr.jundu@gmail.com

Zhihua Cai
School of Computer Science
China University of Geosciences
Wuhan, P. R. China, 430074
zhcai@cug.edu.cn

Abstract

In classification problem, we assume that the samples around the class boundary are more likely to be incorrectly annotated than others, and propose *boundary-conditional class noise* (BCN). Based on the BCN assumption, we use unnormalized Gaussian and Laplace distributions to directly model *how class noise is generated*, in symmetric and asymmetric cases. In addition, we demonstrate that Logistic regression and Probit regression can also be reinterpreted from this class noise perspective, and compare them with the proposed models. The empirical study shows that, the proposed asymmetric models overall outperform the benchmark linear models, and the asymmetric Laplace-noise model achieves the best performance among all.

Introduction

Take handwritten digit recognition for example. During the process of manual annotation (to obtain the *labelled* training data), it is fairly easy to correctly annotate legible handwritten digits, whereas mistakes are often made on the ambiguous ones. Therefore, given a set of *annotated* digits, it is reasonable to assume that, the legible samples are often correctly labelled, whereas incorrect labels are more likely to be attached to the ambiguous samples. Similar assumptions can also be applied to many other real-world applications, such as, speech recognition, spam filter, etc.

This assumption can be formalized in a more general manner: We denote by y the observed corrupted label, and by y^t the unknown true label, in a classification problem. Given a sample $\boldsymbol{x}$, we assume that, $p(y \neq y^t|\boldsymbol{x})$ *tends to be low when $\boldsymbol{x}$ is far from the class boundary (like the legible digits), and tends to become higher when $\boldsymbol{x}$ gets closer (like the ambiguous digits).* We call this type of noise *boundary-conditional class noise* (abbreviated BCN).

Based on the BCN assumption, instead of modelling how data is generated (i.e., $p(\boldsymbol{x}, y)$, as in generative learning), or modelling conditional class probability directly (i.e., $p(y|\boldsymbol{x})$, as in discriminative learning), in this paper, we propose to *model how this boundary-conditional class noise is generated (i.e., $p(y \neq y^t|\boldsymbol{x})$).* More specifically, we assume that the class noise $p(y \neq y^t|\boldsymbol{x})$ is distributed as an *unnormalized Gaussian* and an *unnormalized Laplace* centred at the linear class boundary, and propose Gaussian-noise model and Laplace-noise model respectively. These two models are then further adapted to asymmetric cases.

Given this class noise perspective, we also reinterpret Logistic regression and Probit regression by using class noise probability $p(y^t \neq y|\boldsymbol{x})$. These two models are also adapted to asymmetric cases. Demonstrations are made to compare Logistic regression and Probit regression with the proposed class noise models.

Empirical study is conducted on synthetic data and real-world UCI (Bache and Lichman 2013) data. The experimental results demonstrate that the asymmetric models overall outperform the benchmark linear models (including Logistic regression, Probit regression, and LinearSVM with L1 and L2 loss). In addition, the proposed asymmetric Laplace-noise model achieves the best performance among all.

Related Work

Class noise has been extensively studied in machine learning community. A recent comprehensive survey can be found in (Frénay and Verleysen 2014). In general, researchers use different strategies to solve the problem: Some aim to identify and eliminate mislabelled samples, as in (Brodley and Friedl 1999; Zhu, Wu, and Chen 2003); some tend to obtain more data or re-weight samples to improve data quality, as in (Sheng, Provost, and Ipeirotis 2008; Rebbapragada and Brodley 2007); others make certain assumptions on class noise and build noise-tolerant models, as in (Angluin and Laird 1988; Dawid and Skene 1979; Lawrence and Schölkopf 2001; Raykar et al. 2010; Natarajan, Dhillon, and Ravikumar 2013). Our work falls into the last category.

More specifically, (Angluin and Laird 1988) proposed a simple class noise framework *random classification noise* (RCN), which has been commonly used thereafter. The more flexible *class-conditional class noise* (CCN) framework has also been extensively studied, as in (Lawrence and Schölkopf 2001; Raykar et al. 2010; Natarajan, Dhillon, and Ravikumar 2013). However, both of these two frameworks only consider *sample-independent* class noise, which imposes a rigid constraint on real-world applications. In contrast, motivated by real-world observations, the *boundary-*

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

conditional class noise (BCN) proposed in this paper considers *sample-dependent* class noise, which leads to more realistic and flexible models. (See Section "Problem Formalization" for the comparison based on the graphic model representations.)

In addition, in most existing research, class noise is handled in an additional procedure on top of generative or discriminative models. In comparison, this paper aims to provide a novel perspective to (1) directly build discriminative models through modelling class noise distributions, and (2) reinterpret existing discriminative models from the class noise perspective.

Asymmetric distributions have also been studied previously, as in (Bennett 2003; Kato, Omachi, and Aso 2002). But as far as we know, little work has been done to directly model class noise with asymmetric distributions.

Assumption Verification

In this section, we verify the BCN assumption on a real-world data set.

(Snow et al. 2008) conducted a series of human linguistic annotation experiments on Amazon's Mechanical Turk. The purpose of these experiments is to compare the AMT non-expert annotations with the gold standard labels provided by experts. One annotation experiment is *affective text analysis*[1]. More specifically, each non-expert annotator is presented with a list of short headlines, and is asked to provide numeric ratings in the interval $[-100, 100]$ to denote the overall emotional valence. 100 headline samples are selected, and 10 non-expert annotations are collected for each of them. In addition, the gold standard labels for these 100 samples (also in the interval $[-100, 100]$) are also provided for comparison. Note that, the provided numeric rating represents the *degree of valence*, where 100 and -100 indicate strong positive and strong negative respectively.

In the class noise setting, we regard the non-expert annotations as the corrupted labels, and the gold standard labels as the true ones. As the emotional state (positive or negative valence) is only reflected by the sign of the rating, we define $y = sign(\text{non-expert annotation}), y^t = sign(\text{gold standard label})$ where $sign(.)$ is the sign function, and the function values 1, -1, and 0 represent positive, negative, and neutral valence respectively. We further define the annotation noise rate for a given headline as:

$Noise\_Rate = p(y \neq y^t | headline) = \frac{\#annotations(y \neq y^t)}{\#annotations}$.

We plot the annotation noise rate for each headline in Figure 1, and see how it changes with the gold standard label. Note that, Gold Standard Label $= 0$ can be regarded as the boundary to discriminate positive and negative valence.

We can see clearly from Figure 1 that, the annotation noise rate peaks when the gold standard label is about 0 (i.e., at around boundary), and it decreases when the true label goes up towards 100 or goes down towards -100 (i.e., gets far away from the boundary). This observation matches our intuition that, the annotators are more likely to provide the true label when the headline is either strongly positive or strongly negative, while the incorrect labels are provided more often when the headline leans to neural. More importantly, this real-world data set clearly verifies our BCN assumption.

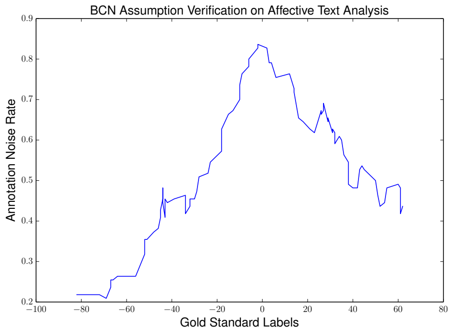
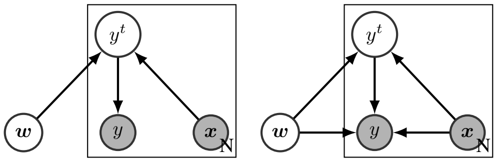


Figure 1: BCN Assumption on Affective Text Analysis.[2]

Problem Formalization

We consider only linear models for binary classification in this paper. In the class noise setting, we denote by y the observed (corrupted) label and by y^t the true (hidden) label, where $y, y^t \in \{0, 1\}$. We denote by $\boldsymbol{w}$ the model parameter. In the linear model case, the goal is to build a linear classification boundary $\boldsymbol{w}^T\boldsymbol{x} = 0$[3], such that $y^t = I(\boldsymbol{w}^T\boldsymbol{x} \geq 0)$, where $I(.)$ is the indicator function (which equals to 1 if its argument is true and 0 otherwise).



(a) RCN / CCN Framework (b) BCN Framework

Figure 2: Class Noise Frameworks

Figure 2(a) shows the framework of the most commonly used *random classification noise* (RCN) model and *class-conditional class noise* (CCN) model. Given a set of N corrupted samples, both $\boldsymbol{x}$ and y are observed, whereas y^t is hidden, and $\boldsymbol{w}$ is to be learned. y^t depends on $\boldsymbol{x}$ and $\boldsymbol{w}$, and y depends on y^t only. In both of these two frameworks, as long as y^t is determined, y is no longer affected by $\boldsymbol{x}$ or $\boldsymbol{w}$. That is, $p(y|y^t, \boldsymbol{x}, \boldsymbol{w}) = p(y|y^t)$.

In contrast, Figure 2(b) shows the framework of the proposed *boundary-conditional class noise* (BCN) model, where two extra links connecting $\boldsymbol{x}$ and $\boldsymbol{w}$ to y are added. In this case, y also depends on both $\boldsymbol{x}$ and $\boldsymbol{w}$, therefore $p(y|y^t, \boldsymbol{x}, \boldsymbol{w}) \neq p(y|y^t)$.

Instead of directly modelling $p(y|y^t, \boldsymbol{x}, \boldsymbol{w})$, we model class noise $p(y \neq y^t|\boldsymbol{x})$, where $\boldsymbol{w}$ is omitted to keep the notation uncluttered. The BCN assumption can be further formalized, which imposes the constraints on $p(y \neq y^t|\boldsymbol{x})$:

[1] The raw annotation data is provided at $https://sites.google.com/site/nlpannotations/$.

[2] For better demonstration, the annotation noise rate plotted in Figure 1 is smoothed by neighbourhood averaging with radius 5.

[3] Here $\boldsymbol{x}$ contains all the original features plus one dummy feature with constant value 1.

Assumption 1 (BCN).
Given two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:
If $\boldsymbol{w}^T\boldsymbol{x}_i = \boldsymbol{w}^T\boldsymbol{x}_j$, then $p(y_i \neq y_i^t|\boldsymbol{x}_i) = p(y_j \neq y_j^t|\boldsymbol{x}_j)$;
If $sign(\boldsymbol{w}^T\boldsymbol{x}_i) = sign(\boldsymbol{w}^T\boldsymbol{x}_j)$ and $|\boldsymbol{w}^T\boldsymbol{x}_i| > |\boldsymbol{w}^T\boldsymbol{x}_j|$, then $p(y_i \neq y_i^t|\boldsymbol{x}_i) < p(y_j \neq y_j^t|\boldsymbol{x}_j)$.

Under the BCN assumption, we start from defining $p(y \neq y^t|\boldsymbol{x})$ in both symmetric and asymmetric cases. $p(y|\boldsymbol{x})$ can therefore be further derived. (See Section "Representation" for details.) We then set up loss functions based on maximum likelihood and maximum a posteriori, and propose learning algorithms to optimize $\boldsymbol{w}$. (See Section "Learning" for details.) Given the optimal $\boldsymbol{w}^*$, the predictions can then be made from $y^t = I({\boldsymbol{w}^*}^T\boldsymbol{x} \geq 0)$.

Model Representation

We define $p(y \neq y^t|\boldsymbol{x})$ and further model $p(y|\boldsymbol{x})$ in this section. More specifically, given the linear classification boundary $\boldsymbol{w}^T\boldsymbol{x} = 0$, and $y^t = I(\boldsymbol{w}^T\boldsymbol{x} \geq 0)$, we can have:

$$p(y=1|\boldsymbol{x}) = \sum_{y^t} p(y=1|y^t,\boldsymbol{x})p(y^t|\boldsymbol{x})$$

$$= \begin{cases} p(y=1|y^t=0,\boldsymbol{x}) \\ p(y=1|y^t=1,\boldsymbol{x}) \end{cases} = \begin{cases} p(y^t \neq y|\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0 \\ 1 - p(y^t \neq y|\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0 \end{cases}$$

Symmetric Case

We discuss *symmetric class noise distribution* in this subsection. More specifically, in addition to Assumption 1, we further assume that:

Assumption 2 (BCN_Symmetric).
Given two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:
If $\boldsymbol{w}^T\boldsymbol{x}_i = -\boldsymbol{w}^T\boldsymbol{x}_j$, then $p(y \neq y^t|\boldsymbol{x}_i) = p(y \neq y^t|\boldsymbol{x}_j)$.

Intuitively, this additional assumption indicates that, the samples that have the same distance to the boundary but *located on different sides* are equally likely to be corrupted. The class noise distribution is therefore symmetric w.r.t. the class boundary.

According to Assumptions 1 and 2, we first propose two class noise models: symmetric Gaussian-noise model and symmetric Laplace-noise model, and then reinterpret traditional Logistic regression and Probit regression from this class noise perspective.

More specifically, we suppose that the linear classification boundary is $\boldsymbol{w}^T\boldsymbol{x} = 0$, and assume that the class noise is distributed as an *unnormalized Gaussian* centred at the linear boundary with variance σ^2. We therefore have $p(y \neq y^t|\boldsymbol{x}) = a\exp(-\frac{(\boldsymbol{w}^T\boldsymbol{x})^2}{2\sigma^2}) = a\exp(-(\frac{\boldsymbol{w}^T\boldsymbol{x}}{\sqrt{2}\sigma})^2)$.

Given that $p(y \neq y^t|\boldsymbol{x})$ reaches its maximum 0.5 on the linear boundary (where $\boldsymbol{w}^T\boldsymbol{x} = 0$), we have $a = 0.5$. In addition, $\sqrt{2}\sigma$ can be absorbed into $\boldsymbol{w}$ while still keeping the boundary unchanged. The symmetric Gaussian-noise model then can be represented by both $p(y \neq y^t|\boldsymbol{x})$ and $p(y = 1|\boldsymbol{x})$, as shown in Table 1.

Similarly, assuming an *unnormalized Laplace* centred at the linear boundary for class noise, symmetric Laplace-noise model can also be represented by $p(y \neq y^t|\boldsymbol{x})$ and $p(y = 1|\boldsymbol{x})$; see Table 1 for details.

As traditional discriminative models, both Logistic regression and Probit regression have been commonly used to directly model $p(y|\boldsymbol{x})$. It turns out that they can also be intuitively reinterpreted from class noise perspective by $p(y \neq y^t|\boldsymbol{x})$; also see Table 1 for details.

For better demonstration, Figure 3 compares the above four symmetric models from class noise and conditional class probability perspectives, both in $1D$ case (where $x = 0$ is set as the class boundary).

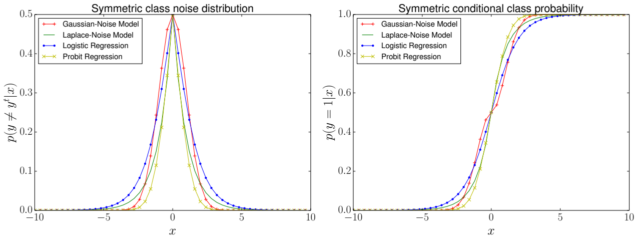


Figure 3: Symmetric class noise distributions and conditional class probabilities for all the four models, in 1D sample space where $x = 0$ is the boundary.

Asymmetric Case

We argue that, Assumption 2 (which leads to the symmetric class noise distributions) may *not* hold in some cases. Intuitively, given two samples located on different sides of the boundary, even if they are equally far away from the boundary, they are still likely to be corrupted with different probabilities.

Therefore, in this subsection, we consider only the constraints from Assumption 1, and discuss *asymmetric class noise distributions*. Note that, the asymmetric case can be considered as a more generic scenario, where symmetric distributions are only special cases (when Assumption 2 also holds).

Similar to the previous section, we suppose that the linear classification boundary is $\boldsymbol{w}^T\boldsymbol{x} = 0$. To accommodate the asymmetry property, we introduce a scale parameter λ ($\lambda > 0$), and assume that the class noise is distributed as an *unnormalized Gaussian* with variance $(\sigma/\lambda)^2$. We then have $p(y \neq y^t|\boldsymbol{x}) = \frac{1}{2}\exp(-\frac{(\boldsymbol{w}^T\boldsymbol{x})^2}{2(\sigma/\lambda)^2}) = \frac{1}{2}\exp(-(\lambda \cdot \frac{\boldsymbol{w}^T\boldsymbol{x}}{\sqrt{2}\sigma})^2)$.
We again have $\sqrt{2}\sigma$ absorbed into $\boldsymbol{w}$ while still keeping the boundary unchanged, and end up with
$p(y \neq y^t|\boldsymbol{x}) = \frac{1}{2}\exp(-(\lambda\boldsymbol{w}^T\boldsymbol{x})^2)$.

Asymmetric property can be implemented by setting different λ on different sides of the class boundary. Alternatively, as re-scaling λ is equivalent to re-scaling $\boldsymbol{w}$, we can constrain $\lambda = 1$ on one side of the class boundary, and keep λ as a free parameter on the other side. Consequently, the asymmetric Gaussian-noise models can also be represented by both $p(y \neq y^t|\boldsymbol{x})$ and $p(y = 1|\boldsymbol{x})$, as shown in Table 1.

Similarly, $p(y \neq y^t|\boldsymbol{x})$ and $p(y = 1|\boldsymbol{x})$ for asymmetric Laplace-noise model, asymmetric Logistic regression and asymmetric Probit regression are also summarized in Table 1.

For better demonstration, Figure 4 compares the above four asymmetric models from class noise and conditional class probability perspectives, where λ is set to 0.3.

Table 1: Summary of Symmetric and Asymmetric Models

	Model	$p(y \neq y^t \vert \boldsymbol{x})$	$p(y = 1 \vert \boldsymbol{x})$
Symmetric	Gaussian-Noise Model	$\frac{1}{2}\exp(-(\boldsymbol{w}^T\boldsymbol{x})^2)$	$\begin{cases}\frac{1}{2}\exp(-(\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ 1-\frac{1}{2}\exp(-(\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0\end{cases}$
	Laplace-Noise Model	$\frac{1}{2}\exp(-\vert\boldsymbol{w}^T\boldsymbol{x}\vert)$	$\begin{cases}\frac{1}{2}\exp(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ 1-\frac{1}{2}\exp(-\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$
	Logistic Regression	$\frac{1}{1+\exp(\vert\boldsymbol{w}^T\boldsymbol{x}\vert)}$	$\frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x})}$
	Probit Regression	$\begin{cases}\int_{-\infty}^{\boldsymbol{w}^T\boldsymbol{x}} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \int_{\boldsymbol{w}^T\boldsymbol{x}}^{+\infty} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$	$\int_{-\infty}^{\boldsymbol{w}^T\boldsymbol{x}} \mathcal{N}(\theta\vert 0,1)d\theta$
Asymmetric	Gaussian-Noise Model	$\begin{cases}\frac{1}{2}\exp(-(\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \frac{1}{2}\exp(-(\lambda\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$	$\begin{cases}\frac{1}{2}\exp(-(\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ 1-\frac{1}{2}\exp(-(\lambda\boldsymbol{w}^T\boldsymbol{x})^2) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$
	Laplace-Noise Model	$\begin{cases}\frac{1}{2}\exp(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \frac{1}{2}\exp(-\lambda\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$	$\begin{cases}\frac{1}{2}\exp(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ 1-\frac{1}{2}\exp(-\lambda\boldsymbol{w}^T\boldsymbol{x}) & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$
	Logistic Regression	$\begin{cases}\frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x})} & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \frac{1}{1+\exp(\lambda\boldsymbol{w}^T\boldsymbol{x})} & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$	$\begin{cases}\frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x})} & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \frac{1}{1+\exp(-\lambda\boldsymbol{w}^T\boldsymbol{x})} & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$
	Probit Regression	$\begin{cases}\int_{-\infty}^{\boldsymbol{w}^T\boldsymbol{x}} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \int_{\lambda\boldsymbol{w}^T\boldsymbol{x}}^{+\infty} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$	$\begin{cases}\int_{-\infty}^{\boldsymbol{w}^T\boldsymbol{x}} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0, \\ \int_{-\infty}^{\lambda\boldsymbol{w}^T\boldsymbol{x}} \mathcal{N}(\theta\vert 0,1)d\theta & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0.\end{cases}$

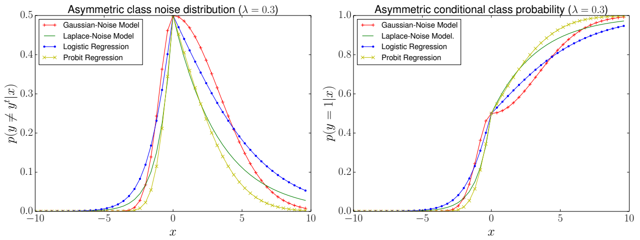


Figure 4: Asymmetric class noise distributions and conditional class probabilities for all the four models, in 1D sample space where $x = 0$ is the boundary.

Model Learning

In the previous section, we have proposed several algorithms to model $p(y|\boldsymbol{x})$ (parametrized by $\boldsymbol{w}$ in the symmetric case, and by $\boldsymbol{w}$ and λ in the asymmetric case). In this section, we discuss the learning algorithm to optimize the parameters given a set of training samples.

Maximum Likelihood (ML) Estimation

We use maximum likelihood to optimize the parameters in this subsection. More specifically, the negative log conditional likelihood is used as the loss function: $L_{ML}(\boldsymbol{w}, \lambda) = -\sum_{i=1}^{N}\{y_i \log h(\boldsymbol{x}_i) + (1-y_i)\log(1-h(\boldsymbol{x}_i))\}$ where N is the number of all the training samples, $\boldsymbol{x}_i$ and y_i are the ith training sample and the corresponding observed (corrupted) label respectively, and $h(\boldsymbol{x}_i) \triangleq p(y_i = 1|\boldsymbol{x}_i)$ (where parameters $\boldsymbol{w}$ and λ are omitted to keep the notation uncluttered).

It can be shown that, in the symmetric case, the loss function is differentiable w.r.t. $\boldsymbol{w}$ for all the four models, the traditional gradient methods therefore can be directly applied. In the asymmetric case, however, the loss function is no longer differentiable w.r.t. $\boldsymbol{w}$ and λ (when $\boldsymbol{w}^T\boldsymbol{x} = 0$), we then apply *subgradient methods* for optimization. The subgradients[4] (w.r.t. $\boldsymbol{w}$ and λ) for the four asymmetric models are summarized in Table 2.

With the extra scale parameter λ, the asymmetric models are expected to have lower bias thus achieving lower (or at least equal) loss function values (compared to their symmetric counterparts). However, subgradient methods might still stuck at local minima with relatively high loss function values.

To overcome this limitation, we *initialize the parameters to the more sensible values*. Specifically, for all the asymmetric models, we initialize λ to 1, and $\boldsymbol{w}$ to the final solutions of the symmetric counterparts. Subgradient methods then can be applied. The optimization might still end up with local minima, however, it is guaranteed that lower (or at least equal) loss function values can be achieved. See Algorithm 1 for details.

Algorithm 1 Learning asymmetric models

```
Set $\lambda = 1$
Initialize $\boldsymbol{w}$ randomly
Apply gradient method to optimize $\boldsymbol{w}$:
    $\boldsymbol{w}_{start} = \arg\min_{\boldsymbol{w}} L(\boldsymbol{w}, \lambda = 1)$
Initialize $\lambda = 1$
Initialize $\boldsymbol{w} = \boldsymbol{w}_{start}$
Apply subgradient method to optimize $\boldsymbol{w}$ and $\lambda$:
    $\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} L(\boldsymbol{w}, \lambda)$
    $\lambda^\star = \arg\min_{\lambda} L(\boldsymbol{w}, \lambda)$
```

Maximum A Posteriori (MAP) Estimation

We use maximum a posteriori to optimize the parameters in this subsection.

More specifically, we assume a zero mean isotropic Gaussian prior on $\boldsymbol{w}$: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$, where α is the precision parameter, and $\boldsymbol{I}$ is the identity matrix.

[4] For simplicity, all the subgradients are derived for one training sample $(\boldsymbol{x}, y)$; the extension to the entire training set is straightforward thus omitted.

Table 2: Subgradients of negative log likelihood for asymmetric models

Asymmetric	$\frac{\partial L_{ML}}{\partial \boldsymbol{w}}$	$\frac{\partial L_{ML}}{\partial \lambda}$
Gaussian-Noise Model	$\begin{cases} \frac{y-h(\mathbf{x})}{1-h(\mathbf{x})} \cdot 2\mathbf{w}^T\mathbf{x} \cdot \mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})} \cdot 2\lambda^2\mathbf{w}^T\mathbf{x} \cdot \mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$	$\begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})} \cdot 2\lambda(\mathbf{w}^T\mathbf{x})^2 & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$
Laplace-Noise Model	$\begin{cases} \frac{h(\mathbf{x})-y}{1-h(\mathbf{x})} \cdot \mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})} \cdot \lambda\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$	$\begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})} \cdot \mathbf{w}^T\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$
Logistic Regression	$\begin{cases} (h(\mathbf{x})-y) \cdot \mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ (h(\mathbf{x})-y) \cdot \lambda\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$	$\begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ (h(\mathbf{x})-y) \cdot \mathbf{w}^T\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$
Probit Regression	$\begin{cases} \frac{h(\mathbf{x})-y}{h(\mathbf{x})(1-h(\mathbf{x}))} \cdot \mathcal{N}(\mathbf{w}^T\mathbf{x} \mid 0,1) \cdot \mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})(1-h(\mathbf{x}))} \cdot \mathcal{N}(\lambda\mathbf{w}^T\mathbf{x} \mid 0,1) \cdot \lambda\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$	$\begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} < 0, \\ \frac{h(\mathbf{x})-y}{h(\mathbf{x})(1-h(\mathbf{x}))} \cdot \mathcal{N}(\lambda\mathbf{w}^T\mathbf{x} \mid 0,1) \cdot \mathbf{w}^T\mathbf{x} & \text{if } \mathbf{w}^T\mathbf{x} \geq 0. \end{cases}$

We also assume a Gamma prior (with shape parameter α' and scale parameter β') on λ: $p(\lambda) = Gamma(\alpha', \beta')$. We further set the mode of the Gamma prior to 1, such that the symmetric class-noise models (where $\lambda = 1$) are preferred: $\frac{\alpha'-1}{\beta'} = 1 \Rightarrow \alpha' = \beta' + 1$. The prior on λ therefore can be formulated: $p(\lambda) = Gamma(\beta + 1, \beta)$ where $\beta \triangleq \beta' > 0$.

By further assuming the priors of $\boldsymbol{w}$ and λ are independent, the loss function (i.e., negative log posterior) can be formulated:

$$L_{MAP}(\boldsymbol{w}, \lambda) = -\sum_{i=1}^{N}\{y_i \log h(\boldsymbol{x}_i) + (1 - y_i)\log(1 - h(\boldsymbol{x}_i))\} + \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + \beta(\lambda - \ln\lambda)$$

Similar to the previous subsection, the gradient and subgradient methods are used to optimize the parameters for symmetric and asymmetric models respectively. The subgradients of negative log posterior (L_{MAP}) can be derived: $\frac{\partial L_{MAP}}{\partial \boldsymbol{w}} = \frac{\partial L_{ML}}{\partial \boldsymbol{w}} + \alpha\boldsymbol{w}$, $\frac{\partial L_{MAP}}{\partial \lambda} = \frac{\partial L_{ML}}{\partial \lambda} + \beta(1 - 1/\lambda)$.

Empirical Study

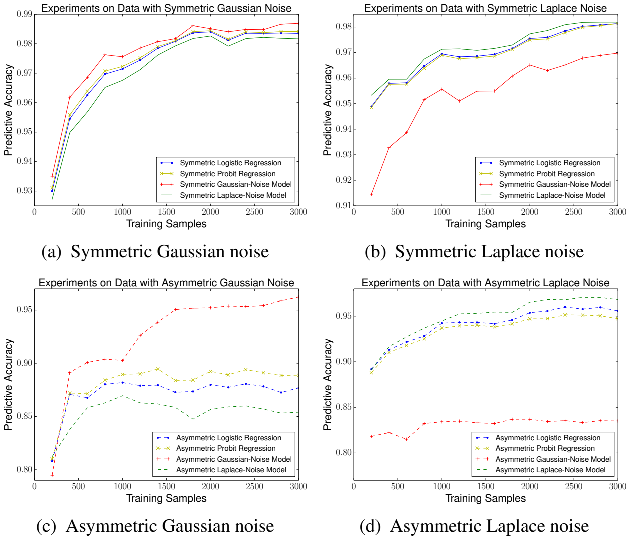
We conduct empirical study in this section. More specifically, experiments are conducted on the synthetic data with injected class noise in Section “Synthetic Data”, and on real-world UCI data sets in Section “UCI Data”.

Synthetic Data

To better observe the behaviour of the proposed models, in this subsection, we conduct experiments on synthetic data with various injected class noise.

A set of 2-D data ($\boldsymbol{x} = (x_1, x_2)$, and $x_1, x_2 \in [-1, 1]$) is generated randomly. The true class labels y^t are produced by setting the true class boundary as $x_1 + x_2 = 0$. The corrupted class labels y are further produced, by injecting four types of class noise, namely *symmetric Gaussian noise*, *symmetric Laplace noise*, *asymmetric Gaussian noise*, and *asymmetric Laplace noise*, according to Section “Model Representation”. λ is set to 0.3 in the asymmetric cases.

Eight models (four symmetric and four asymmetric, as in Table 1) with ML estimation are built on 200 to $3,000$ training samples with corrupted labels (i.e., $(\boldsymbol{x}, y)$ tuples)[5], and are tested on $10,000$ test samples with *true* labels (i.e., $(\boldsymbol{x}, y^t)$ tuples). The process is repeated 10 times, and the average predictive accuracies on the test data are recorded for comparison.



(a) Symmetric Gaussian noise

(b) Symmetric Laplace noise

(c) Asymmetric Gaussian noise

(d) Asymmetric Laplace noise

Figure 5: Experiments on synthetic data

Figure 5 shows the experimental results[6] on the four corrupted data sets. It can be observed that, when a certain type of noise is injected, the corresponding proposed algorithm always performs the best. More specifically, symmetric Gaussian-noise model, symmetric Laplace-noise model, asymmetric Gaussian-noise model, and asymmetric Laplace-noise model all have the best predictive performance in Figures 5(a), 5(b), 5(c) and 5(d) respectively. This clearly demonstrates the advantages of the proposed algorithms with certain types of noise.

UCI Data

We also conduct the experiments on 22 data sets from UCI Machine Learning Repository (Bache and Lichman 2013).

[5] We vary the training data size to make more reliable experimental observations.

[6] In the experiments, the symmetric models usually outperform their asymmetric counterparts when symmetric noise is injected, whereas the asymmetric models overall work significantly better with asymmetric noise. For better demonstration, we plot symmetric models only in Figures 5(a) and 5(b), and asymmetric models only in Figures 5(c) and 5(d).

Table 3: Predictive accuracies on 22 UCI data sets

		Benchmark Models				Asymmetric Models			
Dataset	Note	Logistic	Probit	L1-SVM	L2-SVM	Logistic	Probit	Gaussian-Noise	Laplace-Noise
biodegradation	-	0.8711	0.8690	**0.8753**	0.8735	0.8721	0.8705	0.8665	0.8749
cardiotocography	-	0.9895	0.9891	0.9882	0.9892	0.9898	0.9896	0.9895	**0.9900**
ILPD	-	0.7218	0.7254	0.7150	0.7251	0.7285	**0.7344**	0.7207	0.7242
ionosphere	-	0.8914	0.8923	0.8889	0.8897	0.8929	**0.8946**	0.8943	0.8909
letter_recognition_uv	Class “u” vs Class “v”	0.9957	0.9953	0.9941	0.9950	0.9961	0.9958	0.9949	**0.9966**
magic04	-	0.7910	0.7901	0.7918	0.7893	0.7911	0.7902	0.7862	**0.7929**
mammographic_masses	-	0.8277	0.8265	0.8299	0.8265	**0.8354**	0.8349	0.8267	0.8348
optdigits_46	Class “4” vs Class “6”	0.9936	0.9935	0.9930	0.9927	0.9942	0.9941	0.9941	**0.9944**
pendigit_46	Class “4” vs Class “6”	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
pima-indians-diabetes	-	0.7754	0.7743	0.7728	0.7728	0.7764	0.7758	0.7695	**0.7770**
pop_failures	-	0.9611	0.9607	0.9611	0.9591	0.9628	0.9615	0.9554	**0.9631**
sat_1vsRest	Class “1” vs rest	0.9870	0.9868	0.9867	0.9872	0.9871	0.9868	0.9864	**0.9873**
segment_1vsRest	Class “1” vs rest	**0.9978**	**0.9978**	**0.9978**	**0.9978**	**0.9978**	**0.9978**	**0.9978**	**0.9978**
semeion_46	Class “4” vs Class “6”	0.9929	0.9932	0.9938	0.9960	0.9947	0.9957	**0.9963**	0.9950
sensor_readings_24_ForwardvsRest	Class “Forward” vs rest	0.7599	0.7590	0.7548	0.7600	0.7673	0.7663	0.7577	**0.7701**
sonar	-	0.7804	0.7841	0.7908	0.7778	0.7953	**0.7958**	0.7850	0.7952
spambase	-	0.9282	0.9262	0.9289	0.9261	0.9290	0.9274	0.9193	**0.9315**
transfusion	-	0.7735	0.7735	0.7635	0.7725	0.7817	0.7778	0.7743	**0.7844**
vertebal_2c	-	0.8503	0.8497	0.8545	0.8526	0.8674	**0.8681**	0.8603	0.8668
wdbc	-	0.9812	0.9808	0.9773	0.9784	0.9819	0.9812	0.9777	**0.9826**
winequality-red_6vsRest	Class “6” vs rest	0.6012	0.6004	0.5996	0.5917	0.6203	0.6193	0.6015	**0.6204**
winequality-white_6vsRest	Class “6” vs rest	0.5677	0.5676	0.5515	0.5654	0.5707	0.5697	0.5485	**0.5747**

Table 4: T-test summary w/t/l on 22 UCI data sets

		Benchmark Models				Asymmetric Models			
		Logistic	Probit	L1-SVM	L2-SVM	Logistic	Probit	Gaussian-Noise	Laplace-Noise
Benchmark Models	Logistic	0/22/0	6/15/1	8/11/3	8/13/1	0/5/17	1/10/11	10/8/4	0/7/15
	Probit	1/15/6	0/22/0	7/10/5	3/17/2	0/4/18	0/4/18	8/11/3	0/6/16
	L1-SVM	3/11/8	5/10/7	0/22/0	5/9/8	2/8/12	3/6/13	5/9/8	0/6/16
	L2-SVM	1/13/8	2/17/3	8/9/5	0/22/0	0/8/14	0/9/13	7/10/5	0/7/15
Asymmetric Models	Logistic	17/5/0	18/4/0	12/8/2	14/8/0	0/22/0	7/13/2	16/5/1	1/14/7
	Probit	11/10/1	18/4/0	13/6/3	13/9/0	2/13/7	0/22/0	16/6/0	2/10/10
	Gaussian-Noise	4/8/10	3/11/8	8/9/5	5/10/7	1/5/16	0/6/16	0/22/0	1/6/15
	Laplace-Noise	**15/7/0**	**16/6/0**	**16/6/0**	**15/7/0**	**7/14/1**	**10/10/2**	**15/6/1**	**0/22/0**

On all the data sets, categorical features are converted to binary (numeric), samples with missing values are removed, duplicate features are removed, and multiple class labels are converted / reduced to binary. (See Column “Note” in Table 3)

Four asymmetric models with MAP estimation are tested on each data set. In comparison, we also present the results on four benchmark linear models: Logistic regression, Probit regression, L1-SVM (Linear SVM with hinge loss, (Fan et al. 2008)), and L2-SVM (Linear SVM with squared hinge loss, (Fan et al. 2008)), all with L2 regularization.

Grid-search on regularization coefficients using 10-fold cross-validation is applied (Hsu, Chang, and Lin 2010). More specifically, for each model, the regularization coefficients are chosen from $\{2^{-5}, 2^{-3}, \cdots, 2^{13}, 2^{15}\}$, and only the ones with the best CV accuracy are picked. The whole process is repeated 10 times on each data set, and the average predictive accuracies are recorded for comparison.

The average predictive accuracies are shown in Table 3, where the highest accuracy on each data set is highlighted in bold. The overall t-test (paired t-test with 95% significance level) results are also shown in Table 4, where the “w/t/l” in each cell indicates that the algorithm in the corresponding row wins on “w”, ties on “t”, and loses on “l” data sets, in comparison with the algorithm in the corresponding column.

It can be observed from Tables 3 and 4 that, the proposed asymmetric models have overall superior predictive performance compared to the benchmark models. In addition, the asymmetric Laplace-noise model clearly performs the best among all the eight models.

Conclusions

To summarize, we assume that the samples around the class boundary are more likely to be corrupted than others, and propose *boundary-conditional class noise* (BCN). We design Gaussian-noise models and Laplace-noise models to directly model how BCN is generated. Both Logistic regression and Probit regression are reinterpreted from this class noise prospective, and all the models are further adapted to asymmetric cases. The empirical study shows that, the proposed asymmetric models overall outperform the benchmark linear models, and the asymmetric Laplace-noise model achieves the best performance among all.

References

Angluin, D., and Laird, P. 1988. Learning From Noisy Examples. *Machine Learning* 343–370.

Bache, K., and Lichman, M. 2013. UCI machine learning repository.

Bennett, P. N. 2003. Using Asymmetric Distributions to Improve Text Classifier Probability Estimates. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 111–118.

Brodley, C. E., and Friedl, M. A. 1999. Identifying Mislabeled Training Data. *Journal of Artificial Intelligence Research* 11:131–167.

Dawid, A. P., and Skene, A. M. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28(1):20–28.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9:1871–1874.

Frénay, B., and Verleysen, M. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* 25(5):845–869.

Hsu, C.-w.; Chang, C.-c.; and Lin, C.-j. 2010. A Practical Guide to Support Vector Classification.

Kato, T.; Omachi, S.; and Aso, H. 2002. Asymmetric Gaussian and Its Application to Pattern Recognition. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*. 405–413.

Lawrence, N. D., and Schölkopf, B. 2001. Estimating a Kernel Fisher Discriminant in the Presence of Label Noise. In *Proceedings of 18th International Conference on Machine Learning*, 306–313.

Natarajan, N.; Dhillon, I. S.; and Ravikumar, P. 2013. Learning with Noisy Labels. In *Advances in Neural Information Processing Systems (NIPS)*, 1196–1204.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning From Crowds. *Journal of Machine Learning Research* 11:1297–1322.

Rebbapragada, U., and Brodley, C. 2007. Class Noise Mitigation Through Instance Weighting. In *ECML '07 Proceedings of the 18th European conference on Machine Learning*, 708–715.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 614–622.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and Fast But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 254–263.

Zhu, X.; Wu, X.; and Chen, Q. 2003. Eliminating Class Noise in Large Datasets. In *Proceedings of the 20th International Conference on Machine Learning*, 920–927.