BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 7 2013, pages 870–877
doi:10.1093/bioinformatics/btt078

Gene expression

Advance Access publication February 15, 2013

Classification of mislabelled microarrays using robust sparse logistic regression

Jakramate Bootkrajang* and Ata Kabán

School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Previous studies reported that labelling errors are not uncommon in microarray datasets. In such cases, the training set may become misleading, and the ability of classifiers to make reliable inferences from the data is compromised. Yet, few methods are currently available in the bioinformatics literature to deal with this problem. The few existing methods focus on data cleansing alone, without reference to classification, and their performance crucially depends on some tuning parameters.

Results: In this article, we develop a new method to detect mislabelled arrays simultaneously with learning a sparse logistic regression classifier. Our method may be seen as a label-noise robust extension of the well-known and successful Bayesian logistic regression classifier. To account for possible mislabelling, we formulate a label-flipping process as part of the classifier. The regularization parameter is automatically set using Bayesian regularization, which not only saves the computation time that cross-validation would take, but also eliminates any unwanted effects of label noise when setting the regularization parameter. Extensive experiments with both synthetic data and real microarray datasets demonstrate that our approach is able to counter the bad effects of labelling errors in terms of predictive performance, it is effective at identifying marker genes and simultaneously it detects mislabelled arrays to high accuracy.

Availability: The code is available from http://cs.bham.ac.uk/~jxb008.

Contact: J.Bootkrajang@cs.bham.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on December 21, 2012; revised on February 6, 2013; accepted on February 9, 2013

1 INTRODUCTION

High-throughput microarray technologies make it possible to measure the expression levels of thousands of genes. Our ability to use these data to reliably predict the presence of a certain disease and to better understand the biological mechanisms underlying the development of disease is of fundamental importance from the perspective of treatment and prevention. Statistical machine learning methods have already shown a lot of promise towards these goals, and methods that can deal with high dimensional and low sample size settings have been the subject of considerable research efforts over the last decade.

However, the classical machinery of learning a classifier relies on a set of labelled examples, and the quality of a classifier depends crucially on the accurate labelling of these data. Unfortunately, the task of labelling is complex and not without ambiguities. As a result, there is no guarantee that the class labels are all correct; in fact, there is an increasing realization that labelling errors are not uncommon in microarray data—see Malossini *et al*. (2006) and Zhang *et al*. (2009).

The presence of class label noise in training sets has been reported to deteriorate the performance of the existing classifiers in a broad range of classification problems (Krishnan and Nandy, 1990; Lawrence and Schölkopf, 2001; Malossini *et al*., 2006; Yang *et al*., 2012; Yasui *et al*., 2004). Although, the problem posed by the presence of class label noise is acknowledged, often it is naively ignored in practice. Part of the reason may be that symmetric label noise can be relatively harmless—however, asymmetric noise inevitably deteriorates the performance, as it changes the decision boundary between the true classes (Chhikara and McKeon, 1984; Lachenbruch, 1974; Lugosi, 1992).

Various approaches have been devised in the machine learning literature to address the issue of learning from samples with label noise. The seemingly straightforward approach is by means of data preprocessing where any suspect samples are removed or relabelled (Barandela and Gasca, 2000; Brodley and Friedl, 1999; Jiang and Zhou, 2004; Maletic and Marcus, 2000; Muhlenbach *et al*., 2004; Sánchez *et al*., 2003). However, these approaches hold the risk of removing useful data too, which is unsuitable in microarray classification, as the number of training examples is limited.

In sharp contrast with the multitude of methods for microarray classification, there are few attempts to address the problem of label noise in the bioinformatics literature. Malossini *et al*. (2006) pointed out the difference between mislabelled arrays and outliers, and proposed two methods to detect mislabellings based on data perturbation. Zhang *et al*. (2009) developed this work further and obtained improved precision and recall in both synthetic and real data settings. Both of these works are based on data perturbation, and their main focus is to detect suspects that are potentially mislabelled. These methods can help repairing the labels, so we can imagine a two-stage procedure of creating a repaired training set first and feed this to existing classifiers in a second stage. However, one must be aware that any errors made in separate stages of analysis will necessarily accumulate.

In this article, we address the above problems by developing an integrated approach where the ambiguity of the given label assignments is modelled explicitly during the training of a

*To whom correspondence should be addressed.

classifier. This allows us to build on classifiers that have been successful for microarray classification by developing an extension to account for possible label noise. Specifically, here we will harness the sparse Bayesian logistic regression (BLogReg) model proposed by Cawley and Talbot (2006) with a robustness against label noise. From our model formulation, we then derive a new algorithm that alternates between training the classifier and estimating the label noise probabilities. Straightforward calculations further provide the posterior probability of mislabelling for each of the training points. This enables us to detect the suspect samples for possible follow-up study. In addition, our experimental validation results, using both synthetic and real microarray datasets, demonstrate that the proposed method improves on traditional algorithms and achieves a reduced classification error rate. A variant of our approach appears in Bootkrajang and Kabán (2012).

2 METHODS

2.1 A model for label-noise robust logistic regression

We now describe our label-noise robust Logistic Regression (RLogReg) model. We will use the term 'robust' to differentiate this from traditional logistic regression. Consider a set of training data $S = \{(\mathbf{x}_1, \tilde{y}_1), \ldots, (\mathbf{x}_D, \tilde{y}_D)\}$, where $\mathbf{x}_i \in \Re^M$ and $\tilde{y}_i \in \{0, 1\}$, where $\tilde{y}_i$ denotes the observed label of $\mathbf{x}_i$. As in the classical scenario for binary classification, we start with defining the log likelihood:

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{D} \tilde{y}_i \log\left(p(\tilde{y}_i = 1|\mathbf{x}_i, \mathbf{w})\right) + (1 - \tilde{y}_i) \log\left(p(\tilde{y}_i = 1|\mathbf{x}_i, \mathbf{w})\right) \quad (1)$$

where $\mathbf{w}$ is the weight vector orthogonal to the decision boundary and it determines the orientation of the separating hyperplane. If the labels were presumed to be correct, then for a point $\mathbf{x}_i$ we would take

$$p(\tilde{y}_i = 1|\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1 + e^{(-\mathbf{w}^T\mathbf{x}_i)}} \quad (2)$$

and whenever this is above 0.5 we would decide that $\mathbf{x}_i$ belongs to class 1. However, when there is label noise present, making predictions in this way is no longer valid. Instead, we will introduce a latent variable y to represent the true label, and we rewrite $p(\tilde{y}_i = k|\mathbf{x}_i, \mathbf{w})$ as the following:

$$p(\tilde{y}_i = k|\mathbf{x}_i, \mathbf{w}) = \sum_{j=0}^{1} p(\tilde{y}_i = k|y = j)p(y = j|\mathbf{x}_i, \mathbf{w}) \stackrel{def}{=} S_i^k \quad (3)$$

In Equation (3), $p(\tilde{y} = k|y = j) \stackrel{def}{=} \gamma_{jk}$ represents the probability that the label has flipped from the true label j to the observed label k. These parameters form a transition table, which we will call the 'gamma table', Γ, and these label flipping probabilities may be estimated. Using this model, instead of Equation (2) we will have:

$$p(y = 1|\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1 + e^{(-\mathbf{w}^T\mathbf{x}_i)}} \quad (4)$$

We decide that $\mathbf{x}$ belongs to class 1 whenever $p(y = 1|\mathbf{x}, \mathbf{w}) \geq 0.5$.

2.2 Sparsity prior

Microarray data are high dimensional with more features than observations while only a subset of the features is relevant to the target. A vast literature demonstrates that sparsity-inducing regularization approaches are effective in such cases (Cawley and Talbot, 2006; MacKay, 1995; Shevade and Keerthi, 2003). Hence, we now incorporate sparsity in our model described in the previous section. Following Shevade and Keerthi (2003) and Cawley and Talbot (2006), we will use an $L1$ regularization term, which results in the following objective function:

$$\max_{\mathbf{w}} \sum_{i=1}^{D} \log p(\tilde{y}_i|\mathbf{x}_i, \mathbf{w}) - \lambda \parallel \mathbf{w} \parallel_1 \quad (5)$$

where λ is the Lagrange multiplier (or regularization parameter) that balances between fitting the data well and having small parameter values. The $L1$-norm in the regularization term is defined as,

$$\parallel \mathbf{w} \parallel_1 = \sum_{d=1}^{M} |w_d| \quad (6)$$

Now, the regularization parameter λ needs be determined. We cannot use cross-validation, not only for its computational demand, but primarily because it would need a validation set with trusted correct labels, which may be not available. Hence, we adopt the Bayesian regularization approach of Cawley and Talbot (2006), which bypasses the need for cross-validation and determines λ automatically by putting a Jeffrey's prior on λ and integrating it out from the model. This yields the following (see Cawley and Talbot, 2006, for details):

$$\lambda = \frac{N}{\sum_{d=0}^{N} |w_d|} \quad (7)$$

where N denotes the number of non-zero parameters, i.e. those with $w_d \neq 0$ —so $N \leq M$.

2.3 Parameter estimation

It now remains to estimate $\mathbf{w}$ and Γ. Notice that Equation (5) is not differentiable at the origin. Shevade and Keerthi (2003) proposed a simple, yet effective, algorithm to optimize the non-smooth but convex objective function of sparse logistic regression (SLogReg) using the Gauss-Seidel method and using coordinate-wise descent. We will create a modification of this approach to make it applicable to our non-convex objective.

Define $F_d = \frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_d}$, where $w_{d=0}$ is the bias term that is usually left unregularized. The optimality conditions for Equation (5), which are the same as in Shevade and Keerthi (2003) and Cawley and Talbot (2006) can be stated algebraically as the following:

$$\begin{aligned} &F_d = 0 \quad \text{if } d = 0 \\ &F_d = \lambda \quad \text{if } w_d > 0, d > 0 \\ &F_d = -\lambda \quad \text{if } w_d < 0, d > 0 \\ &-\lambda \leq F_d \leq \lambda \quad \text{if } w_d = 0, d > 0 \end{aligned}$$

Accordingly, the violation from optimality of w_d may be summarized as:

$$\begin{aligned} viol_d &= |F_d| && \text{if } d = 0 \\ &= |\lambda - F_d| && \text{if } w_d > 0, d > 0 \\ &= |\lambda + F_d| && \text{if } w_d < 0, d > 0 \\ &= \max(F_d - \lambda, -\lambda - F_d, 0) && \text{if } w_d = 0, d > 0 \end{aligned}$$

We start optimizing the component w_d that makes the largest violation to an optimality condition. At this point, if the objective function was convex then it would be possible to use gradient information to bracket the region where the optimal w_d lies by specifying upper and lower limits (H and L). For example, Shevade and Keerthi (2003) identify 10 different cases for their sparse logistic regression model. However, since our likelihood term is non-convex, the cases identified there are not applicable because the sign of gradients give no information about the interval where the optimal solution resides. Therefore we introduce a simple modification by performing two searches: one in the range $\mathbb{R}^+ \cup \{0\}$ and another in the range $\mathbb{R}^- \cup \{0\}$. We then choose the solution that returns a higher value of the objective function. This modified searching

approach is more general and will work on any locally differentiable function at the expense of a slight increase in computation time. In practice, L and H are finite—provided that the design matrix is standardized and appropriate regularization is imposed on the solution, it is sufficient to search in the (0, 1000) and (−1000, 0) intervals.

Finally, having completed the optimization of $\mathbf{w}$, it remains to derive the update rule for the label-flipping probabilities. Conveniently, these can be estimated via fixed point update equations. By introducing a Lagrange multiplier to ensure that the probabilities in each row of the Γ table sum to 1 and solving the stationary equations, we obtain the following update equations (for details see Bootkrajang and Kabán, 2012):

$$\gamma_{00} = \frac{g_{00}}{g_{00}+g_{01}}, \gamma_{01} = \frac{g_{01}}{g_{00}+g_{01}} \tag{8}$$

$$\gamma_{10} = \frac{g_{10}}{g_{10}+g_{11}}, \gamma_{11} = \frac{g_{11}}{g_{10}+g_{11}} \tag{9}$$

where

$$g_{00} = \gamma_{00}\sum_{i=1}^{D}\left[\frac{(1-\tilde{y}_i)}{S_i^0}(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\right], g_{11} = \gamma_{11}\sum_{i=1}^{D}\left[\frac{\tilde{y}_i}{S_i^1}\sigma(\mathbf{w}^T\mathbf{x}_i)\right]$$
$$g_{01} = \gamma_{01}\sum_{i=1}^{D}\left[\frac{\tilde{y}_i}{S_i^1}(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\right], g_{10} = \gamma_{10}\sum_{i=1}^{D}\left[\frac{(1-\tilde{y}_i)}{S_i^0}\sigma(\mathbf{w}^T\mathbf{x}_i)\right]$$

Derivation details are given in the Supplementary Material.

The optimization of the log-likelihood is then to alternate between optimizing $\mathbf{w}$ along with updating λ according to Equation (7) until convergence is reached, and we alternate this with the fixed point update equations of the label-flipping probabilities. The entire optimization procedure is summarized in Algorithms 1–2.

Algorithm 1 Main loop

Input: Training examples.
Initialize $\mathbf{w} \leftarrow 0$, $\lambda \leftarrow 0$, $I_{nz} \leftarrow \{w_0\}$, $I_z \leftarrow \{w_{d,d\in\{1,n\}}\}$, Γ.
while Optimality violator exists in I_z **do**
  Find the greatest optimality violator, v, in I_z
  repeat
    Optimize w_v using Algorithm 2
    $I_z \leftarrow I_z \backslash \{w_v\}$
    $I_{nz} \leftarrow I_{nz} \cup \{w_v\}$
    Find the maximum optimality violator, v, in I_{nz}
  until No violator exists in I_{nz}
  Update the entries of Γ by Equations (8) and (9)
  Update regularization parameter, λ by Equation (7)
end while
Output: Optimized weight vector, $\mathbf{w}$. Optimized Γ.

Algorithm 2 Optimization of $\mathbf{w}$

Input: Violating component, w_v, I_z, I_{nz}
$w_v \leftarrow 0$
if w_v satisfies optimality conditions **then**
  $I_z \leftarrow I_z \cup \{w_v\}$
  $I_{nz} \leftarrow I_{nz} \backslash \{w_v\}$
  break
else
  Restore previous value of w_v
  $t_1 \leftarrow$ Optimize Equation (1) w.r.t w_v in the range $(-lim, 0)$ range
  $t_2 \leftarrow$ Optimize Equation (1) w.r.t w_v in the range $(0, lim)$ range
end if
$w_v \leftarrow t_i$ that maximize Equation (1), where $i \in \{1, 2\}$.
Output: Optimized w_v

2.4 Detecting mislabelled points

For an observation $(\mathbf{x}_i, \tilde{y}_i)$, the probability of it being mislabelled can be computed as the following:

$$p(y \neq \tilde{y}_i|\mathbf{x}_i) = \sum_{j=0, j\neq\tilde{y}_i}^{1} p(y=j|\mathbf{x}_i) \tag{10}$$

This may be thought of as the models 'degree of belief' that $\mathbf{x}_i$'s label is incorrect. We may use it either in this form, or in a hard-thresholded form (i.e. predict that the point $\mathbf{x}_i$ is mislabelled if $p(y \neq \tilde{y}_i|\mathbf{x}_i) \geq 0.5$).

2.5 A note on low sample size, high dimensional data

Since additional parameters Γ are being estimated from the data, we expect that RLogReg will require more training examples to deliver its full potential. In microarray datasets, the training set size is often of the order of tens only. A possible workaround in such cases is to guide the algorithm by presetting the gamma table from domain knowledge about the likely proportion of mislabelled data. When such knowledge exists, the values of gamma may either be fixed throughout the optimization process or they may be seeded initially and then optimized.

3 RESULTS

3.1 Experiment setting

We will compare the classification performance of RLogReg, RLogReg with fixed gamma table (denoted RLogReg-F) and its traditional counterpart, i.e. BLogReg of Cawley and Talbot (2006). The reader is referred to Cawley and Talbot (2006) for a comparison between BLogReg against the Relevance Vector Machine (RVM) and SLogReg (Shevade and Keerthi, 2003) where BLogReg was shown to be superior. We shall demonstrate that our proposed robust extension of BLogReg performs better than the original BLogReg in terms of classification performance when there is label noise present in the training set. Moreover, our model can be used to identify mislabelled arrays for potential follow-on study.

Before proceeding, we should comment that symmetric and asymmetric label flipping have very different consequences in classification. Symmetric or uniform flipping means that each class is affected by label flipping in the same proportion. In contrast, asymmetric or non-uniform flipping is when the label flips from one class to another more often than vice-versa. The latter type of label flipping has been theoretically shown (Lugosi, 1992) to degrade the performance of an algorithm to a much larger degree, as it modifies the decision boundary between the true classes. Our empirical study (Bootkrajang and Kabán, 2012) also demonstrated this. Therefore, we will mainly focus our attention on datasets with asymmetric label noise and indeed expect the advantages of our approach to be most apparent in that setting.

To demonstrate the benefit of having a label noise model embedded in the classifier, we start with experiments on synthetic data where labels were asymmetrically flipped at the rate of 30%. The use of synthetic data for controlled experiments is standard in bioinformatics (see e.g. Zhang *et al.*, 2009), as it allows us assess the performance of a new approach against a ground truth. We shall then move on to analysing real microarray

Table 1. Characteristics of the datasets used in the reported experiments

Dataset	No. of samples		No. of genes	No. of wrong labels	
	Class 1	Class 2		Class 1	Class 2
Synth-500	250	250	100–1000	0	75
Synth-100	50	50	100–1000	0	15
Colon	40 (T)	22 (N)	2000	5	4
Breast	25 (ER+)	24 (ER−)	7129	4	5

datasets where label noises have not been injected artificially. These datasets have been previously reported to contain wrongly labelled samples. Finally, we shall assess the ability of our proposed approach to identifying mislabelled arrays using Receiver Operating Characteristics (ROC) analysis.

3.2 Datasets

We generate synthetic data by sampling points from a standard Gaussian distribution where the class label associated with each point is assigned by a logistic function with a predefined weight vector $\mathbf{w}$ having only three relevant features, $w_1 = w_2 = w_3 = 10/3, w_i = 0, \forall i > 3$, following Ng (2004). We create sets with 500 training points and sets with 100 training points together with independent test sets of 100 points each time, and call these datasets *Synth-500* and *Synth-100,* respectively. The dimensionality of the synthetic datasets ranges from 100 up to 1000. Asymmetric label noise was artificially injected into each synthetic dataset at the 30% rate.

Further, we use two real microarray datasets: Colon cancer (Alon *et al*., 1999) and Breast cancer (West *et al*., 2001)—both of which are known to contain some mislabelled arrays. No artificial label flipping is injected in these data. We standardize these datasets so the rows of the $D \times M$ design matrix (where D is the number of observations and M is the dimensionality) of the input sample will have zero mean and unit variance. Table 1 summarizes the characteristics of all of these datasets used. Additional datasets and results are given in the Supplementary Material.

3.2.1 Error measures While in the case of synthetic data the true labels can be used to validate the predictive accuracy of our algorithm, in the real microarray data there is no absolute ground truth. Since the labels given in the datasets may be incorrect, the issue of what should count as a misclassification must be defined. We define two variants for measuring out-of-sample error rates:

- Corrected (CRT): Count misclassification errors against the 'corrected' labels where corrections are made cf. the mislabellings reported in the literature.
- Cleansed (CLN): Exclude any mislabelled suspects (known in the literature) from the test sets for the purpose of evaluation, so these are always placed into the training set instead; then count the misclassification errors on test sets in the usual way.

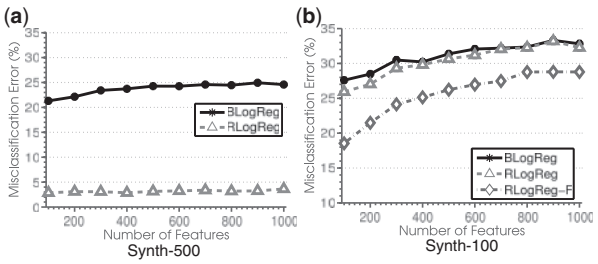

Fig. 1. Misclassification on test set, as obtained by RLogReg and BLogReg, respectively, on synthetic datasets with 30% asymmetric label noise. Left: Training sets of size 500; Right: Training sets of size 100. RLogReg-F denotes the version of RLogReg with the gamma matrix pre-set to its correct value

3.3 Results and analysis

3.3.1 Results on synthetic data The average misclassification error rates on the Synth-500 and Synth-100 datasets are shown in Figure 1 as the data dimension is varied. Each point on these plots represents the average misclassification rate on the test sets, where the average is taken over 500 independent repetitions of the experiment. The error bars are too small to be visible. We see that RLogReg achieves significantly lower error rates than BLogReg on the datasets that contain more training examples (Synth-500). This clearly demonstrates the advantage of modelling the label noise process. On the smaller size dataset (Synth-100), however, the performance gain becomes marginal—this is because the accurate estimation of the additional parameters (label-flipping probabilities) requires sufficient training data for our approach to achieve its full potential. Nevertheless, it is should be noticed that even in the small sample setting, RLogreg performs no worse than BLogReg on all the datasets tested (additional results are given in the Supplementary Material.). More importantly, the rightmost plot shows that we can counter the problem of small sample sizes by using prior knowledge about the extent of label noise, e.g. by pre-defining the gamma table. We denote this version as RLogReg-F in the figure, and we see this significantly improves the classification accuracy in the small sample setting.

Beyond classification performance, it is of interest to evaluate the methods' ability to identify the relevant predictive genes. Figure 2 shows the estimated weight vectors as obtained by BLogReg and RLogReg respectively from 100-dimensional synthetic data with only the first three features being relevant. The classifiers were trained on 250 training examples per class that were subjected to 30% asymmetric label flipping. We see that RLogReg achieved a more accurate estimation of the weight vector, while BLogReg became confused by the noisy labels and selected too many false non-zero weights. This is an important advantage of RLogReg over BLogReg when it comes to finding a small set of predictive marker genes.

3.3.2 Results on colon cancer The colon cancer classification task aims to distinguish between normal tissue and tumour. According to Alon *et al*. (1999), there is biological evidence that the samples T2, T30, T33, T36, T37, N8, N12, N34, N36 may be mislabelled. The proportion of mislabelling in the two classes is unequal; hence, this is a case of asymmetric label

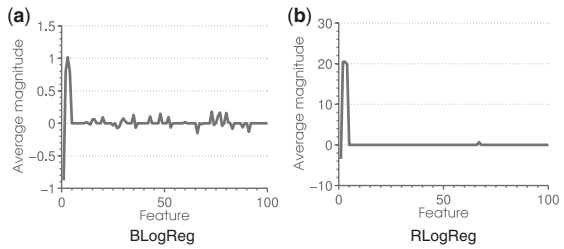

Fig. 2. Comparison of the magnitude of weights for the 100 features as obtained in one run of BLogReg and RLogReg, respectively, on synthetic data that contains only three relevant features (250 training examples in each class, 30% asymmetric label noise). We see that BLogReg selects too many features whereas RLogReg has a better ability to turn off the irrelevant ones

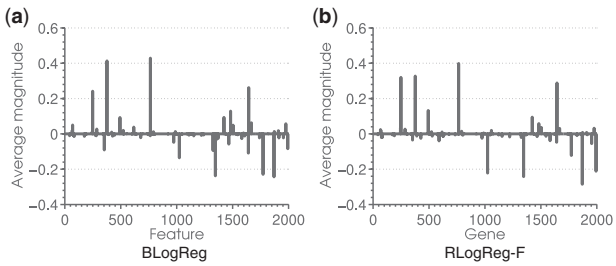

Fig. 3. Comparison of the average weights of features selected by BLogReg and RLogReg-F on the Colon cancer dataset over 1000 bootstrap repeats (50 train/12 test)

Table 2. LOO misclassification (%) on Colon Cancer dataset

Algorithm	LOO-CRT	LOO-CLN	No. of genes
BLogReg	8.06 ± 0.44	7.55 ± 0.64	11.94 ± 0.41
RLogReg	9.68 ± 0.48	9.43 ± 0.66	11.85 ± 0.41
RLogReg-F	4.83 ± 0.35	1.88 ± 0.54	9.21 ± 0.45

The average number of selected genes (±standard deviation) was computed from the CLN runs.

flipping that can distort the correct decision boundary of the classes. The limited number of training observations implies that a good estimate of the gamma table may be difficult to obtain from the data alone (as we have seen in the previous section), nevertheless prior knowledge of the noise proportions may still allow us to exploit the advantages of having a noise model as integral part of our classifier. Therefore, we include RLogReg-F in our experiments, with the gamma table set to the true label-flipping proportions. Table 2 reports the leave-one-out (LOO) errors in terms of the error measures defined in Section 3.2.1, and we also give the average number of genes selected by the three methods considered.

The results confirm the expectations. RLogReg that attempts to estimate the gamma table along with all other parameters is marginally worse than BLogReg (although not statistically significantly so, according to the unpaired t-test), while RLogReg-F improves over BLogReg in all validation criteria used, and it also selects a smaller fraction of relevant features.

Figure 3 shows the average magnitude of each gene according to BLogReg and RLogReg-F, respectively. These are averages of $\mathbf{w}$ estimates across 1000 bootstrap repetitions to inspect possible systematic differences. These average weights turned out to be quite similar for BLogReg and RLogReg-F, with the exception of a few genes that had been ranked differently by the two methods. To see this, a summary of top ten selected genes and their estimated weights are given in Tables 3 and 4.

3.3.3 Results on breast cancer We further apply the proposed model on the Breast Cancer dataset from West *et al.* (2001). The aim is to discriminate between oestrogen-positive and oestrogen-negative observations. According to West *et al.* (2001), there is biological evidence that the arrays 11, 14, 16, 31, 33, 40, 43, 45, 46 are mislabelled. However, unlike the Colon dataset, we observe the nature of label flipping in the Breast cancer dataset is rather close to symmetric. As a consequence, mislabelling might do less harm to traditional classifiers in terms of class prediction on future arrays. Table 5 summarizes LOO error rates together with the numbers of genes selected by the classifiers. The picture is quite similar to what we have seen in the case of Colon, although the differences tend to be smaller, as the label noise here is more symmetric.

We also see that RLogReg did pretty well with a limited amount of training data, but of course the difficulty of accurate estimation of the gamma table from such few points remains an issue. In fact, the estimated gamma table of RLogReg may converge to identity in such conditions, which statistically will result in a weight vector that is identical to that of BLogReg. As previously, knowledge of the extent of noise can be used here, resulting in a slight improvement for RLogReg-F. Finally, as somewhat expected, the average magnitude of gene weights from BLogreg and RLogreg-F look similar, as shown in Figure 4, which was expected by the symmetric nature of the label noise in this dataset.

3.4 Computation time

We should give an indication of the added computation overhead required by our noise modelling relative to the existing BLogReg. One LOO loop on all datasets considered took on average 4 s for RLogReg, while BLogReg required roughly 0.2 s on an Intel's Core-i5 3.2 GHz machine. We believe this extra computation time is most worthwhile especially when the training set size is sufficiently large to exploit the full potential of the presented approach.

3.5 Detecting mislabelled instances

One of the most appealing features of our proposed algorithm is the possibility to detect mislabelled examples from the data, in addition to classification and gene selection.

There are two types of possible errors: (i) a false positive is when a sample is believed to be mislabelled despite it is in fact labelled correctly; and (ii) a false negative is when a sample is believed to be labelled correctly despite its label is in fact incorrect. A good way to summarize both, while also making use of the probabilistic outputs given by the sigmoid function, is by

Table 3. Relative importance of top 10 genes selected by the BLogReg algorithm

Gene number	Gene annotation	Average magnitude
765	Human cysteine-rich protein (CRP) gene, exons 5 and 6	0.4289
377	H.sapiens mRNA for GCAP-II/uroguanylin precursor	0.4132
1644	C4-DICARBOXYLATE TRANSPORT SENSOR PROTEIN DCTB (*Rhizobium leguminosarum*)	0.2618
1870	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR (HUMAN)	−0.2435
249	Human desmin gene, complete cds	0.2416
1346	60S RIBOSOMAL PROTEIN L24 (*Arabidopsis thaliana*)	−0.2376
1772	COLLAGEN ALPHA 2(XI) CHAIN (*Homo sapiens*)	−0.2292
1024	ATP SYNTHASE A CHAIN (*Trypanosoma brucei*)	−0.1356
1482	Human spermidine synthase gene, complete cds	0.1290
1641	Human enkephalin B (enkB) gene, exon 4 and 3′ flank and complete cds	−0.1090

Table 4. Relative importance of the top 10 genes selected by RLogReg-F algorithm

Gene number	Gene annotation	Average magnitude
765	Human cysteine-rich protein (CRP) gene, exons 5 and 6	0.3988
377	H.sapiens mRNA for GCAP-II/uroguanylin precursor	0.3273
249	Human desmin gene, complete cds.	0.3201
1644	C4-DICARBOXYLATE TRANSPORT SENSOR PROTEIN DCTB (*R.leguminosarum*)	0.2883
1870	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR (HUMAN)	−0.2852
1346	60S RIBOSOMAL PROTEIN L24 (*A.thaliana*)	−0.2420
1024	ATP SYNTHASE A CHAIN (*T.brucei*)	−0.2220
1993	Human hormone-sensitive lipase (LIPE) gene, complete cds	−0.2114
493	MYOSIN HEAVY CHAIN, NONMUSCLE (*Gallus gallus*)	0.1325
1772	COLLAGEN ALPHA 2(XI) CHAIN (*H.sapiens*)	−0.1224

Table 5. LOO misclassification (%) on Breast Cancer dataset

Algorithm	LOO-CRT	LOO-CLN	No. of genes
BLogReg	18.37 ± 0.79	2.50 ± 0.40	9.22 ± 0.58
RLogReg	18.37 ± 0.79	2.50 ± 0.40	9.10 ± 0.63
RLogReg-F	16.33 ± 0.76	0.00 ± 0.29	7.58 ± 0.50

The average number of selected genes (±standard deviation) was computed from the CLN runs.

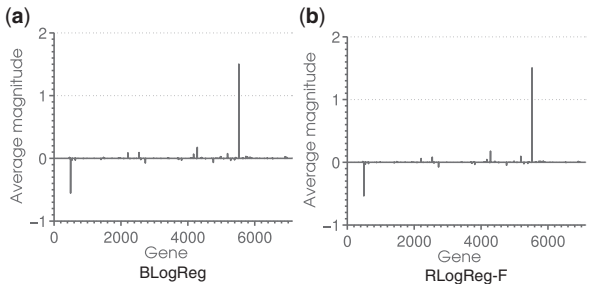

Fig. 4. Comparison of the average weights of features selected by BLogReg and RLogReg-F on the Breast Cancer dataset over 1000 bootstrap repeats (39 train/10 test)

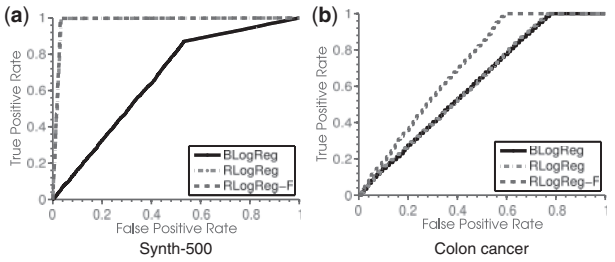

Fig. 5. Average ROC curves for BLogReg, RLogReg and RLogReg-F on *Synth-500* and colon cancer benchmarks. For consistency with classification result bootstrap is performed on *Synth-500* while LOO is used to obtain the result for colon cancer. The prediction is based on hard-thresholded rule

constructing the ROC curves. The area under the ROC curve signifies the probability that a randomly drawn and mislabelled example would be flagged by the proposed algorithms. Figure 5 shows the ROC curves for *Synth-500* and Colon cancer datasets. Superimposed for reference, we also plotted the ROC curves that correspond to BLogReg. BLogReg considers that all points have the correct labels, and it has not been designed to spot mislabelled points. The best we can do is to take that mistakes made on the training points are mislabelling predictions. From

Table 6. Identifying mislabelled samples in colon cancer dataset

Source	Suspects identified									Extra samples identified
Alon *et al*. (1999)	T2	T30	T33	T36	T37	N8	N12	N34	N36	
Furey *et al*. (2000)	—	◦	◦	◦	—	◦	—	◦	◦	
Li *et al*. (2001)	—	◦	◦	◦	—	—	—	◦	◦	
Kadota *et al*. (2003)	◦	—	—	—	◦	◦	—	◦	◦	T6, N2
Malossini *et al*. (2006) (RAPIV)	—	◦	◦	◦	◦	◦	—	◦	◦	N28, N29, N40
Malossini *et al*. (2006) (PRAPIV)	—	◦	◦	◦	◦	◦	—	◦	◦	N2, N28
BLogReg	◦	◦	◦	◦	—	◦	◦	◦	◦	T3, T32, N35, N40
BLogReg (%)	1	9	14	63	0	9	18	32	37	
RLogReg-F	◦	◦	◦	◦	—	◦	◦	◦	◦	N2, T32, N40
RLogReg-F (%)	4	22	55	79	0	15	15	37	68	

The detections for RLogReg-F are based on the hard threshold rule ($p(\tilde{y} \neq y|\mathbf{x}, \mathbf{w}) \geq 0.5$). The first line is the 'gold standard' that is backed up by biological evidence in the literature.

Figure 5, we see the gap between the two curves is significant and well apparent in the experiment on Synth-500. This quantifies the gain that our modelling approach is able to obtain. The gain for Colon is smaller but still significant, despite the dataset size is so limited, provided that RLogReg incorporates knowledge about the proportion of mislabelling (i.e. RLogReg-F).

3.6 Comparison with previous findings

In addition to comparisons that quantify the benefits of having a noise model, we compare our results with previously identified mislabelling in the Colon cancer samples. We conduct 100 bootstrap repetitions drawing subsets of size 50 from the total of 62 points randomly while imposing that none of the suspects from the literature are left out. In Table 6, after quoting the previous detections from the literature, we report the mislabelling detections obtained by BLogReg-F and BLogReg respectively, in two forms: (i) from the run that returned the largest number of detections, and (ii) the percentage that a particular array was flagged up as a mislabelling during the 100 repetitions.

It is interesting to note that RLogReg-F was able to identify up to seven mislabelled points, and these also agree with the majority of previously reported detections using other algorithms (i.e. for T30, T33, T36, N34 and N36). BLogReg is also able to find up to seven mislabelled samples but with fewer true positives and more false positives.

From both figures, we see that RLogReg-F is able to identify mislabelled arrays more often than BLogReg can.

4 CONCLUSIONS

We proposed a robust extension of sparse Bayesian logistic regression for classification in the presence of labelling errors. The numerical experiments suggest that our approach is superior to its traditional counterpart when the training data contains labelling errors, and more significantly so when the label-flipping distribution is asymmetric. Simultaneously, our methods are effective in identifying marker genes and detecting mislabelled data. Since our robust model needs to estimate the label-flipping probabilities together with the parameters of the classifier, it does require more training data to achieve its full potential. However, in our experience, RLogReg performs statistically no worse than BLogReg even when the training set sizes are small. The need for more data can also be relaxed by incorporating knowledge about the extent of label noise.

Funding: J.B. is supported by the Royal Thai Government. A.K. acknowledges the MRC Discipline Hopping Award G0701858 (ID no. 85545).

Conflict of Interest: none declared.

REFERENCES

Alon,U. *et al.* (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.

Barandela,R. and Gasca,E. (2000) Decontamination of training samples for supervised pattern recognition methods. In: *Advances in Pattern Recognition, Lecture Notes in Computer Science*, Vol. 1876. Springer, Berlin Heidelberg, pp. 621–630.

Bootkrajang,J. and Kabán,A. (2012) Label-noise robust logistic regression and its applications. In: *Proceeding of Machine Learning and Knowledge Discovery in Databases - European Conference, ECML-PKDD* 2012, Part I, pp. 143–158.

Brodley,C.E. and Friedl,M.A. (1999) Identifying mislabeled training data. *J. Artif. Intell. Res.*, **11**, 131–167.

Cawley,G.C. and Talbot,N.L. (2006) Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics*, **22**, 2348–2355.

Chhikara,R.S. and McKeon,J. (1984) Linear discriminant analysis with misallocation in training samples. *J. Am. Stat. Assoc.*, **79**, 899–906.

Furey,T.S. *et al.* (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Jiang,Y. and Zhou,Z.H. (2004) Editing training data for k-NN classifiers with neural network ensemble. In: *Advances in Neural Networks, Lecture Notes in Computer Science*, Vol. 3173. Springer, pp. 356–361.

Kadota,K. *et al.* (2003) Detecting outlying samples in microarray data: a critical assessment of the effect of outliers on sample classification. *Chem. Bio. Inform. J.*, **3**, 30–45.

Krishnan,T. and Nandy,S.C. (1990) Efficiency of discriminant analysis when initial samples are classified stochastically. *Pattern Recognit.*, **23**, 529–537.

Lachenbruch,P.A. (1974) Discriminant analysis when the initial samples are misclassified II: non-random misclassification models. *Technometrics*, **16**, 419–424.

Lawrence,N.D. and Schölkopf,B. (2001) Estimating a kernel fisher discriminant in the presence of label noise. In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 306–313.

Li,L. *et al*. (2001) Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Comb. Chem. High Throughput Screen.*, **4**, 727–739.

Lugosi,G. (1992) Learning with an unreliable teacher. *Pattern Recognit*, **25**, 79–87.

MacKay,D.J. (1995) Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network*, **6**, 469–505.

Maletic,J.I. and Marcus,A. (2000) Data cleansing: beyond integrity analysis. In: *Proceedings of the Conference on Information Quality*, pp. 200–209.

Malossini,A. *et al*. (2006) Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, **22**, 2114–2121.

Muhlenbach,F. *et al*. (2004) Identifying and handling mislabelled instances. *J. Intell. Inf. Syst.*, **22**, 89–109.

Ng,A.Y. (2004) Feature selection, L1 vs. L2 regularization, and rotational invariance. In: *Proceedings of the 21st International Conference on Machine Learning, ICML '2004*, pp. 78–85.

Sánchez,J.S. *et al*. (2003) Analysis of new techniques to obtain quality training sets. *Pattern Recognit. Lett.*, **24**, 1015–1022.

Shevade,S.K. and Keerthi,S.S. (2003) A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, **19**, 2246–2253.

West,M. *et al*. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl Acad. Sci. USA*, **98**, 11462–11467.

Yang,T. *et al*. (2012) Multiple kernel learning from noisy labels by stochastic programming. In: *Proceedings of the 29th International Conference on Machine Learning, ICML '1202*, pp. 233–240.

Yasui,Y. *et al*. (2004) Partially supervised learning using an EM-boosting algorithm. *Biometrics*, **60**, 199–206.

Zhang,C. *et al*. (2009) Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model. *Bioinformatics*, **25**, 2708–2714.