Gene expression

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 labelling, 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
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 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 $S = \{ (x_1, \tilde{y}_1), \ldots, (x_d, \tilde{y}_d) \}$, where $x_i \in \mathbb{R}^N$ and $\tilde{y}_i \in \{0, 1\}$, where $\tilde{y}_i$ denotes the observed label of $x_i$. As in the classical scenario for binary classification, we start with defining the log likelihood:

$$ L(w) = \sum_{i=1}^{D} \tilde{y}_i \log(p(\tilde{y}_i = 1|x_i, w)) + (1 - \tilde{y}_i) \log(p(\tilde{y}_i = 1|x_i, w)) $$

(1)

where $w$ 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 $x_i$ we would take

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

(2)

and whenever this is above 0.5 we would decide that $x_i$ 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 $y_i$ to represent the true label, and we rewrite $p(\tilde{y}_i = 1|x_i, w)$ as the following:

$$ p(\tilde{y}_i = 1|x_i, w) = \sum_{j=0}^{1} p(\tilde{y}_i = j|y) = \sigma(y = j|x_i, w) $$

(3)

In Equation (3), $p(\tilde{y}_i = j|y) = p_{yj}$ 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’, $\Gamma$, and these label flipping probabilities may be estimated. Using this model, instead of Equation (2) we will have:

$$ p(y = 1|x_i, w) = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}} $$

(4)

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

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:

$$ \max_w \sum_{i=1}^{D} \log(p(y_i|x_i, w)) - \lambda \| w \|_1 $$

(5)

where $\lambda$ 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,

$$ \| w \|_1 = \sum_{j=1}^{M} |w_j| $$

(6)

Now, the regularization parameter $\lambda$ 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 $\lambda$ automatically by putting a Jeffreys’s prior on $\lambda$ and integrating it out from the model. This yields the following (see Cawley and Talbot, 2006, for details):

$$ \lambda = \frac{N}{\sum_{j=1}^{N} |w_j|} $$

(7)

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

2.3 Parameter estimation

It now remains to estimate $w$ and $\Gamma$. 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 $F_d = \frac{\sigma(0)}{\sigma(5)}$, where $w_{d,\text{bias}}$ 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:

$$ F_d = 0 \quad \text{if} \quad d = 0 $$

$$ F_d = \lambda \quad \text{if} \quad w_d > 0, d > 0 $$

$$ F_d = -\lambda \quad \text{if} \quad w_d < 0, d > 0 $$

$$ -\lambda \leq F_d \leq \lambda \quad \text{if} \quad w_d = 0, d > 0 $$

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

$$ \text{viol}_d = |F_d| \quad \text{if} \quad d = 0 $$

$$ = |\lambda - F_d| \quad \text{if} \quad w_d > 0, d > 0 $$

$$ = |\lambda + F_d| \quad \text{if} \quad w_d < 0, d > 0 $$

$$ = \max(F_d - \lambda, -\lambda - F_d) \quad \text{if} \quad w_d = 0, d > 0 $$

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 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 $w$, 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 $\Gamma$ table sum to 1 and solving the stationary equations, we obtain the following update equations (for details see Bootkrajang and Kabań, 2012):

$$y_{00} = \frac{g_{00}}{g_{00} + g_{01}}, y_{01} = \frac{g_{01}}{g_{00} + g_{01}}$$

(8)

$$y_{10} = \frac{g_{10}}{g_{10} + g_{11}}, y_{11} = \frac{g_{11}}{g_{10} + g_{11}}$$

(9)

where

$$g_{00} = \sum_{i=1}^{D} \left( \frac{(1 - \tilde{y}_i)}{S_i} (1 - \sigma(w^T x_i)) \right), g_{01} = \sum_{i=1}^{D} \left( \frac{\tilde{y}_i}{S_i} \sigma(w^T x_i) \right)$$

$$g_{10} = \sum_{i=1}^{D} \left( \frac{\tilde{y}_i}{S_i} (1 - \sigma(w^T x_i)) \right), g_{11} = \sum_{i=1}^{D} \left( \frac{(1 - \tilde{y}_i)}{S_i} \sigma(w^T x_i) \right)$$

Derivation details are given in the Supplementary Material.

The optimization of the log-likelihood is then to alternate between optimizing $w$ along with updating $\lambda$ 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 $w \leftarrow 0$, $\lambda \leftarrow 0$, $I_{0} \leftarrow \{w_0\}$, $I_{1} \leftarrow \{w_{d, d[1, n]}\}$, $\Gamma$.

while Optimality violator exists in $I_{0}$ do

repeat

Optimize $w_{0}$ using Algorithm 2

$I_{2} \leftarrow I_{0} \setminus \{w_{0}\}$

$I_{a} \leftarrow I_{0} \cup \{w_{0}\}$

Find the maximum optimality violator, $v$, in $I_{a}$

until No violator exists in $I_{a}$

Update the entries of $\Gamma$ by Equations (8) and (9)

Update regularization parameter, $\lambda$ by Equation (7)

end while

**Output:** Optimized weight vector, $w$. Optimized $\Gamma$.

### Algorithm 2 Optimization of $w$

**Input:** Violating component, $w_{v}, I_{a}, I_{0}$

$w_{v} \leftarrow 0$

if $w_{v}$ satisfies optimality conditions then

$I_{a} \leftarrow I_{a} \cup \{w_{v}\}$

$I_{a} \leftarrow I_{a} \setminus \{w_{v}\}$

break

else

Restore previous value of $w_{v}$

$i_{1} \leftarrow$ Optimize Equation (1) w.r.t $w_{1}$ in the range $(-\infty, 0)$ range

$i_{2} \leftarrow$ Optimize Equation (1) w.r.t $w_{2}$ in the range $(0, \infty)$ range

end if

$w_{v} \leftarrow i_{1}$ that maximize Equation (1), where $i \in \{1, 2\}$.

**Output:** Optimized $w_{v}$.

### 2.4 Detecting mislabelled points

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

$$p(y \neq \tilde{y}|x) = \frac{1}{\sum_{j=0}^{n} p(y = j|x)}$$

(10)

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

### 2.5 A note on low sample size, high dimensional data

Since additional parameters $\Gamma$ 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 Kabań, 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 to assess the performance of a new approach against a ground truth. We shall then move on to analysing real microarray
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 $w$ having only three relevant features, $w_1 = w_2 = w_3 = 10/3, w_i = 0, i > 3$, 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 mislabelings 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.

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of samples</th>
<th>No. of genes</th>
<th>No. of wrong labels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class 1</td>
<td>Class 2</td>
<td>Class 1</td>
</tr>
<tr>
<td>Synth-500</td>
<td>250</td>
<td>250</td>
<td>100–1000</td>
</tr>
<tr>
<td>Synth-100</td>
<td>50</td>
<td>50</td>
<td>100–1000</td>
</tr>
<tr>
<td>Colon</td>
<td>40 (T)</td>
<td>22 (N)</td>
<td>2000</td>
</tr>
<tr>
<td>Breast</td>
<td>25 (ER+)</td>
<td>24 (ER–)</td>
<td>7129</td>
</tr>
</tbody>
</table>

![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.](image)

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 modeling 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 mislabeling in the two classes is unequal; hence, this is a case of asymmetric label
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

![Figure 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.](https://example.com/fig2.png)

![Figure 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).](https://example.com/fig3.png)

---

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LOO-CRT</th>
<th>LOO-CLN</th>
<th>No. of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLogReg</td>
<td>8.06 ± 0.44</td>
<td>7.55 ± 0.64</td>
<td>11.94 ± 0.41</td>
</tr>
<tr>
<td>RLogReg</td>
<td>9.68 ± 0.48</td>
<td>9.43 ± 0.66</td>
<td>11.85 ± 0.41</td>
</tr>
<tr>
<td>RLogReg-F</td>
<td>4.83 ± 0.35</td>
<td>1.88 ± 0.54</td>
<td>9.21 ± 0.45</td>
</tr>
</tbody>
</table>

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

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

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

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

<table>
<thead>
<tr>
<th>Gene number</th>
<th>Gene annotation</th>
<th>Average magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>765</td>
<td>Human cysteine-rich protein (CRP) gene, exons 5 and 6</td>
<td>0.3988</td>
</tr>
<tr>
<td>377</td>
<td>H. sapiens mRNA for GCAP-II/uroguanylin precursor</td>
<td>0.3273</td>
</tr>
<tr>
<td>249</td>
<td>Human desmin gene, complete cds</td>
<td>0.3201</td>
</tr>
<tr>
<td>1644</td>
<td>C4-DICARBOXYLATE TRANSPORT SENSOR PROTEIN DCTB (<em>Rhizobium leguminosarum</em>)</td>
<td>0.2838</td>
</tr>
<tr>
<td>1870</td>
<td>PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR (HUMAN)</td>
<td>-0.2852</td>
</tr>
<tr>
<td>1346</td>
<td>60S RIBOSOMAL PROTEIN L24 (<em>Arabidopsis thaliana</em>)</td>
<td>-0.2420</td>
</tr>
<tr>
<td>1024</td>
<td>ATP SYNTHASE A CHAIN (<em>Trypanosoma brucei</em>)</td>
<td>-0.2220</td>
</tr>
<tr>
<td>1993</td>
<td>Human hormone-sensitive lipase (LIPE) gene, complete cds</td>
<td>-0.2114</td>
</tr>
<tr>
<td>493</td>
<td>MYOSIN HEAVY CHAIN, NONMUSCLE (<em>Gallus gallus</em>)</td>
<td>0.1325</td>
</tr>
<tr>
<td>1772</td>
<td>COLLAGEN ALPHA 2(XI) CHAIN (<em>Homo sapiens</em>)</td>
<td>-0.1224</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LOO-CRT</th>
<th>LOO-CLN</th>
<th>No. of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLogReg</td>
<td>18.37 ± 0.79</td>
<td>2.50 ± 0.40</td>
<td>9.22 ± 0.58</td>
</tr>
<tr>
<td>RLogReg</td>
<td>18.37 ± 0.79</td>
<td>2.50 ± 0.40</td>
<td>9.10 ± 0.63</td>
</tr>
<tr>
<td>RLogReg-F</td>
<td>16.33 ± 0.76</td>
<td>0.00 ± 0.29</td>
<td>7.58 ± 0.50</td>
</tr>
</tbody>
</table>

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

construc
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


Lachenbruch, P.A. (1975) Discriminant analysis when the initial samples are misclassified II: non-random misclassification models. Technometrics, 16, 419-424.


