# Label-Noise Robust Logistic Regression and Its Applications

Jakramate Bootkrajang and Ata Kabán

School of Computer Science, The University of Birmingham,
Birmingham, B15 2TT, UK
{J.Bootkrajang,A.Kaban}@cs.bham.ac.uk

**Abstract.** The classical problem of learning a classifier relies on a set of labelled examples, without ever questioning the correctness of the provided label assignments. However, there is an increasing realisation that labelling errors are not uncommon in real situations. In this paper we consider a label-noise robust version of the logistic regression and multinomial logistic regression classifiers and develop the following contributions: (i) We derive efficient multiplicative updates to estimate the label flipping probabilities, and we give a proof of convergence for our algorithm. (ii) We develop a novel sparsity-promoting regularisation approach which allows us to tackle challenging high dimensional noisy settings. (iii) Finally, we throughly evaluate the performance of our approach in synthetic experiments and we demonstrate several real applications including gene expression analysis, class topology discovery and learning from crowdsourcing data.

## 1 Introduction

In the context of supervised learning, a classification rule is to be derived from a set of labelled examples. Regardless of the learning approach used, the induction of the classification rule crucially relies on the given class labels. Unfortunately, there is no guarantee that the class labels are all correct. The presence of class label noise inherent in training samples has been reported to deteriorate the performance of the existing classifiers in a broad range of classification problems [12,25,21]. Remarkably, examples of mislabelling have been reported even in biomedical sciences where the number of instances is only of the order of tens [1,15,26]. There is an increasing research literature that aims to address the issues related to learning from samples with noisy class label assignments. The seemingly straightforward approach is by means of data preprocessing where any suspect samples are removed or relabelled [3,2,9]. However, these approaches hold the risk of removing useful data too, especially when the number of training examples is limited.

In this paper, we take a model based approach and consider a label-noise robust logistic regression and multinomial logistic regression. There are already several works employing latent variable models of this kind, especially in the field of epidemiology, econometrics and computer-aided diagnosis (see [20,7,23]

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 143–158, 2012.

and references therein), and more recently for learning from crowds [23]. Our approach develops these ideas further while it differs in certain respects. [20] studied label-noise robust logistic regression with known label flipping probabilities but they reckon problems when these probabilities are unknown. In turn, we try to learn the classifier jointly with estimating the label flipping probabilities. The robust model discussed in [7] is also structurally similar to ours although they provided no algorithmic solution to learning the model. In contrast, one of our novel contributions in this paper is to develop an efficient learning algorithm together with a proof of its convergence. The recent work in [23] focuses on learning from multiple noisy labels and demonstrate that multiple sets of noisy labels increases performance. In contrast, our goal is to learn with a single set of noisy labels – which is considerably harder. In addition, we develop a novel sparsity-promoting regularisation approach which allows us to tackle challenging high dimensional noisy settings.

## 2 Label-Noise Robust Logistic Regression

We now describe the label-noise robust logistic regression (rLR) model. We will use the term 'robust' to differentiate this from traditional logistic regression (LR). Consider a set of training data $D = \{(\mathbf{x}_1, \tilde{y}_1), \ldots, (\mathbf{x}_N, \tilde{y}_N)\}$, where $\mathbf{x}_n \in \mathbb{R}^m$ and $\tilde{y}_n \in \{0, 1\}$, where $\tilde{y}_n$ denotes the observed (possibly noisy) label of $\mathbf{x}_n$. In the classical scenario for binary classification, the log likelihood is defined as:

$$\sum_{n=1}^{N} \tilde{y}_n \log p(\tilde{y} = 1|\mathbf{x}_n, \mathbf{w}) + (1 - \tilde{y}_n) \log p(\tilde{y} = 0|\mathbf{x}_n, \mathbf{w}). \tag{1}$$

where $\mathbf{w}$ is the weight vector orthogonal to the decision boundary and it determines the orientation of the separating plane. If all the labels were presumed to be correct, we would have $p(\tilde{y} = 1|\mathbf{x}_n, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}_n) = \frac{1}{1+e^{(-\mathbf{w}^T\mathbf{x}_n)}}$ and whenever this is above 0.5 we would decide that $\mathbf{x}_n$ belongs to class 1.

However, when label noise is 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 model $p(\tilde{y} = k|\mathbf{x}_n, \mathbf{w})$ as the following:

$$p(\tilde{y} = k|\mathbf{x}_n, \mathbf{w}) = \sum_{j=0}^{1} p(\tilde{y} = k|y = j)p(y = j|\mathbf{x}_n, \mathbf{w}) \stackrel{def}{=} S_n^k \tag{2}$$

where $k \in \{0, 1\}$. Therefore, instead of Eq.(1), we define the log likelihood of our model as the following:

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \tilde{y}_n \log S_n^1 + (1 - \tilde{y}_n) \log S_n^0 \tag{3}$$

In Eq.(2), $p(\tilde{y} = k|y = j) \stackrel{def}{=} \gamma_{jk}$ represents the probability that the label has flipped from the true label $j$ into the observed label $k$. These parameters

form a transition table that we will refer to as the 'gamma matrix' from now on. Now, to classify a novel data point $\mathbf{x}_q$, we predict that $\hat{y}_q = 1$ whenever $p(y = 1|\mathbf{x}_q, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}_q) = \frac{1}{1+e^{(-\mathbf{w}^T\mathbf{x}_q)}}$ returns a value greater than 0.5, and $\hat{y}_q = 0$ otherwise.

### 2.1 Parameter Estimation with Multiplicative Updates

Learning the rLR requires us to estimate the weight vector $\mathbf{w}$ as well as the label-flipping probabilities $\gamma_{jk}$. To optimise the weight vector, we can use any nonlinear optimiser. Here we decided to employ conjugate gradients because of its well known computational efficiency, which basically performs the Newton update step along the direction $\mathbf{u} = \mathbf{g} - \mathbf{u}^{old}\beta$, where $\mathbf{g} = \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w})$ is the gradient:

$$\mathbf{g} = \sum_{n=1}^{N}\left[\left(\frac{\tilde{y}_n(\gamma_{11}-\gamma_{01})}{S_n^1} + \frac{(1-\tilde{y}_n)(\gamma_{10}-\gamma_{00})}{S_n^0}\right)\sigma(\mathbf{w}^T\mathbf{x}_n)(1-\sigma(\mathbf{w}^T\mathbf{x}_n))\cdot\mathbf{x}_n\right] \quad (4)$$

One may verify that setting $\gamma_{01}$ and $\gamma_{10}$ to 0 and $\gamma_{00}, \gamma_{11}$ to 1, after some algebra, Eq.(4) will reduce to the well-known gradient expression of classical logistic regression. The parameter $\beta$ that works best in practice can be obtained from the Hestenes-Stiefel formula, $\beta = \frac{\mathbf{g}^T(\mathbf{g}-\mathbf{g}^{old})}{(\mathbf{u}^{old})^T(\mathbf{g}-\mathbf{g}^{old})}$. Then, the update equation for $\mathbf{w}$ is simply the following:

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \frac{\mathbf{g}^T\mathbf{u}}{\mathbf{u}^T\mathbf{H}\mathbf{u}}\mathbf{u}, \quad (5)$$

where $\mathbf{H}$ is the Hessian matrix.

To obtain the updates for the label-flipping probabilities, we introduce Lagrange multipliers to ensure that $\gamma_{00} + \gamma_{01} = 1$ and $\gamma_{10} + \gamma_{11} = 1$. Conveniently, after some algebra, the stationary equations yield the following multiplicative update equations:

$$\gamma_{00} = \frac{\gamma_{00}\sum_{n=1}^{N}\left[\frac{(1-\tilde{y}_n)}{S_n^0}(1-\sigma(\mathbf{w}^T\mathbf{x}_n))\right]}{\gamma_{00}\sum_{n=1}^{N}\left[\frac{(1-\tilde{y}_n)}{S_n^0}(1-\sigma(\mathbf{w}^T\mathbf{x}_n))\right] + \gamma_{01}\sum_{n=1}^{N}\left[\frac{\tilde{y}_n}{S_n^1}(1-\sigma(\mathbf{w}^T\mathbf{x}_n))\right]} \quad (6)$$

$$\gamma_{11} = \frac{\gamma_{11}\sum_{n=1}^{N}\left[\frac{\tilde{y}_n}{S_n^1}\sigma(\mathbf{w}^T\mathbf{x}_n)\right]}{\gamma_{10}\sum_{n=1}^{N}\left[\frac{(1-\tilde{y}_n)}{S_n^0}\sigma(\mathbf{w}^T\mathbf{x}_n)\right] + \gamma_{11}\sum_{n=1}^{N}\left[\frac{\tilde{y}_n}{S_n^1}\sigma(\mathbf{w}^T\mathbf{x}_n)\right]} \quad (7)$$

Our rLR training algorithm is then to alternate between updating each parameter in turn, until convergence. It is worth noting that the objective we are trying to optimise is non-convex. Hence, the result will inevitably depend on the initialisation of those parameters, and we will return to this point in the experimental section. However, convergence to a local optimum is guaranteed, as we shall see shortly.

### 2.2 Multiclass Label-Noise Robust Logistic Regression

It is both useful and straightforward to generalise the two-class rLR of the previous section to multiclass problems. We again introduce the true class label $y$ as a latent variable and write:

$$p(\tilde{y}=k|\mathbf{x}_n,\mathbf{w}_k)=\sum_{j=0}^{K-1}p(\tilde{y}=k|y=j)\cdot p(y=j|\mathbf{x}_n,\mathbf{w}_j)\stackrel{def}{=}S_n^k \tag{8}$$

where $p(y=k|\mathbf{x}_n,\mathbf{w}_k)$ is modelled using a softmax function, $\frac{e^{(\mathbf{w}_k^T\mathbf{x}_n)}}{\sum_{l=0}^{K-1}e^{(\mathbf{w}_l^T\mathbf{x}_n)}}$, and $\mathbf{w}_k$ is the weight vector corresponding to class $k$. The maximum likelihood (ML) estimate of $\mathbf{w}_k$ is obtained by maximising the data log-likelihood,

$$\mathcal{L}(\mathbf{w})=\sum_{n=1}^{N}\sum_{k=0}^{K-1}\delta(\tilde{y}_n=k)\log S_n^k \tag{9}$$

where $\delta(\tilde{y}_n=k)$ is the Kronecker delta function that gives the value 1 when its argument is true and the value 0 otherwise. The optimisation is again accomplished using the conjugate gradient method where the gradient becomes:

$$\mathbf{g}=\sum_{n=1}^{N}\sum_{k=0}^{K-1}\frac{\delta(\tilde{y}_n=k)}{S_n^k}\times\frac{e^{(\mathbf{w}_c^T\mathbf{x}_n)}\mathbf{x}_n\left(\sum_{j=0}^{K-1}(\gamma_{ck}-\gamma_{jk})\cdot e^{(\mathbf{w}_j^T\mathbf{x}_n)}\right)}{\left(\sum_{l=0}^{K-1}e^{(\mathbf{w}_l^T\mathbf{x}_n)}\right)^2} \tag{10}$$

Further, the estimates of $\gamma_{jk}$ in the gamma matrix again can be obtained by efficient multiplicative update equations:

$$\gamma_{jk}=\frac{1}{C}\times\gamma_{jk}\sum_{n=1}^{N}\frac{\delta(\tilde{y}_n=k)}{S_n^k}\cdot\frac{e^{(\mathbf{w}_j^T\mathbf{x}_n)}}{\sum_{l=0}^{K-1}e^{(\mathbf{w}_l^T\mathbf{x}_n)}}, \tag{11}$$

where the constant term $C$ equals $\sum_{k=0}^{K-1}\gamma_{jk}\sum_{n=1}^{N}\frac{\delta(\tilde{y}_n=k)}{S_n^k}\times\frac{e^{(\mathbf{w}_j^T\mathbf{x}_n)}}{\sum_{l=0}^{K-1}e^{(\mathbf{w}_l^T\mathbf{x}_n)}}$.

To classify a new point, we decide $\hat{y}_q=\arg\max_k\frac{e^{(\mathbf{w}_k^T\mathbf{x}_q)}}{\sum_{l=0}^{K-1}e^{(\mathbf{w}_l^T\mathbf{x}_q)}}$.

## 3 Convergence of the Algorithm

We shall now prove that the learning algorithms proposed in the previous sections, for both rLR and rmLR, converge. The idea of the proof is to show that the objective function being optimised, Eq.(9) is nondecreasing under any of our parameter updates. Indeed, the maximisation w.r.t. the weight vector $\mathbf{w}$ by the conjugate gradient method (CG) enjoys the known property of CG to provide monotonically improving estimation of the target [8], which guarantees that an objective function being maximised is nondecreasing. Now, it remains to prove that our multiplicative updates for $\gamma_{jk}$ are also guaranteed to be nondecreasing. To do this, we use the notion of an auxiliary function, in a similar spirit to the proofs in [14].

**Definition 1.** *$G(h, h')$ is an auxiliary function for $F(h)$ if*

$$G(h, h') \leq F(h), G(h, h) = F(h) \tag{12}$$

*are satisfied.*

The definition is useful because of the following lemma.

**Lemma 1.** *[14] If $G$ is an auxiliary function, then $F$ is nondecreasing under the update*

$$h^{i+1} = \arg\max_{h} G(h, h^i) \tag{13}$$

*Proof.* $F(h^{i+1}) \geq G(h^{i+i}, h^i) \geq G(h^i, h^i) = F(h^i)$

We will show that by defining an appropriate auxiliary function to the objective function Eq. (9), regarded as a function of $\Gamma$, the update equations Eq.(11) for $\gamma_{jk}$ are guaranteed to converge to a local optimum.

**Lemma 2.** *Define*

$$G(\Gamma, \tilde{\Gamma}) = \sum_{n=1}^{N} \sum_{k=0}^{K-1} \delta(\tilde{y}_n = k) \sum_{j=0}^{K-1} \frac{\tilde{\gamma}_{jk} p(y = j|\boldsymbol{x}_n, \boldsymbol{w})}{\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y = l|\boldsymbol{x}_n, \boldsymbol{w})} \times \left( \log \tilde{\gamma}_{jk} p(y = j|\boldsymbol{x}_n, \boldsymbol{w}) - \log \frac{\tilde{\gamma}_{jk} p(y = j|\boldsymbol{x}_n, \boldsymbol{w})}{\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y = l|\boldsymbol{x}_n, \boldsymbol{w})} \right) \tag{14}$$

*This is an auxiliary function for*

$$\mathcal{L}(\Gamma) = \sum_{n=1}^{N} \sum_{k=0}^{K-1} \delta(\tilde{y}_n = k) \log \sum_{j=0}^{K-1} \gamma_{jk} p(y = j|\boldsymbol{x}_n, \boldsymbol{w}) \tag{15}$$

*Proof.* For $G(\Gamma, \tilde{\Gamma})$ to be an auxiliary function it needs to satisfy the aforementioned conditions. It is straightforward to verify that $G(\Gamma, \Gamma) = \mathcal{L}(\Gamma)$. To show that $G(\Gamma^{i+1}, \Gamma^i) \leq \mathcal{L}(\Gamma^{i+1})$, we observe that:

$$\log \sum_{j=0}^{K-1} \gamma_{jk} p(y = j|\mathbf{x}_n, \mathbf{w}) \geq \sum_{j=0}^{K-1} \alpha_{jk} \log \left( \frac{\gamma_{jk} p(y = j|\mathbf{x}_n, \mathbf{w})}{\alpha_{jk}} \right), \tag{16}$$

by Jensen's inequality and due to the convexity of the log function. This inequality is valid for all non-negative $\alpha_{jk}$ that sum to one. Setting

$$\alpha_{jk} = \frac{\tilde{\gamma}_{jk} p(y = j|\mathbf{x}_n, \mathbf{w})}{\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y = l|\mathbf{x}_n, \mathbf{w})}, \tag{17}$$

we see that our objective function $\mathcal{L}(\Gamma)$ is always greater than or equal to the auxiliary function (14).

**Lemma 3.** *The multiplicative update rule of the label flipping probability* $\gamma_{jk}$ *given in Eq. (11) is guaranteed to converge.*

*Proof.* The maximum of $G(\Gamma, \tilde{\Gamma})$ with respect to $\gamma_{jk}$ is found by setting the derivative to zero:

$$\frac{dG(\Gamma, \Gamma^i)}{d\gamma_{jk}} = \sum_{n=1}^{N} \delta(\tilde{y}_n = k)\frac{\alpha_{jk}}{\gamma_j k} - \lambda = 0, \tag{18}$$

Using the fact that $\sum_j \gamma_{jk} = 1$, we obtain the value of the Lagrange multiplier $\lambda$. Putting it back into Eq. (18) we arrive at:

$$\gamma_{jk} = \frac{1}{C} \times \tilde{\gamma}_{jk} \sum_{n=1}^{N} \delta(\tilde{y}_n = k) \cdot \frac{p(y = j|\mathbf{x}_n, \mathbf{w})}{\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y = l|\mathbf{x}_n, \mathbf{w})}, \tag{19}$$

where $C$ equals $\sum_{k=0}^{K-1} \tilde{\gamma}_{jk} \sum_{n=1}^{N} \delta(\tilde{y}_n = k) \frac{p(y=j|\mathbf{x}_n, \mathbf{w})}{\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y=l|\mathbf{x}_n, \mathbf{w})}$. Writing out posterior probability $p(y = j|\mathbf{x}_n, \mathbf{w})$ as a softmax function and noting that by definition $\sum_{l=0}^{K-1} \tilde{\gamma}_{lk} p(y = l|\mathbf{x}_n, \mathbf{w})$ equals $S_n^k$, Eq. (19) then takes the same form as the update rule in Eq. (11). Since $G(\Gamma, \tilde{\Gamma})$ is an auxiliary function, it is guaranteed that the value of $\mathcal{L}$ is nondecreasing under this update.

**Theorem 1.** *By alternating between the updates of the weight vector* $\boldsymbol{w}$ *while the matrix* $\Gamma$ *is held fixed, and the updates of the elements of* $\Gamma$ *while* $\boldsymbol{w}$ *is fixed, the objective function of rmLR is nondecreasing and is thus guaranteed to converge.*

*Proof.* The proof follows directly from the fact that optimising $\mathbf{w}$ using CG is monotonically nondecreasing and from *Lemma 3*, that optimising $\Gamma$ is also nondecreasing. Consequently, the objective function being optimised is monotonically increasing under alternating these iterations.

Finally, note that as rmLR is a direct generalisation of rLR, the proof also covers the case of rLR.

### 3.1 Comparison with EM Based Optimisation

The algorithm developed in [23] in the context of multiple sets of noisy labels could also be instantiated for our problem, as an alternative to the above approach. The method in [23] proposes an EM algorithm where the true labels are the hidden variables. Instead, we had these hidden variable integrated out when optimising the parameters. It is hence interesting to see how they compare.

Similar to [23], let $y_n$ be the hidden true labels, and denote $P_n = p(y_n = 1|\mathbf{x}, \mathbf{w}, \tilde{y}_n)$ the posterior of these. Then, the expected complete log likelihood (so-called Q-function) can then be written as:

$$\mathcal{Q}(\mathbf{w}, \Gamma) = \sum_{n=1}^{N} P_n \log(\gamma_{11}^{\tilde{y}_n} \gamma_{10}^{1-\tilde{y}_n} \sigma(\mathbf{w}^T \mathbf{x}_n)) + (1 - P_n) \log(\gamma_{01}^{\tilde{y}_n} \gamma_{00}^{1-\tilde{y}_n} (1 - \sigma(\mathbf{w}^T \mathbf{x}_n))) \tag{20}$$

– The E-step involves optimising $P_n$ based on given data and current estimated of $\gamma_{jk}$:

$$P_n = \frac{\gamma_{11}^{\tilde{y}_n}\gamma_{10}^{1-\tilde{y}_n}\sigma(\mathbf{w}^T\mathbf{x}_n)}{\gamma_{11}^{\tilde{y}_n}\gamma_{10}^{1-\tilde{y}_n}\sigma(\mathbf{w}^T\mathbf{x}_n) + \gamma_{01}^{\tilde{y}_n}\gamma_{00}^{1-\tilde{y}_n}(1-\sigma(\mathbf{w}^T\mathbf{x}_n))} \tag{21}$$

– The M-step then re-estimate the value of $\gamma_{jk}$ using $P_n$ from the E-step. For example $\gamma_{11}$ can be update using:

$$\gamma_{11} = p(\tilde{y}_n = 1|y_n = 1) = \frac{\sum_{n=1}^{N} P_n\tilde{y}_n}{\sum_{n=1}^{N} P_n} \tag{22}$$

Now, observe that substituting the r.h.s. of $P_n$ into the M-step equations, we recover our multiplicative form of updates – with one subtle but important difference: In the EM approach $P_n$ is computed with old values of the parameters (from the previous iteration). Instead, our multiplicative updates use the latest fresh values of all the parameters they depend on. This implies that our algorithm has a better chance to converge in fewer iterations, and in addition it saves the storage cost of the posteriors $P_n$ during the iterations. Worth noting also that $P_n$ can be useful for interpretation — however we can compute this after convergence using the final values of the parameters.

## 4 Sparse Extension via a Bayesian-Regularised Generalised Lasso

In many real world problems, especially in biomedical domains, we are faced with high dimensional data with more features than observations, while only a subset of the features is relevant to the target. Sparsity-inducing regularisation approaches have been effective in such cases [19,24,4]. In this section we show that our model can be extended to support such regularised estimation. Akin to generalised Lasso [24], we will employ L1-regularisation terms on each component of $\mathbf{w}$. We should mention that other approaches such as Automatic Relevance Determination based on t-prior [19] could also be used in a similar manner.

Our regularised objective is now the following:

$$\max_{\mathbf{w}} \sum_{n=1}^{N} \log p(\tilde{y}_n|\mathbf{x}_n, \mathbf{w}) - \sum_{i=1}^{m} \alpha_i|w_i| \tag{23}$$

where $m$ is the number of features and $\alpha_i$ are Lagrange multipliers that balance between fitting the data well and having small parameter values. Eq.(23) is not differentiable at the origin. To counter this, here we adopt a very simple, yet effective smooth approximation originally proposed in [16] for Lq-regularisation. This is to approximate $|w_i| \approx (w_i^2 + \gamma)^{1/2}$, and we have set $\gamma = 10^{-8}$ in the reported experiments.

Now, the regularisation parameters $\alpha_i$ need to be determined. The common approach would be to use cross-validation — however, this would need to make use of the labels of the validation sets, which have no guarantee of being correct

in our problem setting. We turn to a Bayesian regularisation approach where $\alpha_i$ is eliminated from the model by marginalisation. Bayesian regularisation was found comparable in performance to cross-validation [5], and in particular it was also demonstrated to be effective for L1-regularised logistic regression [4].

Our version will be different from the one in [4] mainly because the latter is tied to their specific implementation in that $\alpha_i = \alpha$ for all $i$ for which $w_i \neq 0$, and a Jeffreys hyperprior is posited only on these non-zero components. This requires an estimate of the number of non-zeros. Instead, we will simply posit independent Jeffreys priors on each $\alpha_i$ and let the ones that are not supported by the data die out naturally.

We begin by considering a Bayesian interpretation of the problem in Eq.(23). That is, the posterior distribution of $\mathbf{w}$, conditional on $\boldsymbol{\alpha}$, can be written as

$$p(\mathbf{w}|\mathcal{D}, \boldsymbol{\alpha}) \propto p(\mathcal{D}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha}). \tag{24}$$

Now the first term on the r.h.s is the data likelihood, while the second term corresponds to our regularisation term. If we take logarithm of the whole expression, we have: $\log p(\mathbf{w}|\mathcal{D}, \boldsymbol{\alpha}) = \log p(\mathcal{D}|\mathbf{w}) + \log p(\mathbf{w}|\boldsymbol{\alpha}) + const.$

Thus, the regularisation term in Eq.(23) is just the negative logarithm of the conditional prior distribution, conditioned on $\boldsymbol{\alpha}$, up to an additive constant. The conditional prior $p(\mathbf{w}|\boldsymbol{\alpha})$ is then given by a product of independent Laplace distributions with parameters $\boldsymbol{\alpha}$: $p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{m} p(w_i|\alpha_i) = \frac{\prod_{i=1}^{m} \alpha_i}{2^m} \exp\left(-\sum_{i=1}^{m} \alpha_i |w_i|\right) \approx \frac{\prod_{i=1}^{m} \alpha_i}{2^m} \exp\left(-\sum_{i=1}^{m} \alpha_i (w_i^2 + \gamma)^{1/2}\right)$. Now, we want to eliminate its dependency on $\boldsymbol{\alpha}$ by marginalisation, i.e. to have the marginal prior as the following:

$$p(\mathbf{w}) = \int p(\mathbf{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\alpha} \tag{25}$$

For this, we posit Jeffrey's priors, $p(\alpha_i) \propto \frac{1}{\alpha_i}$, on each $\alpha_i$. This is the non-informative improper limit of a Gamma prior, and it has the advantage that it is parameter-free. Substituting this and $p(w_i|\alpha_i)$ and performing the integral in Eq. (25), $\int_0^\infty \frac{1}{\alpha_i} \frac{\alpha_i}{2} \exp(-\alpha_i (w_i^2 + \gamma)^{1/2}) d\alpha_i = \frac{1}{2(w_i^2+\gamma)^{1/2}}$,we obtain the following marginal prior:

$$p(\mathbf{w}) = \frac{1}{2} \prod_{i=1}^{m} \frac{1}{(w_i^2 + \gamma)^{1/2}}, \tag{26}$$

which implies that negative log of the marginal prior $-\log p(\mathbf{w}) = \sum_{i=1}^{m} \log\left((w_i^2 + \gamma)^{1/2}\right) + const.$ Now, replacing the regularisation term that appears in Eq.(23) by the above marginal prior, and taking derivative with respect to the model parameters $w_i$ again as we did before, now we have:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{n=1}^{N} \left[\tilde{y} \log(S_n^1) + (1 - \tilde{y}) \log(S_n^0)\right] + \frac{1}{(w_i^2 + \gamma)^{1/2}} \frac{\partial}{\partial w_i} \left( \sum_{i=1}^{m} \log((w_i^2 + \gamma)^{1/2}) \right) \tag{27}$$

From this, we read off the estimates of the regularisation parameters as:

$$\alpha_i = \frac{1}{(w_i^2 + \gamma)^{1/2}} \tag{28}$$

The optimisation of the log-likelihood is then to alternate between optimising $\mathbf{w}$ along with updating $\alpha_i$ according to Eq.(28) until convergence in reached, and of course, we alternate this with the fixed point update equations of the label flipping probabilities given in the previous sections. Generalising the sparse regression procedure described in this section to multi-class settings is straightforward.

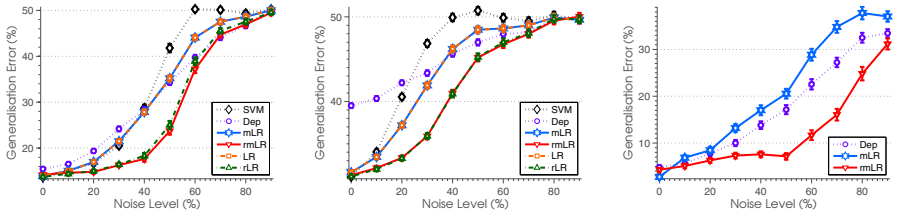
## 5 Experimental Validation and Applications

### 5.1 Simulated Label Noise

Before presenting real applications where no ground truth is available for an objective validation, we first assess our algorithm on real world data sets using artificial class label noise. We used three standard data sets from the UCI repository: *Boston*, *Liver* (binary) and *Iris* (multiclass). Since it has been shown theoretically that symmetric label noise is relatively harmless, for example see [18], here only asymmetric label noise of various levels was artificially injected for the purpose of systematic testing. In addition, we will compare our result to two existing methods: (i) Depuration [2], which is a non-parametric method based on nearest neighbours, previously proposed for the same problem of dealing with label-noise in classification; and (ii) Support Vector Machines (SVM), which has the well-known margin and slack-variable mechanism built in, and which may provide some robustness. The reason to compare with SVM is to find out to what extent class label noise could be considered to be a normal part of any classification problem — and conversely, to what extent it actually needs the special treatment that we developed in the previous sections. Code that reproduces the results of our experiments is available on request.

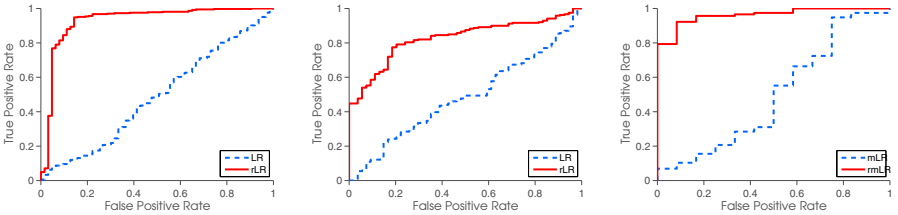
It should be noted that when applying Depuration and SVM, we again face with the problem of model selection. A general approach to model selection is a standard cross validation technique. Although this works well in a traditional setting where all class labels are correct, it is no longer applicable here. This problem was also reported in [13], However, the solution they resort to is simply to assume that there is a trusted validation set available. This may be unrealistic in many real situations, and especially so in small-sample problems as in [26].

Figure 1 summarises our results on three classification data sets. It is clear that both rLR and rmLR outperform each algorithm on each of the data sets tested. Depuration (denoted as 'Dep' in the figure) tends to perform well in a very high level of noise (i.e. 50% upwards) while at the lower range, its performance is slightly worse. The comparative results with SVM also demonstrate convincingly that class label noise does need special attention and it is naive to consider label noise as a normal part of classification problems. We see that our algorithm developed explicitly for this problem does indeed achieve improved classification performance overall.




(a) Boston (Binary) (b) Liver (Binary) (c) Iris (Multiclass)

**Fig. 1.** Classification errors on real world data sets when the labels are artificially flipped asymmetrically




(a) Boston (Binary) (b) Liver (Binary) (c) Iris (Multiclass)

**Fig. 2.** ROC curves. Labels are asymmetrically flipped at 30% noise.

Next, we assess our methods' ability to detect the instances that were wrongly labelled. There are two types of possible errors: (i) a false positive is when a point is believed to be mislabelled when in fact it is labelled correctly; and (ii) a false negative is when a point is believed to be labelled correctly when in fact its label is incorrect. A good way to summarise both, while also using the probabilistic output given by the sigmoid or the softmax functions, may be obtained by constructing the Receiver Operating Characteristic (ROC) curves. Figure 2 shows the ROC curves for all real world data sets tested, at a asymmetric noise level of 30%. Superimposed for reference we also plotted the ROC curves that correspond to the traditional classifier that believes that all points have the correct labels. The gap between the two curves is well apparent in all four cases tested, and it quantifies the gain obtained by our modelling approach in each setting. The area under the ROC curve signifies the probability that a randomly drawn and mislabelled example would be flagged by our method. For the sake of clarity of the graph, the results from Depuration and SVM were not included here as we have already seen that they are inferior to rLR and rmLR. We now turn to demonstrating real applications in several domains.

### 5.2 Application to Finding Mislabelled Gene Arrays in Colon Cancer Data

So far we presented controlled experiments where the label-noise was artificially created. It is now most interesting to demonstrate the effectiveness of our

approach on a data set whose labels are genuinely inaccurate. In this section we take the Colon Cancer data set [1]. This contains expression levels of 2000 genes from 40 tumour and 22 normal colon tissues, and there is some evidence in the biological literature that label noise may be present [6,15,11,21,26].

We split the data into 52 training points and 10 test points. In order to get a more reliable accuracy figure, we have excluded from the test set all of the instances which are suspected to be wrong based on the existing evidence. Instead, these instances will be placed in the training set in all the training-testing splits that we consider. We note that the number of instances affected by label noise is unequal for the two classes: approximately 20% of normal tissues were labelled as tumour while 10% of tumour tissues were labelled as normal. The nature of mislabelling corresponds to a slightly asymmetric flipping scenario that we have previously discussed. Hence the label noise will likely perturb the learning of traditional classifiers.

For illustrative purpose, we evaluate predictive performance on the test set averaged over 1000 training-testing splits and observe that rLR significantly outperform its traditional competitors and achieves an impressive accuracy with the error rate of $2.08 \pm 0.055$, while the performance of LR ($3.66 \pm 0.064$) and SVM ($4.08 \pm 0.063$) lag behind. We should note, these results are not directly comparable to other studies where the mislabelled points have not been excluded from the test set.

More importantly, biologists are interested in understanding the nature of data rather than classification accuracy figures. Here we demonstrate the use of algorithm for detecting the wrongly labelled instances. This is particularly useful in scientific applications, where a sample detected as potentially mislabelled could then be handed over to the domain expert for confirmation or further study. In Table 1 we compare the results of previous attempts at this problem with rLR.

The penultimate line gives the frequency rates of mislabelling detections computed from 20 independent runs on the whole data set, with independent random initialisation, and using $\sum_{j=0, j\neq\tilde{y}_n}^{K-1} p(y = j|\mathbf{x}_n, \mathbf{w})$ thresholded at 0.5 each time. Since the objective function that we optimise is non-convex, as mentioned previously, our greedy iterative algorithm finds one of its local optima at each run, and hence different runs can come up with different detections depending on the initialisation. A frequency rate of 1 means that the probability that the true label differs from the observed one for that point was estimated to be higher than 0.5 in all of the 20 repeated runs. A value of 0.15 means that it was estimated to be higher than 0.5 in 3 out of the 20 runs.

We observe that rLR can identify up to 8 distinct mislabelled points, and these cover all except one of the union of all previously identified mislabellings, and do not detect any other point outside this union. We also note that this total of 8 points were not identified at any single run of our algorithm either. This suggests that in this case having several local optima is not necessarily a bad thing for the application, as it allows us to have different views at the problem which may be more comprehensive than a simplified single view. The last row provides

**Table 1.** Identifying mislabelled points from the Colon Cancer data set. The first row is the 'gold standard' with biological evidence. The last two rows present our results (see text for explanations). The rest are the results from previous studies.

| Source | Suspects identified | | | | | | | | | Extra samples identified |
|---|---|---|---|---|---|---|---|---|---|---|
| Alon et al. [1] | T2 | T30 | T33 | T36 | T37 | N8 | N12 | N34 | N36 | |
| Furey et al. [6] | | ○ | ○ | ○ | | ○ | | ○ | ○ | |
| Li et al. [15] | | ○ | ○ | ○ | | | | ○ | ○ | |
| Kadota et al. [11] | ○ | | | | ○ | ○ | | ○ | ○ | T6,N2 |
| Malossini et al. [21] | ○ | ○ | ○ | ○ | | | ○ | ○ | ○ | T8,N2,N28,N29 |
| reg-rLR by frequency | 0.15 | 1.00 | 1.00 | 1.00 | 0 | 0.25 | 0.1 | 1.00 | 1.00 | |
| reg-rLR degree of belief | 0 | 0.70 | 0.66 | 0.76 | 0 | 0.54 | 0.54 | 0.59 | 0.60 | |

the degree of belief, ie. the actual probabilities from $\sum_{j=0,j\neq\tilde{y}_n}^{K-1} p(y=j|\mathbf{x}_n,\mathbf{w})$, without any thresholding, for the single best run out of the 20 independent trials, selected by the best minimum of the objective function being minimised by our algorithm. Seven points, namely T30,T33,T36,N8,N12,N34 and N36 were detected in this best run. The probabilities that we see here mean the confidence of each of these detections. Relating these back to the previous row of the table, the frequency rates, we see that those samples that were identified with a higher degree of belief (T33, T36, N34 and N36) also have a higher frequency of being detected.

### 5.3 Application to Structure Discovery: Inferring a Class-Topology in Multi-class Problems

The next experiment demonstrates a different use of our label-noise robust classifier, namely to infer the internal topological structure of the data classes. For many real-world classification tasks the labelling process is somewhat subjective as there is no clear-cut boundary between the classes. For example, in the case of classifying text messages according to topic, some instances could be assigned to more than one category. Thus, interpreting the gamma matrix as the adjacency matrix of a directed graph could reveal the internal structure of the data set under study. To demonstrate this idea, we employed rmLR on *Newsgroups*[1] data set. The corpus was subject to tokenisation, stop words removal, and Porter stemming to remove the word endings prior to cosine normalisation.

Figure 3 shows the graph derived from the gamma matrix as obtained from *10 Newsgroups*. Each node corresponds to a topic class while the length of an edge connecting two nodes represents the strength of relationship between them. The direction of arrows then correspond to the label flipping directions. It can be seen from this graph that 'atheism' and 'religion' are related topics by looking at the distance between the two as well as the bi-directional flipping relation,

[1] Originally the *Newsgroups* corpus comprises 20 classes of postings, We use the subset of 10 classes from [10], with term frequency count based encoding.

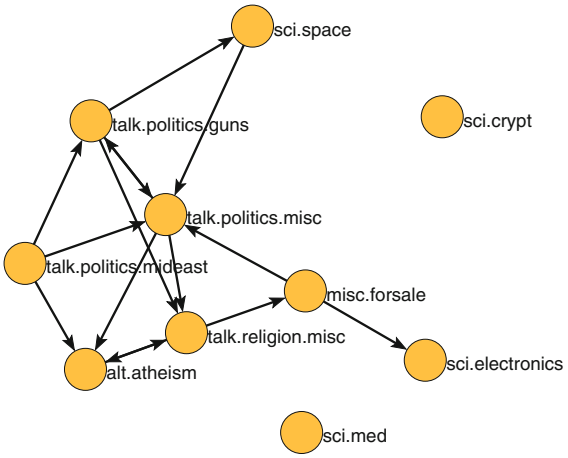



**Fig. 3.** Adjacency graph of the ten topics on the Newsgroups data set

which indeed agrees with our commonsense. Similar observation can also be made between the 'electronics' and 'for-sale' postings. Further, the graph also visually suggests various sub-groupings: for example, all classes related to politics are clustered nearer to each other.

### 5.4 Application to Learning from Crowds: Learning to Classify Images Using Cheaply Obtained Labelled Data

It is well reckoned that careful labelling of large amounts of data by human experts is extremely tiresome. Suppose we were to train a classifier to distinguish an image of 'bike' from other type of images. The standard machine learning approach is to collect training images and manually label each of them — rather labourious. Here, we suggest that we could reduce human expert intervention and obtain the training data cheaply using annotated data from search engines. By searching for images using keyword 'bike' we obtain a set of images that are loosely categorised into 'bike' class, and similarly 'not bike' class by using its negation. This allows us to acquire a large number of training data quickly and cheaply. The problem is of course that the annotations returned by the search engine are somewhat unreliable. This is where rLR comes into play. Here we collected 520 images using the keyword 'bike' and 520 images using the keyword 'not bike' that we call the *WebSearch*[2] dataset. We also manually label all images: a 'bike' image is one that contains a bike as its main object and we make no distinction between a bicycle and a motorbike, everything else is labelled as 'not bike'. This reveals 83 flips from 'bike' to 'not bike' images and 100 flips from 'not bike' to 'bike' category. The manually labelled set is only used for testing purposes. The images are passed through a series of preprocessing including

[2] Collected using Google image search engine: available upon request.

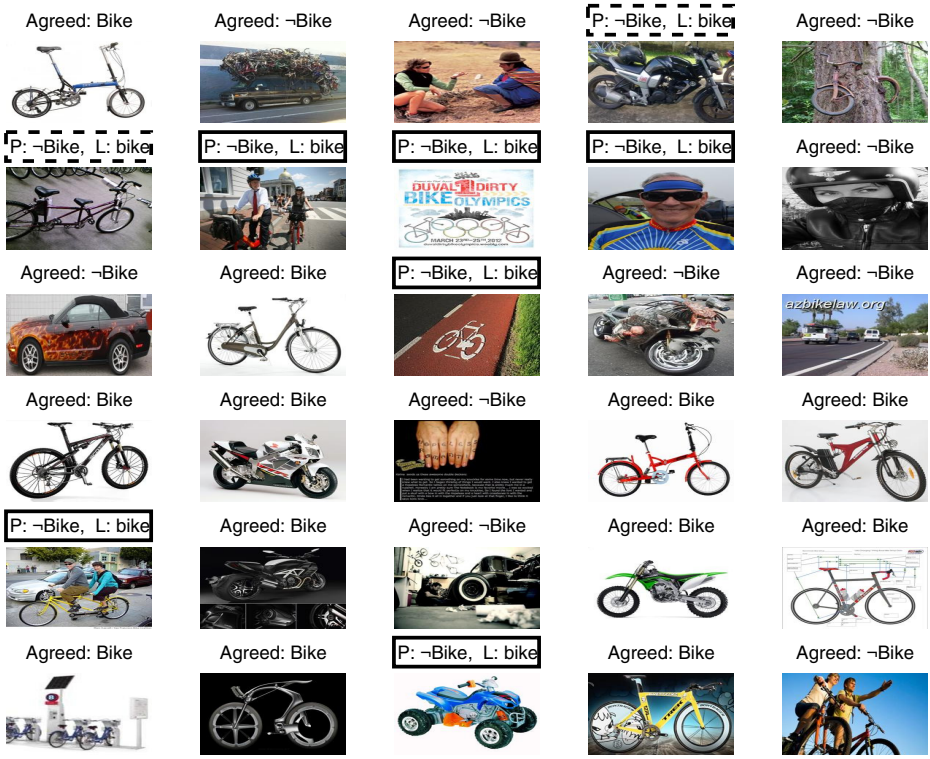



**Fig. 4.** Bike search result. P is the prediction from the classifier while L is the given label from search engine. Boxed instances are the ones that P and L don't agree while dotted boxes are false alarms.

extracting meaningful visual vocabulary using SIFT [17] and extracting texture information using LBP [22], which are ultimately transformed into a 10038-dimensional vector representation.

In Figure 4 we show examples of detecting mislabelled images. The top 30 test images sorted by their posterior probabilities are shown. We see that out of a total of 8 suspicious detections made (boxed), only 2 were false alarms (denoted by dotted box in the figure). Comparatively, the traditional LR model produced 4 false alarms (not shown). To give statistical figures, we then tested these two classifiers that were both trained on 90% of whole dataset using the cheap noisy labels from the search engine, and tested on the ramaining 10%, against the manual labels. We performed 100 independent bootstrap repetitions of this experiment. The average generalisation errors and standard errors were 15.67% $\pm$ 0.04 for rLR and 18.09% $\pm$ 0.04 for standard LR. The improvement of rLR over LR is statistically significant, as tested at the 5% level using a Wilcoxon Rank Sum test. This suggests that there is high potential for learning from unreliable data from the Internet using the label-noise robust algorithm proposed.

## 6 Conclusions

We proposed an efficient algorithm for learning a label-robust logistic regression algorithm for both two-class (rLR) and multiclass (rmLR) classification problems, and we proved its local convergence. We also developed a Bayesian sparse regularised extension for these methods which bypasses the need to perform cross validation for model selection and is hence label-robust in its model selection procedure as well. We demonstrated the working and the advantages of our approach in both controlled synthetic settings and in real applications. In particular, we have seen an application in the biomedical domain, where our method can be used to flag suspicious labels for further follow-up study. We have also seen that the label-flipping probabilities provide an interpretable holistic graphical view of data sets by unearthing the topology that underlies the data classes. Finally, the model can be used to facilitate the task of annotating training examples since it is now possible to learn the classifier from sloppily labelled but cheaply obtained data from crowds. Extending this approach to non-linear classifiers is the subject of our future work.

## References

1. Alon, U., Barkai, N., Notterman, D.A., Gishdagger, K., Ybarradagger, S., Mackdagger, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences of the United States of America 96(12), 6745–6750 (1999)
2. Barandela, R., Gasca, E.: Decontamination of Training Samples for Supervised Pattern Recognition Methods. In: Amin, A., Pudil, P., Ferri, F., Iñesta, J.M. (eds.) SPR 2000 and SSPR 2000. LNCS, vol. 1876, pp. 621–630. Springer, Heidelberg (2000)
3. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167 (1999)
4. Cawley, G.C., Talbot, N.L.C.: Gene selection in cancer classification using sparse logistic regression with bayesian regularization. Bioinformatics/Computer Applications in The Biosciences 22, 2348–2355 (2006)
5. Cawley, G.C., Talbot, N.L.C.: Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. J. Mach. Learn. Res. 8, 841–861 (2007)
6. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 906–914 (2000)
7. Hausman, J.A., Abrevaya, J., Scott-Morton, F.M.: Misclassification of the dependent variable in a discrete-response setting. Journal of Econometrics 87(2), 239–269 (1998)
8. Hestenes, M.R., Stiefel, E.: Methods of Conjugate Gradients for Solving Linear Systems. Journal of Research of the National Bureau of Standards 49(6), 409–436 (1952)
9. Jiang, Y., Zhou, Z.-H.: Editing Training Data for *k*NN Classifiers with Neural Network Ensemble. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 356–361. Springer, Heidelberg (2004)

10. Kabán, A., Tiňo, P., Girolami, M.: A General Framework for a Principled Hierarchical Visualization of Multivariate Data. In: Yin, H., Allinson, N.M., Freeman, R., Keane, J.A., Hubbard, S. (eds.) IDEAL 2002. LNCS, vol. 2412, pp. 518–523. Springer, Heidelberg (2002)
11. Kadota, K., Tominaga, D., Akiyama, Y., Takahashi, K.: Detecting outlying samples in microarray data: A critical assessment of the effect of outliers on sample classification. Chem. Bio. Informatics Journal 3(1), 30–45 (2003)
12. Krishnan, T., Nandy, S.C.: Efficiency of discriminant analysis when initial samples are classified stochastically. Pattern Recognition 23(5), 529–537 (1990)
13. Lawrence, N.D., Schölkopf, B.: Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of the 18th International Conference on Machine Learning, pp. 306–313. Morgan Kaufmann (2001)
14. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 556–562. MIT Press (2001)
15. Li, L., Darden, T.A., Weingberg, C.R., Levine, A.J., Pedersen, L.G.: Gene assessment and sample classification for gene expression data using a genetic algorithm / k-nearest neighbor method. In: Combinatorial Chemistry and High Throughput Screening, pp. 727–739 (2001)
16. Liu, Z., Jiang, F., Tian, G., Wang, S., Sato, F., Meltzer, S.J., Tan, M.: Sparse logistic regression with lp penalty for biomarker identification. Statistical Applications in Genetics and Molecular Biology 6(1), 6 (2007)
17. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, ICCV 1999, vol. 2, pp. 1150–1157. IEEE Computer Society, Washington, DC (1999)
18. Lugosi, G.: Learning with an unreliable teacher. Pattern Recogn. 25, 79–87 (1992)
19. Mackay, D.J.C.: Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6, 469–505 (1995)
20. Magder, L.S., Hughes, J.P.: Logistic regression when the outcome is measured with uncertainty. American Journal of Epidemiology 146(2), 195–203 (1997)
21. Malossini, A., Blanzieri, E., Ng, R.T.: Detecting potential labeling errors in microarrays by data perturbation. Bioinformatics 22(17), 2114–2121 (2006)
22. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)
23. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of Machine Learning Research 11, 1297–1322 (2010)
24. Roth, V.: The generalized lasso. IEEE Transactions on Neural Networks 15, 16–28 (2004)
25. Yasui, Y., Pepe, M., Hsu, L., Adam, B.L., Feng, Z.: Partially supervised learning using an emboosting algorithm. Biometrics 60(1), 199–206 (2004)
26. Zhang, C., Wu, C., Blanzieri, E., Zhou, Y., Wang, Y., Du, W., Liang, Y.: Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model. Bioinformatics 25, 2708–2714 (2009)