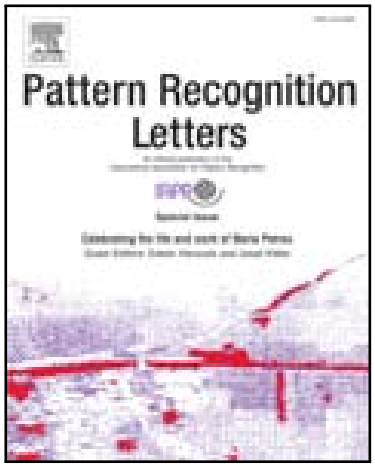
Fast estimation for robust supervised classification with mixture model

Erwan Giry Fouquet, Mathieu Fauvel, Clément Mallet

Journal Pre-proof



Fast estimation for robust supervised classification with mixture models

Erwan Giry Fouquet, Mathieu Fauvel, Clément Mallet

PII: S0167-8655(21)00379-2
DOI: https://doi.org/10.1016/j.patrec.2021.10.020
Reference: PATREC 8403

To appear in: *Pattern Recognition Letters*

Received date: 26 March 2021
Revised date: 3 September 2021
Accepted date: 18 October 2021

Please cite this article as: Erwan Giry Fouquet, Mathieu Fauvel, Clément Mallet, Fast estimation for robust supervised classification with mixture models, *Pattern Recognition Letters* (2021), doi: https://doi.org/10.1016/j.patrec.2021.10.020

Highlights

- The Robust Mixture Discriminant Analysis model leads to a convex optimization problem
- A consensus based formulation can be solved efficiently using ADMM
- This formulation scales efficiently to thousands of samples
- The main limitation of the Robust Mixture Discriminant Analysis comes from the clustering

Pattern Recognition Letters
journal homepage: www.elsevier.com

Fast estimation for robust supervised classification with mixture models

Erwan Giry Fouquet[a,**], Mathieu Fauvel[a], Clément Mallet[b]

[a] *CESBIO, Université de Toulouse, CNES/CNRS/INRAe/IRD/UPS, Toulouse, FRANCE*
[b] *Univ. Gustave Eiffel, IGN-ENSG, LaSTIG, Saint-Mande, FRANCE*

ABSTRACT

Label noise is known to negatively impact the performance of classification algorithms. In this paper, we develop a model robust to label noise that uses both labelled and unlabelled samples. In particular, we propose a novel algorithm to optimize the model parameters that scales efficiently w.r.t. the number of training samples. Our contribution relies on a consensus formulation of the original objective function that is highly parallelizable. The optimization is performed with the Alternating Direction Method of Multipliers framework. Experimental results on synthetic datasets show an improvement of several orders of magnitude in terms of processing time, with no loss in terms of accuracy. Our method appears also tailored to handle real data with significant label noise.

1. Introduction

Success of fully supervised machine learning algorithms depends on the availability of labeled databases in order to train the model parameters. Such databases contain samples, explanatory variables, and their associated response variables. Usually, the response variables (*e.g.*, a categorical variable for classification problems) are obtained through manual annotation. For complex problems or repetitive tasks, this is prone to labeling errors, resulting in noise in the response variables (Frénay and Verleysen, 2014). Some studies have reported noise up to 40% of noise for classification labels (Lee et al., 2017).

Three main strategies for label noise modeling are reported in the literature. The simpler one assumes that the mislabeling is, conditionally to the true label, independent of the explanatory variables, i.e., the noise is uniform among the labels and the probability of confusion is symmetric. A second modeling assumes label-dependent noise: some confusions are more probable than others. Finally, a more realistic strategy postulates that the label noise depends both on the response and explanatory variables. However, its complexity has hindered its adoption, resulting in more studies focusing on the two first strategies (Song et al., 2020).

The presence of noise can dramatically affect the learning algorithm, which leads to overfit data for complex algorithms such as deep neural networks (Zhang et al., 2016). Hence, since the seminal paper of Frénay and Verleysen (2014), emphasis has been put, in the machine learning community, in building label noise resistant algorithms. Recent surveys can be found in (Algan and Ulusoy, 2020; Han et al., 2020; Song et al., 2020). Currently, main efforts are oriented to design dedicated strategies on deep learning architectures (Yao et al., 2019), on sample selection (Shen and Sanghavi, 2019) or combining sample selection and loss correction (Song et al., 2019).

Alternatively, semi-supervised (SS) algorithms are used to address the label noise issue. SS learning has initially emerged to deal with databases with few labeled samples but with a large number of unlabeled samples (Chapelle et al., 2006). It has found recently application in learning with label noise (Bouveyron and Girard, 2009; Yan et al., 2016; Ding et al., 2018; Kong et al., 2019; de Aquino Afonso and Berton, 2020; Li et al., 2020). Most of the SS strategies rely either on iteratively collecting unlabeled samples to mitigate the effect of noisy labels, or on using a class overlap measure computed on unlabeled samples.

Under the SS learning framework, Bouveyron and Girard (2009) have considered two structures in the data: an unsupervised modeling based on mixture models and a supervised modeling relying on the label information. This strategy is based on the *cluster assumption* from SS learning (Chapelle et al., 2006, Chap. 1): "*if some learning data have wrong*

**Corresponding author
e-mail: mathieu.fauvel@inrae.fr (Mathieu Fauvel)

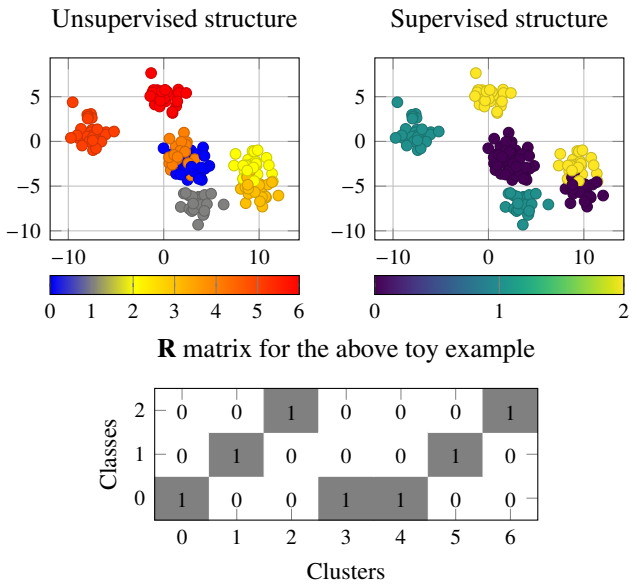


Fig. 1. Robust Mixture Discriminant Analysis modeling: this 2-dimensional synthetic data exemplifies the data model. The colors represent the clusters (7) and the classes (3) for the unsupervised and supervised structures, respectively. The synthetic R matrix that links the unsupervised structure to the supervised structure is given below the scatter plots.

labels, the comparison of the supervised information with an unsupervised modeling of the data allows to detect the inconsistent labels" (Bouveyron and Girard, 2009). Their model is optimized through the maximization of the log-likelihood and provides very good results on several datasets (*e.g.*, USPS and Pascal VOC). However, their original constrained formulation does not scale well with the number of samples and therefore is not applicable to recent large-scale data.

The contribution of this letter is to describe a new algorithm embedding such a SS framework that scales efficiently w.r.t. the number of training samples. The proposed algorithm is based on *Alternating Direction Method of Multipliers* (ADMM) framework (Boyd, 2010). We derive a consensus-based approach of the original algorithm which exploits the convexity of the problem. Results on simulated data demonstrate the speed-up of the proposed algorithm by a factor larger than 100 when the number of training is large. Then, the results on real datasets emphasis the potential of such modeling for data with significant label noise but also the current limitation.

The remainder of the letter is organised as follows. Section 2 presents the original model of Bouveyron and Girard (2009). Then, the proposed algorithm is presented in Section 3. Section 4 shows the superiority of the proposed algorithm w.r.t the original solver on synthetic datasets and Section 5 provides classification results on three real datasets.

2. Robust Mixture Discriminant Analysis

In the following, the data samples are denoted $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. We assume that they are independent realizations of a random vector $X \in \mathbb{R}^d$. It is also assumed that a subset of size $n \ll N$ of the data is provided with their corresponding labels $\{c\}_{\ell=1}^n$ represented by the random discrete variable $C \in \{1, \ldots, k\}$.

This labeled set $\{(\mathbf{x}_\ell, c_\ell)\}_{\ell=1}^n$ represents the supervised structure of the data (Bouveyron and Girard, 2009). The authors also assumed the coexistence of an unsupervised structure of K clusters, represented by the random discrete variable $S \in \{1, \ldots, K\}$. Both structures are illustrated in Figure 1.

Using the conventional mixture model framework, the density function of X is $p(\mathbf{x}) = \sum_{j=1}^K p(X = \mathbf{x}|S = j)p(S = j)$. It can be seen as a statistical representation of the data, whatever the classes of interest. Assuming the classes fully represent the dataset, *i.e.*, each pixel belongs to at least one class, this yields to : $\sum_{i=1}^k p(C = i|S = j) = 1,\ \forall j \in \{1, \ldots, K\}$. Combining these two equalities, Bouveyron and Girard (2009) defined the *Robust Mixture Discriminant Analysis* (RMDA) model:

$$p(\mathbf{x}) = \sum_{i,\ j=1}^{k,\ K} p(C = i|S = j)p(X = \mathbf{x}|S = j)p(S = j). \quad (1)$$

The term $p(C = i|S = j)$ can be interpreted as the probability that the j^{th} cluster belongs to the i^{th} class. For simplicity, it is denoted r_{ij} hereafter.

Noting $\mathbf{r}_i = [r_{i1}, \ldots, r_{iK}]^\top\ \forall i \in \{1, \ldots, k\}$, and $\boldsymbol{\psi} = [p(S = 1|X = \mathbf{x}), \ldots, p(S = K|X = \mathbf{x})]^\top$, the posterior probability of a sample is given by:

$$p(C = i|X = \mathbf{x}) = \mathbf{r}_i^\top \boldsymbol{\psi}. \quad (2)$$

The *maximum a posteriori* rule is used to select its corresponding label.

If we suppose that the unsupervised structure is known, *i.e.*, the clusters have been identified by any clustering technique, the learning problem amounts to estimate the r_{ij} or, through a matrix form, $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_k]^\top$, $\mathbf{R} \in \mathbb{R}^{k \times K}$. Bouveyron and Girard (2009) have proposed to minimize the negative log-likelihood which, after some straightforward simplification, reduces to the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{R}} \quad & \sum_{\ell=1}^n -\ln(\boldsymbol{\beta}_\ell^\top \mathbf{R} \boldsymbol{\psi}_\ell) \\ \text{Constraint to} \quad & \mathbf{R}^\top \mathbf{1}_C = \mathbf{1}_K, \\ & \mathbf{R} \succcurlyeq 0, \end{aligned} \quad (3)$$

where $\mathbf{1}_C \in \mathbb{R}^C$ denotes the vector of ones, $\succcurlyeq$ denotes the element-wise non-negative constraint and $\boldsymbol{\beta}_\ell \in \mathbb{R}^k$ with $\beta_{\ell i} = 1$ if $i = c_\ell$, otherwise $\beta_{\ell i} = 0$.

In their original work, Bouveyron and Girard (2009) have used a standard constrained gradient descent to solve Eq. (3) (*fmincon* from Matlab[1]). The main drawback of such general solver is it neglects the convexity of this optimization problem, resulting in a very slow convergence. Here, we show that the problem is convex w.r.t. $\mathbf{R}$ and propose a fast algorithm to solve it, that scales well with the number of samples.

3. Global consensus with simplex constraint optimization

Bouveyron and Girard (2009) did not discuss about the convexity of the optimization problem. Hence, we shall prove its

[1] https://fr.mathworks.com/help/optim/ug/fmincon.html

convexity (next subsection). Then, a dedicated algorithm based on Alternating Direction Method of Multipliers (ADMM) is proposed.

3.1. Convexity of the minimization problem

The constraints in Eq. (3) are conventional non-negativity and sum-to-one constraints, and therefore are known to be convex (Boyd and Vandenberghe, 2004). The negative log-likelihood can be written as $\sum_{\ell=1}^{n} -\ln(\boldsymbol{\omega}_\ell^\top \mathbf{r}_v)$ where $\boldsymbol{\omega}_\ell = \mathrm{vec}\,(\boldsymbol{\psi}_\ell \boldsymbol{\beta}_\ell^\top)$, $\mathbf{r}_v = \mathrm{vec}\,(\mathbf{R})$ and vec is the vectorization operator of a matrix. Such formulation stems from the fact that $\boldsymbol{\beta}_\ell^\top \mathbf{R} \boldsymbol{\psi}_\ell = \mathrm{trace}(\boldsymbol{\psi}_\ell \boldsymbol{\beta}_\ell^\top \mathbf{R})$ (Magnus, 2010). The Hessian matrix is given by:

$$\frac{\partial^2}{\partial \mathbf{r}_v \partial \mathbf{r}_v^\top}\left(\sum_{\ell=1}^{n} -\ln(\boldsymbol{\omega}_\ell^\top \mathbf{r}_v)\right) = \sum_{\ell=1}^{n} \frac{\boldsymbol{\omega}_\ell \boldsymbol{\omega}_\ell^\top}{(\boldsymbol{\omega}_\ell^\top \mathbf{r}_v)^2},$$

which is the sum of semipositive definite matrices and consequently a semipositive definite matrix. Hence, the negative log-likelihood is convex w.r.t. $\mathbf{R}$. Therefore, the constraint optimization problem in Eq. (3) is convex.

3.2. Consensus approach - ADMM Solver

To efficiently solve Eq. (3), we rewrite the optimization as an equivalent consensus-based problem, but more suitable to large-scale computing. We define (i) one $\mathbf{R}$ per sample and (ii) impose a global shared variable with simplex constraints as a global consensus (Boyd, 2010, Chap. 7). The optimization problem can be subsequently written as:

$$\begin{aligned} \min_{\mathbf{R}_\ell, \mathbf{Z}} \quad & \sum_{\ell=1}^{n} f_\ell(\mathbf{R}_\ell) + g(\mathbf{Z}) \\ \text{Constraint to} \quad & \mathbf{R}_\ell - \mathbf{Z} = \mathbf{0}\ \forall \ell \in \{1, \ldots, n\}, \end{aligned} \tag{4}$$

where $f_\ell(\mathbf{R}_\ell) = -\ln(\boldsymbol{\beta}_\ell^\top \mathbf{R}_\ell \boldsymbol{\psi}_\ell)_+$ for $\ell \in \{1, \ldots, n\}$, $(\cdot)_+ = \max(0, \cdot)$ and $g(\mathbf{Z})$ is defined as the indicator function in the simplex $\mathcal{S}$:

$$g(\mathbf{Z}) = \begin{cases} 0 & \text{if } \mathbf{Z} \in \mathcal{S} = \{\mathbf{Z} | \mathbf{Z} \succcurlyeq 0, \mathbf{Z}^\top \mathbf{1}_C = \mathbf{1}_K\} \\ +\infty & \text{otherwise.} \end{cases} \tag{5}$$

The solutions of Eq. (3) and Eq. (4) are equivalent. Here, the idea is to separate the function to be minimized from the simplex constraint. ADMM is an efficient algorithm to solve a problem such as Eq. (4), assuming fast minimizations of elementary functions f_ℓ and g are available. Boyd (2010) provides a comprehensive review of ADMM algorithms, and in particular for consensus problems (Chap. 7).

Following the ADMM framework, the augmented Lagrangian is:

$$\begin{aligned} &\mathcal{L}_\rho(\mathbf{R}_1, \ldots, \mathbf{R}_n,\ \mathbf{Z}, \mathbf{Y}_1, \ldots, \mathbf{Y}_n) = \\ &\sum_{\ell=1}^{n} \left(f_\ell(\mathbf{R}_\ell) + \langle \mathbf{Y}_\ell, \mathbf{R}_\ell - \mathbf{Z} \rangle_F + \frac{\rho}{2} \|\mathbf{R}_\ell - \mathbf{Z}\|_F^2 \right) + g(\mathbf{Z}), \end{aligned} \tag{6}$$

where $\mathbf{Y}_\ell$ are the Lagrange multipliers, $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product between two real matrices, $\| \cdot \|_F$ is its associated norm and ρ is a penalty parameter. The resulting iterative algorithm is the following:

$$\begin{aligned} \mathbf{R}_\ell^{(t+1)} = \arg\min_{\mathbf{R}_\ell} \Bigg\{ & f_\ell(\mathbf{R}_\ell) + \langle \mathbf{Y}_\ell^{(t)}, \mathbf{R}_\ell - \mathbf{Z}^{(t)} \rangle_F \\ & + \frac{\rho^{(t)}}{2} \|\mathbf{R}_\ell - \mathbf{Z}^{(t)}\|_F^2 \Bigg\}, \end{aligned} \tag{7}$$

$$\begin{aligned} \mathbf{Z}^{(t+1)} = \arg\min_{\mathbf{Z}} \Bigg\{ & g(\mathbf{Z}) + \sum_{\ell=1}^{n} \Bigg(\langle \mathbf{Y}_\ell^{(t)}, \mathbf{R}_\ell - \mathbf{Z} \rangle_F \\ & + \frac{\rho^{(t)}}{2} \|\mathbf{R}_\ell^{(t+1)} - \mathbf{Z}\|_F^2 \Bigg) \Bigg\}, \end{aligned} \tag{8}$$

$$\mathbf{Y}_\ell^{(t+1)} = \mathbf{Y}_\ell^{(t)} + \rho^{(t)} (\mathbf{R}_\ell^{(t+1)} - \mathbf{Z}^{(t+1)}), \tag{9}$$

where (t) denotes the current iteration. The algorithm is efficient since each update step can be computed explicitly or rapidly, as shown in the following.

3.2.1. $\mathbf{R}_\ell$-update

Let $\mathcal{F}_\ell$ be the function to be minimized at iteration (t):

$$\begin{aligned} \mathcal{F}_\ell(\mathbf{R}_\ell) = & -\ln(\boldsymbol{\beta}_\ell^\top \mathbf{R}_\ell \boldsymbol{\psi}_\ell)_+ + \langle \mathbf{Y}_\ell^{(t)}, \mathbf{R}_\ell - \mathbf{Z}^{(t)} \rangle_F \\ & + \frac{\rho^{(t)}}{2} \|\mathbf{R}_\ell - \mathbf{Z}^{(t)}\|_F^2. \end{aligned} \tag{10}$$

We assume $\boldsymbol{\beta}_\ell^\top \mathbf{R}_\ell \boldsymbol{\psi}_\ell > 0$, otherwise $\mathcal{F}_\ell(\mathbf{R}_\ell) = +\infty$ and the corresponding $\mathbf{R}_\ell$ cannot be a minimizer of $\mathcal{F}_\ell$. Its derivative w.r.t. $\mathbf{R}_\ell$ is:

$$\nabla_{\mathbf{R}_\ell} \mathcal{F}_\ell(\mathbf{R}_\ell) = -\frac{\boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top}{\boldsymbol{\beta}_\ell^\top \mathbf{R}_\ell \boldsymbol{\psi}_\ell} + \mathbf{Y}_\ell^{(t)} + \rho^{(t)} (\mathbf{R}_\ell - \mathbf{Z}^{(t)}). \tag{11}$$

Let us note that $\boldsymbol{\beta}^\top \mathbf{R}_\ell \boldsymbol{\psi}_\ell = \langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \mathbf{R}_\ell \rangle_F$. We are looking for $\hat{\mathbf{R}}_\ell$ such that the gradient is canceled: such solution is also solution of $\langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \nabla_{\mathbf{R}_\ell} \mathcal{F}_\ell(\hat{\mathbf{R}}_\ell) \rangle_F = 0$. Injecting Eq. (11) in the previous Frobenius product and arranging the terms leads to:

$$\begin{aligned} & -\|\boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top\|_F^2 + \langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \hat{\mathbf{R}}_\ell \rangle_F \langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \mathbf{Y}_\ell^{(t)} - \rho^{(t)} \mathbf{Z}^{(t)} \rangle_F \\ & \qquad + \rho^{(t)} \langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \hat{\mathbf{R}}_\ell \rangle_F^2 = 0. \end{aligned} \tag{12}$$

Eq. (12) is a quadratic equation w.r.t. $\langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \hat{\mathbf{R}}_\ell \rangle_F$ whose positive root is:

$$\begin{aligned} \left\langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \hat{\mathbf{R}}_\ell \right\rangle_F = & \frac{1}{2} \left\langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \mathbf{Z}^{(t)} - \frac{\mathbf{Y}_\ell^{(t)}}{\rho^{(t)}} \right\rangle_F \\ & + \sqrt{\frac{1}{4} \left\langle \boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top, \mathbf{Z}^{(t)} - \frac{\mathbf{Y}_\ell^{(t)}}{\rho^{(t)}} \right\rangle_F^2 + \frac{\|\boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top\|_F^2}{\rho^{(t)}}} \\ \triangleq & \ \nu_\ell^{(t)} \end{aligned} \tag{13}$$

A negative root cannot be a solution of Eq. (10), as previously discussed. By plugging Eq. (13) into Eq. (11), an explicit solution for Eq. (7) is obtained:

$$\mathbf{R}_\ell^{(t+1)} = \mathbf{Z}^{(t)} + \frac{1}{\rho^{(t)}} \left(\frac{\boldsymbol{\beta}_\ell \boldsymbol{\psi}_\ell^\top}{\nu_\ell^{(t)}} - \mathbf{Y}_\ell^{(t)} \right). \tag{14}$$

The solution can be computed for all $\ell \in \{1, \ldots, n\}$ in parallel, allowing very fast computation for the $\mathbf{R}$-update step.

3.2.2. $\mathbf{Z}$-update and $\mathbf{Y}_\ell$-update

The $\mathbf{Z}$-update corresponds to the projection on the probability simplex. A simple iterative method is used in this work, based on bisection on a one dimensional function, as described in (Parikh, 2014).

The $\mathbf{Y}_\ell$-update is explicit and can be computed in parallel for all $\ell \in \{1, \ldots, n\}$.

3.2.3. Stopping criterion

Following the recommendation of Boyd (2010), the convergence of the algorithm is monitored using the norm of the primal and dual residuals, $\mathbf{Pr}^{(t)}$ and $\mathbf{Dr}^{(t)}$, respectively. These norms are computed as:

$$\|\mathbf{Pr}^{(t)}\|_F^2 = \sum_{\ell=1}^{n} \|\mathbf{R}_\ell^{(t)} - \mathbf{Z}^{(t)}\|_F^2, \quad (15)$$

$$\|\mathbf{Dr}^{(t)}\|_F^2 = n\rho^{(t)}\|\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\|_F^2. \quad (16)$$

The primal residual computes the average distance between each $\mathbf{R}_\ell$ and $\mathbf{Z}$. It is a measure of the consensus. The dual residual measures the changes in the shared variable between two iterations. Once both residuals are sufficiently small *i.e.*, below a specific threshold, the algorithm is stopped. We use the stopping criterion proposed in (Boyd, 2010, Chapter 3.3), for the primal and dual residuals, respectively:

$$\epsilon_{pr}^{(t)} = \sqrt{nkK}\,\epsilon_{abs} + \epsilon_{rel} \max\left[\begin{array}{c} \sqrt{\sum_{\ell=1}^{n} \|\mathbf{R}_\ell^{(t)}\|_F^2}\,, \\ \sqrt{n\rho^{(t)}}\|\mathbf{Z}^{(t)}\|_F \end{array} \right] \quad (17)$$

$$\epsilon_{dr}^{(t)} = \sqrt{nkK}\,\epsilon_{abs} + \epsilon_{rel}\|\mathbf{Y}^{(t)}\|_F, \quad (18)$$

where ϵ_{abs} and ϵ_{rel} are two hyperparameters to be tuned. Default values are 10^{-3} and 10^{-4}, respectively.

3.2.4. Update penalty parameter strategy

In practice, a suitable value for the penalty parameter can improve the convergence of the ADMM algorithm (Xu et al., 2017). In this work, we use the strategy proposed in (Boyd, 2010, Chapter 3.4.1) where ρ is updated after each iteration, based on:

$$\rho^{(t+1)} \hat{=} \begin{cases} \tau\rho^{(t)} & \text{if } \left\|\mathbf{Pr}^{(t)}\right\|_F > \mu \left\|\mathbf{Dr}^{(t)}\right\|_F \\ \rho^{(t)}/\tau & \text{if } \left\|\mathbf{Dr}^{(t)}\right\|_F > \mu \left\|\mathbf{Pr}^{(t)}\right\|_F \\ \rho^{(t)} & \text{otherwise,} \end{cases} \quad (19)$$

where μ and τ are hyperparameters set by the user. The rationale behind this strategy is to approximately balance the weight between primal and dual residuals. In all reported experiments, the values are 10.0 and 2.0, respectively.

4. Experimental results on synthetic data

4.1. Comparison with the standard solver

The proposed convex solver is compared with the original solver (Bouveyron and Girard, 2009). We have implemented the generic solver using the Sequential Least Squares Programming (SLSQP) solver provided by the Scipy library in Python (Virtanen and SciPy 1.0 Contributors, 2020) using the Numpy array programming facility (Harris et al., 2020).

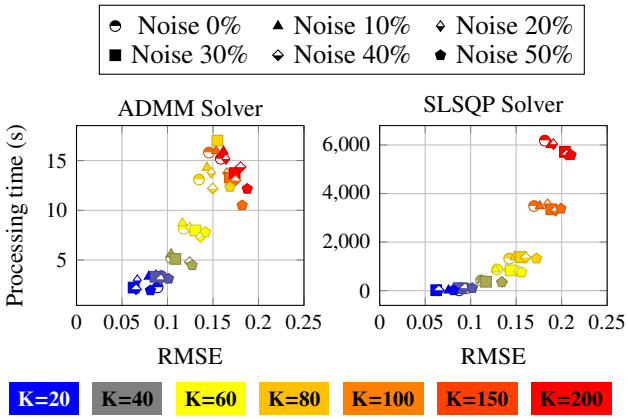


Fig. 2. Results on synthetic data. The colors are related to the numbers of clusters K used in the simulation and the marker shapes are related to the levels of label noise. The values averaged over 5 runs are reported for each configuration.

For the synthetic data set, the clusters are generated as isotropic Gaussian samples in $\mathbb{R}^2$. The number of clusters K was 20, 40, 60, 80, 100, 150, 200 for a fixed number of samples $n = 1{,}000$. Then, a sparse $\mathbf{R}$ matrix is generated to link the clusters to a set of $k = 10$ classes. The effectiveness of the algorithm is assessed using the root mean square error (RMSE) and the processing time over 5 runs. The RMSE is computed as

$$\text{RMSE}(\hat{\mathbf{R}}) = \sqrt{\frac{1}{k \times K} \sum_{i,\ j=1}^{k,\ K} (r_{ij} - \hat{r}_{ij})^2},$$

where r_{ij} and $\hat{r}_{ij}$ are the true and the estimated values, respectively. The results are reported in Figure 2.

From these simulated results, we can first conclude that the quality of the estimation of $\mathbf{R}$ is equivalent between the generic solver and our proposed algorithm whatever the number of clusters or the level of label noise. Hence, both algorithms converge to a very close solution.

At the same time, our algorithm is 5 to 400 times faster than the generic solver. As the number of clusters increases, the size of the optimization problem also increases (expressed as the number of parameters to estimate $k \times K$) and the generic solver takes increasingly time to converge: from approximately 10 seconds for $K = 20$ to 6,000 seconds for $K = 200$. Conversely, our approach only increases from 2 to 15 seconds, respectively.

Another finding is that a higher level of label noise will generally result in a slight decrease of the processing time w.r.t. noise-free configuration: in that case, the consensus is not reachable and the algorithm stops earlier.

4.2. Influence of the number of samples

Another set of simulation targets to assess the scalability of our algorithm w.r.t. the number of training samples. We perform similar simulations, fixing $K = 100$, the label noise to a level of 30%, and increasing the number of samples from 10^3 to

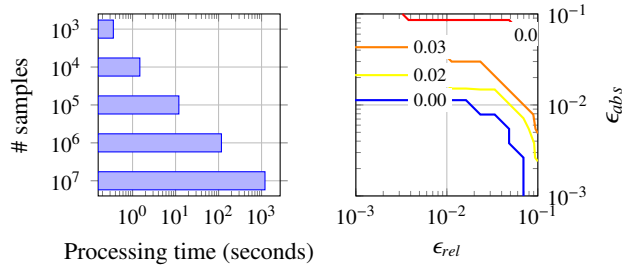


Fig. 3. Influence of the number of samples and of the hyperparameters. *Left*: Processing time as a function of the number of samples for $k = 10$, $K = 100$ and a level of label noise of 30% for the synthetic data set. *Right*: RMSE as a function of ϵ_{rel} and ϵ_{abs} (isolines are constant values).

10^7 with a logarithmic step size of 10. Figure 3 shows the processing time averaged over 5 runs[2]. It scales almost linearly. Even with 10^7 samples, our algorithm takes less than 20 minutes to converge.

4.3. Influence of the hyperparameters

Figure 3 shows the effect of the stopping criterion (defined in Section 3.2.3) on the convergence of the algorithm. For this simulation, 10 clusters, 4 classes, and 10,000 training samples are simulated with the same procedure without label noise. In that case, we can see that small enough ϵ_{abs} and ϵ_{rel} values are necessary to reach a good fit, *i.e.*, a small RMSE.

Since the stopping criteria are normalized w.r.t. the size of the problem (*i.e.*, number of samples, clusters and classes), a general comment would be to use smaller default values than those proposed in Section 3.2.3. However, it is known that ADMM converges quickly to a rough solution, and spends many iterations to converge to the optimal solution (Boyd, 2010): in practice, too small values will unnecessarily increase the processing time. For all our experiments, we found that the values proposed in Section 3.2.3 work well.

We can conclude that our proposed solver is faster by several orders of magnitude than the original solver (Bouveyron and Girard, 2009), while converging to the true solution. In the next section, the robust discriminant model is confronted with several real datasets and competitive algorithms.

5. Results on real data

5.1. Hand-written digits

UCI ML Hand-written digits data set is a challenging handwritten text recognition dataset (Pedregosa et al., 2011). It is composed of 1,797 samples with 64 features (one sample corresponds to an image of 8×8 pixels and was arranged into a feature vector of 64 features) and 10 classes. Half of the data have been used for the training step and the remaining for the validation step. The robust mixture discriminant analysis (RMDA) has been compared with a Quadratic Discriminant Analysis (QDA) and a Random Forest (RF) classifiers, from the Scikit-learn library (Pedregosa et al., 2011). QDA is chosen because it can be considered as an extreme case of RMDA with one cluster per class. RF is adopted for its robustness to label noise (Pelletier et al., 2017). QDA is used with a regularization parameter automatically chosen by cross-validation and 100 trees compose the RF. For RMDA, a conventional Gaussian Mixture Model (GMM) is used to extract the clusters. Several sizes of clusters are investigated: $K \in \{20, 30, \ldots, 150\}$. Label noise is introduced in the data and all experiments are run 20 times, with a random generation of training/validation sets and uniform label noise for each run. The overall accuracy (OA) is used to assess the classification accuracy.

Table 1. Hand-written digits classification results. The number indicates the averaged overall accuracy over the 20 runs. Subscript numbers for RMDA correspond to the number of clusters. The rows correspond to different levels of label noise: 0.1 indicates that the probability of label switch is 0.1 (i.e., 10% of the labels are wrong). Best results for each level of label noise are reported in boldface, and numbers is brackets are the variance of the overall accuracy.

	0.0	0.1	0.2	0.3	0.4	0.5
RF	96.7 (0.6)	96.4 (0.6)	95.5 (0.9)	94.1 (0.9)	91.3 (1.0)	86.2 (1.9)
QDA	97.0 (0.7)	94.9 (1.6)	89.3 (1.7)	88.6 (1.0)	87.9 (1.2)	86.9 (1.3)
$RMDA_{20}$	88.1 (0.9)	87.9 (1.2)	87.9 (1.1)	87.9 (1.1)	87.8 (1.2)	87.5 (1.2)
$RMDA_{30}$	91.7 (0.6)	91.7 (0.7)	91.7 (0.7)	91.5 (0.9)	91.3 (0.8)	91.0 (1.2)
$RMDA_{40}$	94.3 (0.6)	94.3 (0.6)	94.2 (0.7)	93.9 (1.0)	93.6 (1.1)	**92.5 (1.7)**
$RMDA_{50}$	94.9 (0.7)	94.9 (0.8)	94.7 (0.9)	94.6 (0.9)	93.7 (1.0)	92.0 (1.7)
$RMDA_{60}$	94.3 (0.5)	94.3 (0.5)	94.2 (0.6)	93.6 (1.1)	93.1 (1.4)	90.8 (2.5)
$RMDA_{70}$	96.1 (0.4)	96.1 (0.4)	95.9 (0.5)	**95.6 (0.8)**	**94.3 (1.5)**	91.2 (2.0)
$RMDA_{80}$	96.2 (0.4)	96.0 (0.6)	95.7 (0.6)	94.8 (1.4)	93.5 (1.6)	88.6 (2.2)
$RMDA_{90}$	96.6 (0.6)	96.3 (0.7)	95.8 (1.2)	94.9 (1.3)	92.4 (2.5)	88.0 (2.5)
$RMDA_{100}$	96.9 (0.6)	96.6 (0.9)	**96.0 (1.1)**	95.0 (1.7)	93.0 (1.9)	86.1 (2.8)
$RMDA_{110}$	96.6 (0.7)	96.3 (0.7)	95.6 (1.1)	93.6 (1.3)	90.2 (1.8)	84.1 (2.6)
$RMDA_{120}$	97.0 (0.5)	96.4 (0.6)	95.4 (0.9)	93.0 (1.8)	89.4 (2.2)	82.3 (3.5)
$RMDA_{130}$	**97.7 (0.6)**	**97.0 (0.9)**	96.1 (1.1)	93.7 (1.7)	89.5 (2.1)	81.6 (3.1)
$RMDA_{140}$	97.4 (0.7)	96.5 (1.0)	95.2 (1.6)	92.7 (1.9)	88.4 (2.1)	79.4 (2.8)
$RMDA_{150}$	96.7 (0.7)	95.7 (0.9)	93.4 (2.1)	90.0 (3.1)	85.5 (2.5)	77.3 (3.3)

The average OA is reported in Table 1. When the label noise is null, better RMDA results are obtained for a larger number of clusters. However, as the level of label noise increases, the performance in terms of OA decreases for a larger number of clusters. RMDA used with a smaller number of clusters appears more robust w.r.t. label noise. In particular, RMDA with 20 or 30 clusters is slightly affected by the label noise.
QDA performs the worst and is highly label noise dependent. RF performs equally than RMDA for a level of label noise lower than 0.4. For higher cases, the performance significantly drops and RF produces worse OA than RMDA used with 30 to 80 clusters.

A by-product obtained by RMDA is the cluster assignment to each class. From the **R** matrix, it is possible to associate a cluster to one (or more) class. Figure 4 shows the **R** obtained for RMDA coupled with GMM and 30 clusters with no label noise. The sum-to-one and positivity constraints result in a sparse matrix: most of the r_{ij} are null. For instance, only one cluster was assigned to the class "0". In general, a cluster belongs to one or two classes. Once the clusters have been identified for a given class, it is possible to recover their mean values from the GMM model, as shown in Figure 5. For this class, we can see that clusters correspond to various distortion (*e.g.*, translation, dilation ...) of the digit "2". It shows that RMDA properly train is able to learn classes subject to distortion.

[2]Similar simulations with the generic solver took too much times and are not reported.

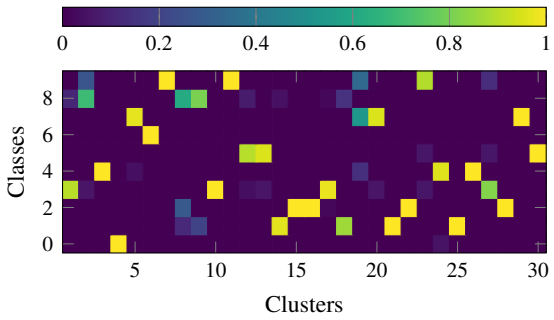


Fig. 4. R matrix for $RMDA_{30}$ for Hand-written digits data set (one cell=one r_{ij} value).

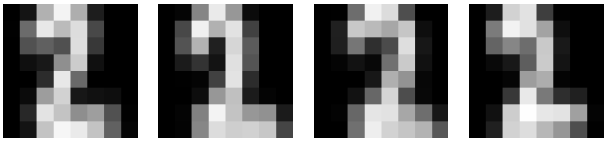


Fig. 5. Mean value for clusters belonging to Hand-written digits class "2" (the feature vector has been reshaped as an image).

5.2. CovType

The second dataset is the *Covtype* dataset, which contains 581,012 samples with 54 features representing cartographic variables (`https://archive.ics.uci.edu/ml/datasets/covertype`). Seven classes of cover types are defined with a highly unbalanced class proportion: the samples of two classes represent 84% of the data. The experimental set-up is the same than for Hand-written digits, except for the number of clusters. Since the number of samples is higher, a larger number of clusters is investigated, up to 300. The averaged OA are reported in Table 2. We only report the results for some K because results are similar for various configurations.

For this data set, RMDA performs the worst, independently of the number of clusters used. Surprisingly, the label noise does not affect the performances of the QDA and RMDA, while RF is affected. Some variables of the data are categorical and break the Gaussian assumption of QDA and GMM used with RMDA, while RF is not affected.

This data set illustrates the dependence of the RMDA model on a good initial clustering step: some of the classes are not represented during the clustering and hence are not recovered by the supervised step. We have investigated several conventional Bayesian criteria for unsupervised models based on the likelihood of the clustering (*e.g.*, AIC, BIC) but they do not provide meaningful results: a plateau is reached, but it is not correlated with the final classification accuracy.

5.3. Satellite Time Series

This third data set called *normalized difference vegetation index* (NDVI) is composed of simulated optical satellite acquisitions (Pelletier et al., 2017), representing the reflectance of agricultural vegetation over one year. 13,000 samples with 15 features are available with 13 classes. The experimental set-up is the same than for Hand-written digits dataset (Section 5.1). The averaged OA are reported in Table 3. We only present the results for $K \in \{20, \ldots, 50\}$ because results for a larger number of clusters are similar to those obtained with $K = 50$.

Table 2. Covtype classification results. Best results for each level of label noise are reported in boldface, and numbers is brackets are the variance of the overall accuracy.

	0.0	0.1	0.2	0.3	0.4	0.5
RF	**94.4 (0.1)**	**94.1 (0.0)**	**93.0 (0.1)**	**90.7 (0.1)**	**86.5 (0.1)**	**79.1 (0.1)**
QDA	68.3 (0.1)	68.8 (0.1)	69.0 (0.1)	69.0 (0.2)	68.9 (0.2)	68.7 (0.3)
$RMDA_{20}$	60.0 (0.1)	60.0 (0.1)	60.0 (0.1)	60.0 (0.1)	60.0 (0.1)	60.0 (0.1)
$RMDA_{30}$	62.3 (0.0)	62.3 (0.0)	62.3 (0.0)	62.3 (0.0)	62.3 (0.1)	62.3 (0.1)
$RMDA_{40}$	62.3 (0.1)	62.3 (0.1)	62.3 (0.0)	62.3 (0.0)	62.3 (0.0)	62.3 (0.0)
$RMDA_{50}$	63.9 (0.1)	63.9 (0.1)	63.9 (0.1)	63.9 (0.1)	63.9 (0.1)	63.9 (0.1)
$RMDA_{60}$	62.0 (0.1)	62.0 (0.1)	62.0 (0.1)	62.0 (0.1)	61.9 (0.1)	61.9 (0.1)
$RMDA_{70}$	64.8 (0.0)	64.8 (0.0)	64.8 (0.0)	64.8 (0.0)	64.8 (0.0)	64.8 (0.0)
$RMDA_{80}$	64.6 (0.0)	64.6 (0.1)	64.6 (0.1)	64.6 (0.1)	64.6 (0.0)	64.6 (0.1)
$RMDA_{90}$	64.4 (0.1)	64.4 (0.1)	64.4 (0.1)	64.4 (0.1)	64.4 (0.1)	64.4 (0.1)
$RMDA_{100}$	64.9 (0.0)	64.9 (0.0)	64.9 (0.0)	64.9 (0.0)	64.9 (0.0)	64.9 (0.0)
$RMDA_{120}$	65.0 (0.1)	65.0 (0.1)	65.0 (0.1)	65.0 (0.1)	65.0 (0.1)	65.0 (0.1)
$RMDA_{140}$	65.9 (0.1)	66.0 (0.1)	66.0 (0.1)	65.9 (0.1)	65.9 (0.1)	65.9 (0.1)
$RMDA_{160}$	66.3 (0.1)	66.3 (0.1)	66.3 (0.1)	66.3 (0.1)	66.3 (0.1)	66.3 (0.1)
$RMDA_{200}$	66.5 (0.1)	66.5 (0.1)	66.5 (0.1)	66.5 (0.1)	66.5 (0.1)	66.5 (0.1)
$RMDA_{225}$	66.4 (0.1)	66.4 (0.1)	66.4 (0.1)	66.4 (0.1)	66.3 (0.1)	66.3 (0.1)
$RMDA_{250}$	67.4 (0.1)	67.4 (0.1)	67.4 (0.1)	67.4 (0.1)	67.4 (0.1)	67.4 (0.1)
$RMDA_{275}$	67.3 (0.1)	67.3 (0.1)	67.3 (0.1)	67.3 (0.1)	67.3 (0.1)	67.2 (0.1)
$RMDA_{300}$	67.6 (0.1)	67.5 (0.1)	67.5 (0.1)	67.5 (0.1)	67.5 (0.1)	67.5 (0.1)

Table 3. NDVI classification results. Best results for each level of label noise are reported in boldface, and numbers is brackets are the variance of the overall accuracy.

	0.0	0.1	0.2	0.3	0.4	0.5
RF	87.1 (0.2)	86.8 (0.2)	86.6 (0.2)	86.5 (0.3)	86.0 (0.4)	84.9 (0.5)
QDA	**89.8 (0.0)**	**87.9 (0.3)**	86.1 (0.6)	84.7 (1.0)	83.4 (1.1)	82.0 (1.3)
$RMDA_{20}$	82.7 (0.0)	82.7 (0.2)	82.7 (0.2)	82.7 (0.3)	82.8 (0.7)	82.8 (0.8)
$RMDA_{30}$	87.3 (0.0)	87.3 (0.2)	**87.1 (0.4)**	**87.1 (0.4)**	**86.9 (0.5)**	**86.9 (0.7)**
$RMDA_{40}$	85.4 (0.0)	85.4 (0.1)	85.4 (0.1)	85.4 (0.1)	85.5 (0.3)	85.3 (0.5)
$RMDA_{50}$	85.2 (0.0)	85.3 (0.1)	85.3 (0.3)	85.3 (0.3)	85.2 (0.3)	85.0 (0.5)

Similar comments for the previous datasets can be drawn. Too small or too large numbers of clusters provide the worst results for RMDA. Yet the algorithm is robust to label noise and provides the best results for a level of label noise greater than 0.2. For NDVI, QDA provides the best results when the level of label noise is limited, but it is significantly affected when the level increases.

Figure 6 shows the mean temporal profile of one class (*sunflower*) and its associated clusters obtained from RMDA. These by-products provide richer information of the underlying process than only class labels: for instance, different trends are observed for the cluster profiles in the second part of the years. It indicates in that case different farmer practices associated to different vegetation regrowth for the second semester of the year, whereas the first part of the year is similar for the three clusters.

6. Discussion and conclusions

A fast algorithm to optimize the Robust Mixture of Discriminant Analysis parameters has been proposed in this letter. It is based on ADMM and exploits the convexity of the loss function. Experimental results show a speed-up of several orders of magnitude w.r.t. the initially proposed solver while providing similar estimation accuracy. Comparisons with standard classifiers on three real-life datasets, with various levels of label noise, confirm the potential of the method.

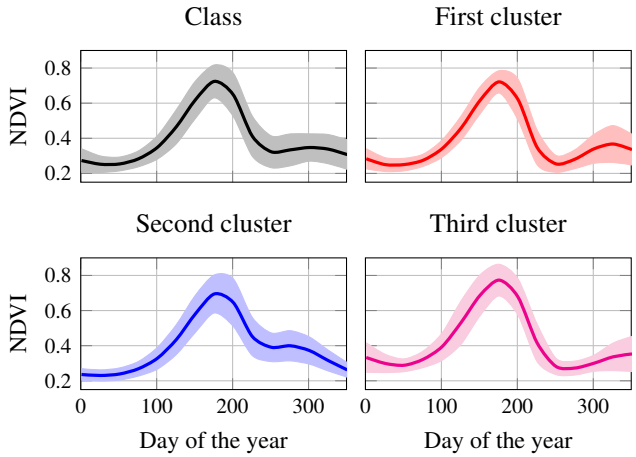


Fig. 6. Mean profile for the *sunflower* class (NDVI dataset) and its associated cluster means from RMDA (thick continuous line). The shade regions correspond to the mean ± the standard deviation.

However, as shown in the experiments, the RMDA is sensitive to the unsupervised modelling provided by the clustering algorithms. For simulations in sections 5.1 and 5.3, the clustering was good enough to reach good classification accuracy, while in section 5.2 it was too coarse to classify correctly the data set. Hence, supervised techniques, such as cross-validation, should be used to select an appropriated clustering, albeit time consuming. In the results reported here, we have applied a brute-force trial-and-error search: several number of clusters values were investigated and the best ones were kept. It may not be tractable for very large scale scenarios.

Current researches are oriented towards differentiable clustering algorithms that can be integrated in the RMDA loss. A full optimization of the model parameters could be performed: the number of clusters and the parameters of the clusters would be the parameters of RMDA, to be jointly learned with the **R** matrix. Deep neural networks for clustering are a natural candidate for such investigations (Moradi Fard et al., 2020).

Acknowledgments

The authors are supported by the French National Research Agency under the grant ANR-18-CE23-0023 and the ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANITI) under grant agreement ANITI ANR-19-PI3A-0004.

References

Algan, G., Ulusoy, İ., 2020. Label noise types and their effects on deep learning. CoRR URL: http://arxiv.org/abs/2003.10471v1.

de Aquino Afonso, B.K., Berton, L., 2020. Analysis of label noise in graph-based semi-supervised learning, in: Hung, C., Cerný, T., Shin, D., Bechini, A. (Eds.), SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, pp. 1127–1134. doi:10.1145/3341105.3374013.

Bouveyron, C., Girard, S., 2009. Robust supervised classification with mixture models: Learning from data with uncertain labels. Pattern Recognition 42, 2649–2658. doi:10.1016/j.patcog.2009.03.027.

Boyd, S., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning 3, 1–122. doi:10.1561/2200000016.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press, New York, NY, USA.

Chapelle, O., Schölkopf, B., Zien, A., 2006. Semi-Supervised Learning (Adaptive Computation and Machine Learning). The MIT Press.

Ding, Y., Wang, L., Fan, D., Gong, B., 2018. A semi-supervised two-stage approach to learning from noisy labels, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1215–1224. doi:10.1109/WACV.2018.00138.

Frénay, B., Verleysen, M., 2014. Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems 25, 845–869. doi:10.1109/tnnls.2013.2292894.

Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I.W., Kwok, J.T., Sugiyama, M., 2020. A survey of label-noise representation learning: Past, present and future. CoRR abs/2011.04406. URL: https://arxiv.org/abs/2011.04406.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. doi:10.1038/s41586-020-2649-2.

Kong, K., Lee, J., Kwak, Y., Kang, M., Kim, S.G., Song, W., 2019. Recycling: Semi-supervised learning with noisy labels in deep neural networks. IEEE Access 7, 66998–67005. doi:10.1109/ACCESS.2019.2918794.

Lee, K.H., He, X., Zhang, L., Yang, L., 2017. Cleannet: Transfer learning for scalable image classifier training with label noise. CoRR URL: http://arxiv.org/abs/1711.07131v2.

Li, J., Socher, R., Hoi, S.C., 2020. Dividemix: Learning with noisy labels as semi-supervised learning, in: ICLR.

Magnus, J.R., 2010. On the concept of matrix derivative. Journal of Multivariate Analysis 101, 2200–2206. doi:10.1016/j.jmva.2010.05.005.

Moradi Fard, M., Thonet, T., Gaussier, E., 2020. Deep k-means: Jointly clustering with k-means and learning representations. Pattern Recognition Letters 138, 185–192. doi:10.1016/j.patrec.2020.07.028.

Parikh, N., 2014. Proximal algorithms. Foundations and Trends® in Optimization 1, 127–239. URL: https://doi.org/10.1561/2400000003, doi:10.1561/2400000003.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Édouard Duchesnay, 2011. Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12, 2825–2830.

Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., Dedieu, G., 2017. Effect of training class label noise on classification performances for land cover mapping with satellite image time series. Remote Sensing 9, 173. doi:10.3390/rs9020173.

Shen, Y., Sanghavi, S., 2019. Learning with bad training data via iterative trimmed loss minimization, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), ICML, pp. 5739–5748. URL: http://proceedings.mlr.press/v97/shen19e.html.

Song, H., Kim, M., Lee, J.G., 2019. SELFIE: Refurbishing unclean samples for robust deep learning, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, pp. 5907–5915. URL: http://proceedings.mlr.press/v97/song19b.html.

Song, H., Kim, M., Park, D., Lee, J.G., 2020. Learning from noisy labels with deep neural networks: a survey. CoRR URL: http://arxiv.org/abs/2007.08199v3.

Virtanen, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272. doi:10.1038/s41592-019-0686-2.

Xu, Z., Figueiredo, M., Goldstein, T., 2017. Adaptive ADMM with Spectral Penalty Parameter Selection, in: Singh, A., Zhu, J. (Eds.), AISTATS, pp. 718–727. URL: http://proceedings.mlr.press/v54/xu17a.html.

Yan, Y., Xu, Z., Tsang, I., Long, G., Yang, Y., 2016. Robust semi-supervised learning through label aggregation, in: AAAI.

Yao, J., Wang, J., Tsang, I.W., Zhang, Y., Sun, J., Zhang, C., Zhang, R., 2019. Deep learning from noisy image labels with quality embedding. IEEE Transactions on Image Processing 28, 1909–1922. doi:10.1109/TIP.2018.2877939.

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. CoRR URL: http://arxiv.org/abs/1611.03530v2.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: