A regularized logistic regression based model for supervised learning

Carlos Brito-Pacheco, Carlos Brito-Loeza and Anabel Martin-Gonzalez

Abstract
In this work, we introduce a new regularized logistic model for the supervised classification problem. Current logistic models have become the preferred tools for supervised classification in many situations. They mostly use either $L_1$ or $L_2$ regularization of the weight vector of parameters. Here we take a different approach by applying regularization not to the weight vector but to the gradient vector of the function representing the separating hyper-surface. We present the mathematical analysis of the model in its continuous setting and provide experimental evidence to show that the new model is competitive with state of the art models.

Keywords
Supervised learning, variational methods, regularization

Received 4 December 2019; revised 9 August 2020; accepted 16 October 2020

Introduction
In the last decades, many supervised learning methods have been developed and the ideas behind those models, reviewed consistently in literature. Among one of the most complete reviews is the one from Singh et al., which tests many different methods on a single dataset and presents the results as a list of advantages and disadvantages of every tested method establishing their application areas. A different study, but also very complete, is the one from Fernández-Delgado et al. where 179 classifiers were evaluated on 121 datasets. It is difficult to take one side and name the best performing supervised learning algorithm since usually this depends on many factors. The work of Kotsiantis et al. brings some light in this direction. In this work the authors conducted a comprehensive review of different classification methods. The results showed that Support Vector Machines (SVM) and Neural Networks (NN) performed the best in terms of accuracy, classification speed and tolerance to parity problems. On the other hand, the very same algorithms presented lack of speed of learning, may have overfitting problems and there is the issue of selecting the best model parameters.

In this work we propose a new supervised binary classification method based in techniques from calculus of variations and functional analysis. By doing so, we are able to exploit the underlying mathematical theory and properties from each of these fields. Further, this allows us to obtain meaningful conditions in the continuous sense rather than just performing optimization discretely. By comparing the performance of our model against classical logistic regression models, SVM, and NN, we will show that the new method is competitive.

This work is organized as follows: first, we introduce the supervised learning problem and the new variational model. Then, we present the mathematical analysis of the model proving existence and uniqueness of the minimizer. Later, we present the numerical solution of the model by its first order optimality condition and the Levenberg-Marquardt method. Further, we present some results and discuss them before stating our conclusions. Finally, in Appendix 1 we present the proof of the Euler Lagrange partial differential equation.
A variational model for supervised learning

The problem

Given a training dataset of observations

\[ T = \{ (x_1, y_1), \ldots, (x_N, y_N) \mid x_i \in \Omega \subset \mathbb{R}^d, y_i \in \mathcal{Y} \} \]  

the supervised learning problem is to find a function \( u = u(x) \) defined by \( u : \Omega \subset \mathbb{R}^d \rightarrow \mathcal{Y} \) which is a good estimator of the value of \( y \) for a new \( x \) not in the training set.

The most popular framework is the minimization of the empirical risk

\[ R(u) = \frac{1}{N} \sum_{i=1}^{N} L(u(x_i), y_i) \]  

where \( L(u, y) \) is the loss function and \( S(u) \) is a regularizer. For the continuous case, however, we have:

\[ \min_u \int_{\Omega} L(u(y), y) \, d\Omega + \lambda S(u) \]  

The basic idea behind this variational model is to penalize wrong predictions through the loss function while allowing some relaxation through the regularizer. This is, the regularizer allows to control the smoothness of \( u \) representing the separating hypersurface. The regularizer also helps by making the problem well-posed and preventing overfitting. We will use a Lebesgue integrable function \( R(u) > 0 \) to somehow measure the complexity of \( u \). The variational model (3) is then transformed into

\[ \min_u F[u] \equiv \int_{\Omega} L(u(y), y) \, d\Omega + \lambda \int_{\Omega} R(u) \, d\Omega \]  

A probabilistic interpretation of (4) is to notice that the empirical risk of choosing \( u \) among a space of hypothesis functions \( \mathcal{H} \) is defined as the expectation of the loss function \( E[L(u(y))] = \int_{\Omega} L(u(y)) \, d\Omega \). Therefore, minimizing the regularized empirical risk, amounts to solving the following minimization problem:

\[ u^* = \arg \min_{u \in \mathcal{H}} F[u] \]  

Radial basis function approximation

Finding \( u^* \) in (5) involves solving numerically a partial differential equation (PDE). Due to the nature of the problem, it is not possible to use traditional methods such as the finite difference method. Therefore, we turn to using a collocation method to reduce the problem to a set of algebraic equations. To this end, we choose a set of radial basis functions (RBF) to approximate \( u(x) \) as a weighted sum of functions \( \phi_i(x) \) and weights \( w_i \) for \( i = 1, \ldots, N \). There are many possible choices for \( \phi_i(x) \). However, the Gaussian RBF kernel is probably the most known and widely used and therefore the one we chose to use all along this work. Thus, we write

\[ u(x) = \sum_{i=1}^{N} w_i \phi_i(x_i) \]  

and

\[ \phi_i(x) = e^{-c \|x-x_i\|^2} \]

where \( c > 0 \) is a constant that we are free to choose and \( \| \cdot \| \) the Euclidean norm. We will call \( c \) the fitting degree of our model.

Notice that in (7), the centers of the RBFs are the data observations \( x_i \). Due to this, the model will make predictions based on the Euclidean distance between points which class is known, and the new input for which its class is unknown while assigning a weight to each point. Therefore, some points will be more important than others when making the predictions. A similar idea to the one used in SVM.

Cross entropy as fitting term

A correct selection of the loss function in (4) is vital for the performance of the model. Here, since the model will be used for binary classification, we take a Bernoulli variable \( Y \in \{0, 1\} \). Generalization to multi-class is straightforward.

The criteria to select an appropriate loss function is that it must maximize the likelihood that, given parameters \( w_{ij} \), the model results in a prediction of the correct class for each input sample and this likelihood should be a function of these parameters. This can be achieved by the minimization of the cross-entropy (also called negative log-likelihood) that effectively maximizes the likelihood. This function is given by:

\[ L(u, y) = -y \ln \sigma(u) - (1 - y) \ln (1 - \sigma(u)) \]  

where \( \sigma(u) = e^u / (1 + e^u) \) is the sigmoid function.

\( L_2 \) function regularization

With the loss function for binary classification already chosen, we now proceed to select a regularizer. Most machine learning models mainly use parameter
regularization by regularizing the weight vector \( w = (w_1, \ldots, w_N) \) instead of \( u \). This is the case of the regularizer

\[
S(w) = \int_{\Omega} ||w||^2 \, d\Omega
\]

(9)

also known as ridge regression or Tikhonov regularization. This type of regularization helps the algorithm to distinguish between those inputs with high variance from the ones with small variance.

There is also the case of

\[
S(w) = \int_{\Omega} ||w|| \, d\Omega
\]

(10)

known as LASSO (Least Absolute Shrinkage and Selection Operator) regression. The key difference here is that LASSO shrinks the less important feature’s coefficient to zero thus, removing some features altogether.

More recently, Belkin et al. proposed that under the assumption that the support of the marginal distribution of the dataset is a compact manifold embedded in \( \mathbb{R}^N \) then a suitable regularizer is given by

\[
S(u) = \frac{1}{2} \int_{\Omega} ||\nabla u||^2 \, d\Omega
\]

(11)

Lin et al. took this idea further and explored with success the use of the Total Variation and Euler elastica as regularizers.

We argue that this way of directly regularizing the separating hypersurface \( u \) is better than regularizing the vector of parameters \( w \). Our argument is that while penalizing \( w \) has the objective of controlling the values \( w_i \) individually which in some cases may growth indiscriminately, penalizing \( u \) or its gradient as in (11) has the intention of controlling the smoothness of the separating hypersurface. The second, has the immediate impact of reducing overfitting.

The model

Let \( u : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R} \) be the classification function and let \( y \in \{0, 1\} \) be the target values. Let \( \sigma(u) \) to represent the probability that an input \( u(x) \) is classified as true or positive i.e. \( y = 1 \). Let

\[
F[u] = -\int_{\Omega} (y \ln \sigma(u) + (1 - y) \ln (1 - \sigma(u))) \, d\Omega + \frac{\lambda}{2} \int_{\Omega} ||\nabla u||^2 \, d\Omega
\]

(12)

Then our supervised learning models amounts to solving (5) with \( F[u] \) defined above.

Analysis of the model

Existence and uniqueness of a minimizer

In this section, we shall determine existence of a minimizer of our model. Our proof of existence consists in showing that some minimizing sequence \( \{u_k\}_{k \geq 0} \) actually does converge to a minimizer. We will show that the minimizing sequence is bounded and therefore there exists a subsequence which is convergent, given the right hypothesis.

**Theorem 3.1 (Existence of a Minimizer).** Let \( \Omega \subset \mathbb{R}^N \) be a bounded open set with a \( C^1 \) boundary and \( \lambda > 0 \) constant. Furthermore, assume that \( L : \Omega \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) is a non-negative convex function that is Lipschitz continuous. Then the functional

\[
F[u] = \int_{\Omega} \left( L(u, y) + \frac{\lambda}{2} ||\nabla u||^2 \right) \, d\Omega
\]

has at least one minimizer \( u \in X \), where

\[
X = \{ u \in W^{1,2}(\Omega) \mid u = v \text{ on } \partial\Omega \text{ in the sense of a trace} \}
\]

for \( v \in W^{1,2} \).

**Proof.** Assume \( m = \inf_{u} F[u] \) is finite. If it were not, we could just pick any \( u \) and we would be done. Let \( \{u_k\}_{k \geq 0} \) be a minimizing sequence such that \( u_k \in X \). By the definition of \( X \), we know that for any fixed \( \tilde{u} \in X \) then \( u_k - \tilde{u} \in W^{1,2}_0 \). Applying the Poincaré inequality yields

\[
||u_k||_{W^{1,2}(\Omega)} - ||\tilde{u}||_{W^{1,2}(\Omega)} \leq ||u_k - \tilde{u}||_{W^{1,2}(\Omega)} \\
\leq c ||\nabla u_k - \nabla \tilde{u}||_{L^2(\Omega)} \\
\leq c ||\nabla u_k||_{L^2(\Omega)} + c ||\nabla \tilde{u}||_{L^2(\Omega)}
\]

where \( c \) is some positive constant. We know that

\[
F[u] = \int_{\Omega} L(u, y) \, d\Omega + c ||\nabla u||^2_{L^2(\Omega)} \geq c ||\nabla u||_{L^2(\Omega)}^2
\]

and so, we can derive

\[
||u_k||_{W^{1,2}(\Omega)} \leq c ||\nabla u_k||_{L^2(\Omega)} + \beta \leq F[u_k]
\]

for constant \( \beta = ||\tilde{u}||_{W^{1,2}(\Omega)} + c ||\nabla \tilde{u}||_{L^2(\Omega)} \).
Since \( u_k \) is a minimizing sequence for \( m \) finite, we have that \( ||u_k||_{W^{1,2}(\Omega)} \) is bounded. Consequently there must exist a subsequence \( \{u_{k_i}\}_{k_i \geq 0} \) such that \( u_{k_i} \to \bar{u} \) in \( W^{1,2}(\Omega) \), for some \( \bar{u} \), in the sense that \( u_{k_i} \to \bar{u} \) and \( \nabla u_{k_i} \to \nabla \bar{u} \) in \( L^2(\Omega) \).

To show that \( F[u_k] \to F[\bar{u}] \), it is enough to show that \( F[u] \) is lower semicontinuous to deduce \( F[\bar{u}] \leq m \). To this end, we must prove two things individually:

\[
\lim \inf_{k \to \infty} \int_\Omega |\nabla u_k| \, d\Omega \geq \int_\Omega |\nabla \bar{u}| \, d\Omega \tag{13}
\]

and

\[
\lim_{k \to \infty} \int_\Omega L(u_k, y) \, d\Omega \geq \int_\Omega L(\bar{u}, y) \, d\Omega \tag{14}
\]

This is straightforward using the convexity of the functionals. In other words, we know that for some \( \varphi \in W^{1,2} \) and \( \nu \in L^2 \), the following is true:

\[
|\nabla u_k| \geq |\nabla \bar{u}| + 2(\nu, \nabla u_k - \nabla \bar{u})_{L^2} \tag{15}
\]

and

\[
L(u_k) \geq L(\bar{u}) + \langle \varphi, u_k - \bar{u} \rangle_{W^{1,2}} \tag{16}
\]

By taking the limit \( k \to \infty \) on both sides of (15) and (16) and since \( u_k \to \bar{u} \) in \( W^{1,2} \) and \( \nabla u_k \to \nabla \bar{u} \) in \( L^2(\Omega) \), it follows immediately that

\[
\lim_{k \to \infty} \int_\Omega \langle \nu, \nabla u_k - \nabla \bar{u} \rangle_{L^2} \, d\Omega = 0 \tag{17}
\]

\[
\lim_{k \to \infty} \int_\Omega \langle \varphi, u_k - \bar{u} \rangle_{W^{1,2}} \, d\Omega = 0 \tag{18}
\]

and therefore (13) and (14) are satisfied. To end the proof, note that since \( \{u_{k_i}\}_{k_i \geq 0} \) was a minimizing sequence, \( F[u_{k_i}] \to m \) and \( F[u] \) is lower semicontinuous, then we must have \( F[\bar{u}] = m \). \( \square \)

This completes our proof for the existence of a minimizer. We will end this section by stating a theorem about the uniqueness of the minimizer which follows by the convexity of the functional. The proof is elementary and therefore will not be presented.

**Theorem 3.2 (Uniqueness of Minimizer).** If \( L \) is strictly convex, then the functional \( F[u] \) in Theorem 3.1 has a unique solution.

**The solution**

In order to solve the expression in (4), we proceed to compute the functional derivative and equate to zero i.e. the first order optimality condition

\[
\frac{\delta F[u]}{\delta u} = 0 \tag{19}
\]

To this end, let \( R = R(u, \nabla u) \) and \( L(u) = L(u, y) \) with \( y \) fixed. Then, the functional derivative in (19) yields the Euler Lagrange equation:

\[
\frac{\delta F}{\delta u} = \frac{\partial f}{\partial u} - \nabla \cdot \frac{\partial f}{\partial \nabla u} = \left[ \frac{\partial L}{\partial u} + \lambda \frac{\partial R}{\partial u} \right] - \nabla \cdot \left[ \frac{\partial L}{\partial \nabla u} + \lambda \frac{\partial R}{\partial \nabla u} \right] \tag{20}
\]

where \( f = f(x, u(x), \nabla u(x)) \). After equating to zero the last equation, our original problem is then reduced to solving the following PDE for particular choices of \( L \) and \( R \):

\[
dL = \lambda \left( dR - \nabla \cdot \frac{\partial R}{\partial \nabla u} \right) = 0 \tag{21}
\]

From the definition of \( L \) in (8) we get

\[
dL = \sigma(u) - y \tag{22}
\]

and for \( R(u, \nabla u) = ||\nabla u||^2 \) as defined in (11) we get

\[
dR - \nabla \cdot \frac{\partial R}{\partial \nabla u} = \frac{d}{du} \frac{||\nabla u||^2}{2} - \nabla \cdot \frac{\partial ||\nabla u||^2}{2} = -\Delta u \tag{23}
\]

Then the condition of optimality is the elliptic PDE

\[
\sigma(u) - y - \Delta u = 0 \tag{24}
\]

To solve the problem numerically, we recall that \( u \) depends on \( w \). Then the problem is reduced to finding the weight vector \( w^* \) which appropriately satisfies (25).

\[
\sigma(u) - \Delta u = \alpha \tag{25}
\]

Function to fit \hspace{2cm} Targets

Having \( N \) datum pairs \((x_i, y_i)\) it is possible to write the problem of finding \( w^* \) as a least squares problem (LSQP). Let \( g(x_i, w) = \sigma(u(x_i, w)) - \Delta u(x_i, w) \) then the LSQP is:

\[
w^* = \arg \min_w \sum_{i=1}^N |y_i + \Delta u(x_i, w) - \sigma(u(x_i, w))|^2 \tag{26}
\]
\[
= \arg \min_w \sum_{i=1}^N \left[ y_i - g(x_i, w) \right]^2 \tag{27}
\]

\[
= \arg \min_w \sum_{i=1}^N \left[ y_i - f_i(w) \right]^2 \tag{28}
\]

\[
= \arg \min_w S(w), \tag{29}
\]

Note we have defined \( f_i(w) = g(x_i, w) \) to obtain the third step. In our particular case, we solve this problem via the Levenberg-Marquardt (LM) method. Notice that a damping parameter \( \eta \) that have to be selected is introduced by the LM method.

**Training with LM**

The LM algorithm consists in estimating \( w^* \) by following a sequence of estimations, each better than the previous one, so that \( w^{(j+1)} = w^{(j)} + \delta \) and therefore \( w^{(n)} \to w^* \) as \( n \to \infty \). We can determine \( \delta \) by looking at the linearisation of \( g_i(w + \delta) = g(x_i, w + \delta) \) which is given by

\[
g_i(w + \delta) \approx g_i(w) + \nabla g_i(w) \delta \tag{30}
\]

\[
\nabla g_i(w) = \nabla_w g(x_i, w) \tag{31}
\]

Equation (31) is the gradient of \( g \) with respect to \( w \). By looking at the expression of the linearisation of the function, we can expand it to get

\[
S(w + \delta) \approx \sum_{i=1}^N \left[ y_i - g_i(w) - \nabla g_i(w) \delta \right] \nonumber
\]

\[
= \left[ y - g(w) \right]^T J \delta = \ldots \nonumber
\]

\[
= \left[ y - g(w) \right]^T \left[ y - g(w) \right] - 2\left[ y - g(w) \right]^T J \delta + \delta^T J^T J \delta. \tag{32}
\]

Taking the derivative of (32) with respect to \( \delta \), we see that

\[
(J^T J)\delta = J^T \left[ y - g(w) \right] \tag{33}
\]

must be satisfied. This last line can also be interpreted as “solve for \( \delta \)”. It is desirable, however, this method to converge smoothly to a solution and so LM added a damping factor to the equation above. Therefore the LM algorithm at each step will solve a damped version of the previous equation:

\[
(J^T J + \eta \text{diag}(J^T J)) \delta = J^T \left[ y - g(w) \right] \tag{34}
\]

where \( \eta \) is a damping parameter which may be fixed or calculated at each iteration. Finally, we can give the algorithm to train our model along with the mathematical expressions used.

**Algorithm 1:** Training the model (Levenberg-Marquardt)

**Data:**

- \( x \) such that \( x = (x_1, \ldots, x_N)^T \) is the matrix of observations
- \( \lambda \) is a regularization parameter
- \( c \) is the fitting degree
- \( \eta \) is a dampening parameter
- \( M \) is the number of iterations

**Result:** The weight vector \( w^* \)

begin

\[
w(0) \leftarrow (0, \ldots, 0)^T; \tag{35}
\]

while \( j < M \) do

Solve \( J^T J + \eta \text{diag}(J^T J) \delta = J^T \left[ y - g(w) \right] \)

for \( \delta \)

\[
w(j+1) \leftarrow w(j) + \delta \nonumber
\]

end

We now proceed to give the expressions used in the algorithm.

\[
y = [y_1, \ldots, y_N]^T \nonumber
\]

\[
g(w) = \left[ g_1(w), \ldots, g_N(w) \right]^T \nonumber
\]

\[
J = \left[ \nabla g_1(w), \ldots, \nabla g_N(w) \right]^T \nonumber
\]

\[
\nabla g_i(w) = \nabla_w \sigma(u) - \lambda \nabla_w \Delta u \nonumber
\]

\[
\nabla_w \sigma(u) = \sigma'(u)(1 - \sigma(u))[\phi_1, \ldots, \phi_N]^T \nonumber
\]

\[
\nabla_w \Delta u = [\Delta \phi_1, \ldots, \Delta \phi_N]^T \nonumber
\]

\[
\Delta \phi_i = c(c|x_i|^2 - d) \phi_i \nonumber
\]

**Results**

In this section we present some results to illustrate the performance of our Regularized Logistic Regression (RLR) model. We tested RLR against the nine binary datasets shown in Table 1. For comparison purposes we decided to compare the results against those obtained from \( L_1 \) and \( L_2 \) regularized logistic regression (classical regularization over the weights), SVM and NN.

The datasets were standardized, i.e. all features in each dataset were centered to zero mean and unit variance. No dimensionality reduction was used in the experiments.

For each dataset we calculated two metrics: Accuracy and Area under the ROC curve (AUC).
The 5-fold cross validation scheme was also used in both metrics. In the experiments, we took care of selecting the different parameters of each method fairly.

For our RLR model we have to choose the triplet \((c, k, g)\), where \(c\) is the fitting degree, \(k\) is the regularization parameter and \(g\) is the LM dampening parameter.

For classical \(L_1\) and \(L_2\) regularized logistic regression, we have to select the regularization parameter \(C\). For a RBF-kernel SVM, we have to select the pair \((C, \gamma)\); where \(C\) is the penalty parameter while \(\gamma\) is the kernel coefficient. The architecture of the NN is a multi-layer perceptron with one hidden layer of 100 nodes which depends on a regularization term \(\alpha\). Both regularized logistic models, as well as SVM and NN, are implementations of the Python library sklearn. The parameters for each model were chosen by the following methodology:

**RLR**

The triplet \((c, \lambda, \eta)\) is searched as follows:

- \(c\) is searched on \((0, 5)\) with step = \(\ln 2\).
- \(\lambda\) is searched on \([0, 10]\) with step = 1.
- \(\eta = 1\) was used for each dataset. This value was empirically found to be the best.

**L_1 and L_2 regularized logistic regression**

\(C\) is searched on \((0, 5)\) with step = \(\ln 2\).

**SVM**

The pair \((C, \gamma)\) is searched as follows:

- \(C\) is searched on \((0, 5)\) with step = \(\ln 2\).
- \(\gamma = 1/m\) was used for all datasets, where \(m\) is the number of features of the dataset. This value was empirically found to be the best.

The parameter \(\alpha = 0.001\) was used for all datasets.

**Accuracy.** In Table 2 we present the classification results in terms of accuracy. The first thing to notice is that both classical \(L_1\) and \(L_2\) regularized logistic regression models performed poorly. Other than the Heart dataset, where the accuracy was similar to that obtained from the other models (meaning that the Heart dataset is linearly separable), they consistently delivered low accuracy values. Due to this, we will not discuss more about these two models in the rest of the paper.

On the other hand, although RLR just outperformed SVM and NN on three datasets out of nine, in general we see that RLR is quite competitive when compared against these two fully established methods. We also calculated the accuracy distances for a better insight. This is, the absolute value of the residual between each method’s accuracy and the one of the top performer. The results are shown in Table 3. On average RLR was down by 0.73%, SVM by 0.51% and NN by 0.89%. With this in mind, we can say that in terms of accuracy, SVM is the best performer, while RLR is second and NN comes a close third.

**AUC score.** The AUC score is another way to evaluate the performance of a classifier. In Table 4, we present the AUC values obtained for the different models and datasets. We also applied a similar analysis to the one presented above for the accuracy values. This is, we calculated the AUC distances (the differences between AUC values) for each dataset. In order to compare the models, we computed the sum of the distances (instead of the averages) obtaining the values that can be seen at the bottom of Table 5: RLR = 0.12, SVM = 0.02 and NN = 0.17. The result suggests that SVM is the best performer, RLR the second and NN is third. This is

### Table 1. In this Table, we present the datasets we used for testing.

| Data   | Dimension | Number of samples |
|--------|-----------|-------------------|
| Australian | 14        | 690               |
| BT      | 4         | 748               |
| BC      | 30        | 569               |
| Bupa    | 6         | 345               |
| German  | 24        | 1000              |
| Haberman| 3         | 306               |
| Heart   | 13        | 270               |
| Sonar   | 60        | 208               |
| VC      | 6         | 310               |

BT: blood transfusion; BC: breast cancer; VC: vertebral column.

### Table 2. The accuracy (%) of each method is outlined in this table.

| Dataset   | \(L_2\) | \(L_1\) | RLR  | SVM  | NN  |
|-----------|---------|---------|------|------|-----|
| Australian| 81.19   | 85.49   | 86.66| 86.81| 87.82|
| BT        | 77.55   | 77.92   | 78.22| 78.22| 77.28|
| BC        | 94.19   | 95.77   | 97.71| 98.74| 98.26|
| Bupa      | 61.13   | 64.43   | 72.46| 72.46| 71.01|
| German    | 75.12   | 75.28   | 76.00| 76.60| 78.30|
| Haberman  | 71.80   | 71.62   | 73.54| 73.87| 74.52|
| Heart     | 84.00   | 84.06   | 84.22| 84.87| 84.01|
| Sonar     | 43.36   | 44.45   | 88.43| 88.92| 87.46|
| VC        | 78.55   | 78.94   | 86.77| 85.48| 83.87|
consistent with what was obtained when analyzing the accuracy values.

**Grading.** Here we introduce a more intuitive analysis based on the AUC. We consider this analysis more robust than the previous. The analysis consists in assigning a “grade” to the results of a classifier by specifying the following grading scheme:

| Grade | \( A \) | \( B \) | \( C \) | \( D \) | \( F \) |
|-------|-------|-------|-------|-------|-------|
| \( A \) | \( 0.9 \leq \text{AUC} \leq 1 \) | \( 0.8 \leq \text{AUC} < 0.9 \) | \( 0.7 \leq \text{AUC} < 0.8 \) | \( 0.6 \leq \text{AUC} < 0.7 \) | \( 0.5 \leq \text{AUC} < 0.6 \) |

By following the above grading scheme, we are partitioning discretely the interval \( [0, 1] \) and assigning each sub-interval a grade. Therefore, small variations within the sub-intervals will be neglected making the evaluation more robust. In Table 4 we see that each of the AUC scores has a letter assigned to it which represents the grade the method received on that particular dataset. To summarize, we arranged the number of times a method received a particular grade in Table 6 and to obtain a quantitative measure, we assigned each grade a value: \( A = 1 \), \( B = 2 \), ..., \( F = 5 \).

In the bottom row in Table 6 it is the weighted total of the grades. The lowest the total value, the better the method. Again SVM obtained the best score. This time, RLR came a close second with only a 1 point difference and NN was very far by 4 points.

**Conclusion**

In this work, we have shown the way a new method for the supervised learning problem can be developed from a variational point of view. We have seen that it is very possible to formally formulate the problem as the minimization of a functional which has an associated PDE that describes the necessary conditions that a minimizer must meet. This equation proved to be easy to solve numerically with the least-squares algorithm Levenberg Marquardt. Tests were conducted and we compared the performance of our own model with the classical \( L_1 \) and \( L_2 \) regularized logistic regression models, NN and an RBF SVM. Both classical regularized models performed poorly as expected in every non-linearly separable dataset. By regularizing \( u \) instead of \( w \) we were able to improve results substantially. From this...
analysis, it was clear that while SVM beat both our model and NN, our model outperformed NN while staying relatively close to the performance of SVM.

Finally, like all formal analyses, it was necessary to prove the existence and uniqueness of a solution for our functional. The validity of this claim turned out to be true by Theorem 3.1. Even more, we were able to prove that, indeed, a solution of the PDE is a minimizer of a functional, thereby, formally justifying our argument that we could minimize our functional by solving the condition of optimality. By and large, our true intention was to showcase that a variational approach can be used and should be used for tackling the supervised learning problem. A variational approach encompasses the areas of functional analysis and the calculus of variations, both of which permitted us to make this work possible. Using the methods of the calculus of variations we were able to guarantee a solution which was facilitated by the theory of functional analysis. Therefore, we can only conclude that the choice of a variational approach in solving the supervised learning problem was satisfactory.

The implementations for the RLR model, benchmarking scripts and datasets may all be found at https://github.com/carlosb/thesis.

Declaration of Conflicting Interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs
Carlos Brito-Pacheco https://orcid.org/0000-0002-5913-2847
Carlos Brito-Loeza https://orcid.org/0000-0003-2970-2113

References
1. Singh A, Thakur N and Sharma A. A review of supervised machine learning algorithms. In: 3rd International conference on computing for sustainable global development (INDIACom), New Delhi, 16–18 March 2016, pp.1310–1315. IEEE.
2. Fernández-Delgado M, et al. Do we need hundreds of classifiers to solve real world classification problems. J Mach Learn Res 2014; 15: 3133–3181.
3. Kotsiantis SB. Supervised machine learning: a review of classification techniques, Informatica, 2007; 31: 249–268.
4. Caruana R, and Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on machine learning, 2006. New York: ACM.
5. Schwenker F and Trentin E. Pattern classification and clustering: a review of partially supervised learning approaches. Pattern Recogn Letters 2014; 37: 4–14.
6. Shore J and Johnson R. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. IEEE Trans Inform Theory 1980; 26: 26–37.
7. Ehsanes AK, Saleh Md, Arashi Mohammad, Kibria BM. Golam. Theory of Ridge Regression Estimation with Applications. John Wiley and Sons, 2019.
8. Santosa F and Symes WW. Linear inversion of band-limited reflection seismograms. SIAM J Sci Stat Comput 1986; 7: 1307–1330.
9. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc, Series B 1996; 58: 267–288.
10. Belkin M, Niyogi P and Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 2006; 7: 2399–2434.
11. Lin T, Xue H, Wang L, et al. Supervised learning via Euler’s elastica models. J Mach Learn Res 2015; 16: 3637–3686.
12. Moré JJ (1978) The Levenberg-Marquardt algorithm: Implementation and theory. In: Watson GA (eds) Numerical Analysis. Lecture Notes in Mathematics, vol. 630. Berlin, Heidelberg: Springer.
13. Hanley JA and McNeil BI. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 1982; 143: 29–36.
14. Pedregosa F, et al. Scikit-learn: machine learning in python. J Mach Learn Res 2011; 12: 2825–2830.
15. Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett 2006; 27: 861–874.

Appendix 1.

In this section, we aim to derive the weak Euler-Lagrange equation and see that solutions of it are in fact minimizers of $F[u]$. We shall require that the supervised learning Lagrangian be in in the class of functions $C^1$ and its derivatives satisfy growth conditions.

Euler-Lagrange equation proof

**Theorem 0.3.** Let $\Omega$ be an open, bounded set with boundary $\partial \Omega$ and $L$ be a Lipschitz continuous such that $L_z$ is continuous itself. If $\bar{u}$ is a minimizer of $F[u]$, then it must satisfy

$$\int_\Omega \lambda (\nabla \bar{u}, \nabla \delta u) + L_z(\bar{u}, y) \delta u \, dx = 0, \forall \delta u \in W^{1,2}_0$$  \hspace{1cm} (35)

for any $x$ not in a set of measure zero.

**Proof.** First, we need to prove that the equation makes sense. That is, that $L_z \delta u \in L^1$ and $(\nabla \bar{u}, \nabla \delta u) \in L^1$. The latter is straightforward, thus we will focus on the
former. Since $\Omega$ is bounded with a regular enough boundary, we will use the fact that $L$ is Lipschitz to deduce that $L \in W^{1,\infty}$. Then, the gradient of $L$ must be essentially bounded and furthermore $L_z \in L^\infty$. Again, using the fact that $\Omega$ is bounded we deduce $L_z \in L^1$.

Now set

$$h(\epsilon) := F[\bar{u} + \epsilon \delta u]$$

and notice that for $\epsilon \neq 0$

$$\frac{h(\epsilon) - h(0)}{\epsilon} = \int_\Omega \frac{1}{\epsilon} \left[ L(u + \epsilon \delta u) - L(u) + \frac{\lambda}{2} |\nabla \bar{u} + \epsilon \nabla \delta u|^2 - \frac{\lambda}{2} |\nabla u|^2 \right] \, dx = \int_\Omega g_\epsilon(x) \, dx$$

where we have defined $g_\epsilon(x)$ as the argument of the integral above. Since a function that is Lipschitz is also absolutely continuous we can write

$$g_\epsilon(x) = \frac{1}{\epsilon} \int_0^\epsilon \frac{d}{d\tau} \left[ \frac{\lambda}{2} |\nabla \bar{u} + \tau \delta u|^2 + L(u + \tau \delta u) \right] \, d\tau$$

$$= \frac{1}{\epsilon} \int_0^\epsilon \langle \nabla \bar{u} + \tau \nabla \delta u, \nabla \delta u \rangle + L_z(u + \tau \delta u, y) \delta u \, d\tau$$

Consequently, we have

$$\int_\Omega g_\epsilon(x) \, dx = \int_\Omega \frac{1}{\epsilon} \int_0^\epsilon \langle \nabla \bar{u} + \tau \nabla \delta u, \nabla \delta u \rangle + L_z(u + \tau \delta u, y) \delta u \, d\tau \, dx$$

Finally, let us conclude that since $L_z$ is essentially bounded we must have

$$g_\epsilon(x) \leq \lambda |\nabla \bar{u}|^2 + \lambda |\nabla \delta u|^2 + C \in L^2(\Omega), \epsilon \neq 0$$

Therefore, we can see that $g_\epsilon$ is dominated by a function allowing us to apply the dominated convergence theorem to deduce

$$\int_\Omega g_\epsilon(x) \, dx \to \int_\Omega \langle \nabla \bar{u}, \nabla \delta u \rangle + L_z(u, y) \delta u \, dx =: h'(0)$$

when $\epsilon \to 0$. Unquestionably, since $\bar{u}$ was a minimizer then $h(0)$ must be a minimum and so $h'(0) = 0$. \[\square\]