Permanent Disability Classification using Hybrid Neuro-Logistic Regression Models

Pedro A. Gutiérrez*, Manuel Baena-García†, Rafael Morales-Bueno† and César Hervás-Martínez*
*Department of Computer Science and Numerical Analysis, University of Córdoba, 14017, Córdoba, Spain
Email: {pagutierrez,chervas}@uco.es
†Departamento de Lenguajes y Ciencias de la Computación Universidad de Málaga, 29071 Málaga, Spain
Email: {mbaena,rmorales}@uma.es

***Abstract*—The social security administrations have to evaluate the degree of disability to offer compensation to those workers who suffer from a continuous alteration of health preventing them from continuing their work. Thanks to the accurate model of classification of disability presented in this paper, it is possible to obtain an approximation of the expected result for each case of disability prior to an individualized evaluation. In this paper, we introduce a novel model for classification of permanent disability, based on the hybridization of a standard logistic regression with Product Unit (PU) neural networks and Radial Basis Function (RBF) networks. The proposed techniques are shown to perform better than other existing Statistical and Artificial Intelligence methods.**

I. Introduction

Permanent disability is a term used in the insurance industry and law. Generally speaking, it means that due to a sickness or injury a person is unable to work in their own, or any occupation for which they are suited by training, education, or experience. In Spain, the evaluation and classification of permanent disability follows a procedure which is clearly defined and divided into three development phases: introduction, instruction and resolution.

The main principles of the measures considered to obtain a consolidated and rational system of permanent disability classification are the contributory element, equity and solidarity. Furthermore, in order to establish greater legal security in the process of determining permanent disability, it is mandatory to elaborate a list of diseases and evaluate their influence on the reduction of work capacity. This list must be created according to objective criteria based on the actual evaluations and the reports of the *disability assessment teams*.

To understand the nature of *permanent disability*, the terminology is defined next. Permanent disability takes into account continuous alteration of health and its impact on the worker's occupational situation. The *disability assessment team* is supported by a *medical unit*. The medical unit's tasks are: to examine the disability situation of the worker, to determine the reduction or alteration of the physical integrity of the worker, to determine the level of incapacity for working, to determine whether the character of the disease is common or professional, to extend the period of medical observation in case of professional diseases, to monitor programs for the control of temporal disability compensations, and to provide technical assistance and advice on any contentious issues concerning occupational disabilities.

In our work we consider three main categories that can be assigned to a worker depending on the degree of permanent disability: *no disability* (when the worker is not assigned the status of permanent disability), *permanent disability* (when the worker is assigned some degree of permanent disability) and *fee* (when the worker is not assigned any degree of permanent disability, but it is financially compensated).

Tulo et al. expose in their work [1] the necessity of establishing innovative solutions regarding to permanent disability evaluation process, and data mining techniques are referenced as a tool to uncover trends in this evaluation. In this sense, the motivation of this study is to obtain an initial model which helps to prepare reports in the process of determining permanent disability. Using this model, a worker could obtain an approximation of the expected result for each case of permanent disability evaluation. The training dataset used to obtain the model is composed of information from reports of a medical unit. Each report is tagged with one of the three categories (no disability, permanent disability or fee). An important characteristic of the dataset is that it is highly unbalanced.

Other related works, using data mining techniques for occupational health analysis, includes the identification of relationships between occupational risk level and immediate/root causes and corrective actions [2], the identification of specific risk groups and factors in occupational safety using regression trees [3], and the identification of characteristics of occupational injuries in the construction industry using association rules [4].

In recent years, artificial neural networks (ANNs) and other related techniques have been successfully used for assessing the risk of occupational injury [5], for classification of industrial jobs with respect to the risk of work related low back disorders [6] and for forecasting occupational accidents [7]. The most popular neural network model is perhaps the back-propagation (BP) neural network [8] due to its simple architecture yet powerful problem-solving ability. In ANNs, the hidden neurons are the functional units and can be considered as generators of function spaces. Most existing neuron models are based on the summing operation of the inputs, and, more particularly, on sigmoidal unit functions,

978-1-4244-9908-3/11/$26.00 ©2011 IEEE

resulting in what is known as the Multilayer Perceptron (MLP). However, alternatives to MLP emerged in the last few years: Product Unit Neural Network (PUNN) models are an alternative to MLPs and are based on multiplicative neurons instead of additive ones. They correspond to a special class of feed-forward neural network introduced by Durbin and Rumelhart [9]. While MLP network models have been very successful, networks that make use of Product Units (PUs) have the added advantage of increased information capacity [9]. That is, smaller PUNNs architectures can be used rather than those used with MLPs [10]. They aim to overcome the non-linear effects of variables by means of non-linear basis functions, constructed with the product of the inputs raised to arbitrary powers. These basis functions express possible strong interactions between the variables, where the exponents may even take on real values and are suitable for automatic adjustment.

Another interesting alternative to MLPs are Radial Basis Function networks (RBFNNs). RBFNNs can be considered a local approximation procedure, and the improvement in both its approximation ability, as well as in the construction of its architecture has been noteworthy [11]. RBFNNs have been used in the most varied domains, from function approximation to pattern classification, time series prediction, data mining, signals processing, and non-linear system modelling and control [12]. RBFNNs use, in general, hyper-ellipsoids to split the pattern space. In many cases, MLP, PU and RBF networks are trained by using evolutionary algorithms (EAs),thus obtaining advantages with respect to traditional training approaches [13], [14], [15].

Logistic Regression (LR) has become a widely used and accepted method of analysis of binary or multi-class outcome variables as it is more flexible and it can predict the probability of the state of a multi-class variable based on the predictor variables. In this paper, we consider the hybridization of some novel networks (PUs and RBFs) with a standard logistic regression to improve the performance of the classifiers in the problem of permanent disability prediction. The hybridization of LR and PUNNs or RBFNNs is done by considering some works in classifier construction, where the hybridization of the LR model and Evolutionary PUNNs (EPUNNs) to obtain binary [16], [17] and multiclass [18] classifiers is proposed. The work has been recently adapted for RBFs [19]. In a first step, an evolutionary algorithm [13] is used to determine the basic structure of the product-unit model. That step can be seen as a global search in the space of the model coefficients. Once the basis functions have been determined by the EA, a transformation of the input space is considered. This transformation is performed by adding the non-linear transformations of the input variables given by the PU functions obtained by the EA. The final model is linear in these new variables together with the initial covariates. On the other hand, the hybridization of the LR and evolutionary RBFNNs is also tested in this paper, in such a way that we combine a linear model with a Radial Basis Function Neural Network (RBFNN) non-linear model and then we estimate the coefficients using logistic regression.

The main contribution of present work is the generation of a model based on artificial neural networks and logistic regression that can be used to obtain an prediction of permanent disability. We experimentally show that the hybrid models involving LR, PUNNs and RBFNNs outperform several other existing classification techniques in the problem of permanent disability prediction, and they are therefore a very interesting tool to be taken into account in this field.

II. Occupational Situation and Permanent Disability

Permanent disability (PD), in its contributory modality, takes into account the continuous alteration of health and, particularly, its impact on occupational situation.

It has an exclusively professional profile and its evaluation should avoid references to other circumstances, such as socio-economic status, age, family, etc. These circumstances may be considered in order to evaluate other effects, but should not be taken into account when determining the degree of disability to be protected by contributory income.

The occupational situations to be protected by the status of permanent disability are:

- Permanent disability which, in practice, stands for the lack of income due to the loss of salary which is a result of either temporary, or permanent disability. This lack of income is alleviated by financial aid.
- The necessity to recover psycho-physical well being.
- The necessity to receive financial support during the process of recovery.
- The process of reintegrating a disabled person into work environment, which should be protected by selective employment.

Depending on the determining cause, permanent disability is classified according to the following degrees:

- *Partial PD* for usual occupation means that a worker's capacity to perform his/her job is diminished by not less than 33%. However, it does not prevent him/her from performing tasks which are fundamental for his/her occupation.
- *Total PD* for usual occupation means that a worker is unable to perform tasks which are fundamental for his/her occupation, but may opt for a different occupation.
- *Absolute PD* means that a worker is unable to perform any profession.
- *Grand disability* means that a worker who is affected by PD due to his/her physical and functional impairments requires assistance in basic life activities such as dressing up, moving from one place to another, eating, etc.
- Non-Disabling Permanent Damages refers to permanent impairments which do not have impact on work capacity, but mean that a worker's physical integrity is reduced.

In case of accidents the term "usual occupation" should be understood as the work performed by the worker at the moment of the accident.

A. Initial data and variables

The medical unit of the disability assessment team elaborates synthesis medical reports (SMR) to evaluate permanent disability. We use these reports as a source of information for our experiments. Synthesis medical reports are based on:

1) Clinical examination performed by a medical evaluator.
2) Medical reports provided by the patient.
3) Complementary tests and examinations requested by the medical evaluator.

The data used here had been obtained from the SMRs and the reports of the sessions held by the disability assessment team which were then compiled into files. Some data, like age or sex, have been extracted directly from these documents while others, like occupational repercussion, have been collected by qualified personal.

For each file, the following attributes have been obtained:

- From the SMRs:
  Age, sex, occupation, sick leave period, and diseases.
- From the reports of the sessions held by the disability assessment team:
  Classification (permanent disability degree), contingency, and period of time between examinations.
- From the qualified personal:
  Occupational repercussion. The following information has been taken into account when evaluating it as low, middle or high:
  - Functional repercussion of different diseases.
  - Worker's occupation.

The classification (permanent disability degree) is grouped into:

- No disability (ND).
- Permanent disability (PD).
- Fee (F).

The contingency can be classified into two types:

- Common
  - Common disease (CD)
  - Non-working accident (NWA)
- Professional
  - Occupational disease (OD)
  - Working accident (WA)

We have considered the code of the Spanish "National Classification of Occupations" (CNO-94) to collect the data related to the professions. To gather the data related to the diseases, the "International Classification of Diseases" (ICD9-CM) has been considered.

The final variables used in our work are shown in table I. A total of 978 records have been extracted from the data between 2002 and 2003.

TABLE I
LIST OF VARIABLES AND ASSOCIATED DESCRIPTION OF THE DATASET OBTAINED FROM THE SYNTHESIS MEDICAL REPORTS AND THE REPORTS OF THE SESSIONS HELD BY THE DISABILITY ASSESSMENT TEAM

Variable	Description
x_1	Age
x_2	Sex
x_{3-21}	CNO-94
x_{22}	Sick leave time
x_{23-42}	Principal categories of ICD9-CM
x_{43}	Low occupational repercussion
x_{44}	Middle occupational repercussion
x_{45}	High occupational repercussion
x_{46}	Total number of diseases
x_{47}	CD contingency
x_{48}	NWA contingency
x_{49}	OD contingency
x_{50}	WA contingency
x_{51}	Period of time between examinations
Class	Description
ND	No disability
PD	Permanent disability
F	Fee

III. NEURO-LOGISTIC MODELS

In the classification problem, some measurements x_i, $i = 1, 2, ..., k$ are taken on a single pattern, and the patterns are classified into one of J categories. The measurements x_i are random observations from these J classes, J being a finite number. A training sample $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2, ..., N\}$ is available, where $\mathbf{x}_n = (x_{1n}, ..., x_{kn})$ is the vector of measurements taking values in $\mathbf{\Omega} \subset \mathbb{R}^k$, and $\mathbf{y}_n$ is the class level of the n-th individual. The common technique of representing the class levels using a "1-of-J" encoding vector is adopted, $\mathbf{y} = \left(y^{(1)}, y^{(2)}, ..., y^{(J)}\right)$, such as $y^{(l)} = 1$ if $\mathbf{x}$ corresponds to an example belonging to class l and $y^{(l)} = 0$ otherwise. Based on the training sample, we wish to find a decision function $F : \mathbf{\Omega} \to \{1, 2, ..., J\}$ for classifying the individuals. In other words, F provides a partition, say $D_1, D_2, ..., D_J$, of $\mathbf{\Omega}$, where D_l corresponds to the l-th class, $l = 1, 2, ..., J$, and measurements belonging to D_l will be classified as coming from the l-th class. A misclassification occurs when the decision rule F assigns an individual (based on the measurement vector) to a class j when it is actually coming from a class $l \neq j$.

To evaluate the performance of the classifiers the corrected classified rate (CCR or C) is defined by

$$C = \frac{1}{N} \sum_{n=1}^{N} I(F(\mathbf{x}_n) = \mathbf{y}_n), \tag{1}$$

where $I(\bullet)$ is the zero-one loss function. A good classifier tries to achieve the highest possible C in a given problem. It is usually assumed that the training data are independent and identically distributed samples from an unknown probability distribution. Suppose that the conditional probability that $\mathbf{x}$ belongs to class l verifies: $p\left(y^{(l)} = 1 \middle| \mathbf{x}\right) > 0$, $l = 1, 2, ..., J$, $\mathbf{x} \in \mathbf{\Omega}$, and sets the function:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \log \frac{p\left(y^{(l)} = 1 \middle| \mathbf{x}\right)}{p\left(y^{(J)} = 1 \middle| \mathbf{x}\right)},$$

where θ_l is the weight vector corresponding to class l, and $f_J(\mathbf{x}, \theta_J) = 0$. Under a multinomial logistic regression, the probability that $\mathbf{x}$ belongs to class l is then given by:

$$p\left(y^{(l)} = 1 \middle| \mathbf{x}, \theta\right) = \frac{\exp f_l\left(\mathbf{x}, \theta_l\right)}{\sum_{j=1}^{J} \exp f_j\left(\mathbf{x}, \theta_j\right)},\ l = 1, 2, ..., J,$$

where $\theta = (\theta_1, \theta_2, ..., \theta_{J-1})$. For binary problems ($J = 2$), this is known as logistic regression (or soft-max in neural network literature).

The classification rule coincides with the optimal Bayes' rule. In other words, an individual should be assigned to the class which has the maximum probability, given the measurement vector $\mathbf{x}$:

$$F(\mathbf{x}) = \hat{l}, \text{where } \hat{l} = \arg\max_{l} f_l(\mathbf{x}, \hat{\theta}_l), \text{for } l = 1, ..., J.$$

On the other hand, due to the normalization condition we have:

$$\sum_{l=1}^{J} p\left(y^{(l)} = 1 \middle| \mathbf{x}, \theta\right) = 1,$$

and the probability for one of the classes (the last one, in our case) does not need to be estimated. Observe that we have considered $f_J(\mathbf{x}, \theta_J) = 0$.

The hybrid Neuro-Logistic models are based on the combination of the standard linear model and nonlinear terms constructed with RBFs or PUs, which captures possible locations in the covariate space. The general expression of the model is given by:

$$f_l(\mathbf{x}, \theta_l) = \alpha_0^l + \sum_{i=1}^{k} \alpha_i^l x_i + \sum_{j=1}^{m} \beta_j^l B_j(\mathbf{x}, \mathbf{w}_j) \qquad (2)$$

where $l = 1, 2, ..., J-1$, $\theta_l = (\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \mathbf{W})$ is the vector of parameters for each discriminant function, $\boldsymbol{\alpha}^l = (\alpha_0^l, \alpha_1^l, ..., \alpha_k^l)$ and $\boldsymbol{\beta}^l = \left(\beta_1^l, ..., \beta_m^l\right)$ are the coefficients of the multilogistic regression model and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$ are the parameters of the nonlinear transformations. The difference between PUNNs and RBFNNs is related to the activation function considered in the hidden layer. In this way, Product Units (PUs) are considered for PUNNs:

$$B_j(\mathbf{x}, \mathbf{w}_j) = \prod_{i=1}^{k} x_i^{w_{ji}} \qquad (3)$$

where w_{ji} is the weight of the connection between input neuron i and hidden neuron j and $\mathbf{w}_j = (w_{j1}, \ldots, w_{jk})$ is the weight vector. On the other hand, Gaussian RBFs are considered for RBFNNs:

$$B_j(\mathbf{x}, \mathbf{w}_j) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{r_j^2}\right) \qquad (4)$$

where $\mathbf{w}_j = (\mathbf{c}_j, r_j)$, $\mathbf{c}_j = (c_{j1}, c_{j2}, \ldots,\ c_{jk})$ is the centre or average of the j-th Gaussian RBF transformation, r_j is the corresponding radius or standard deviation and $c_{ji}, r_j \in \mathbb{R}$. The general structure of this kind of models can be analyzed in Fig. 1.

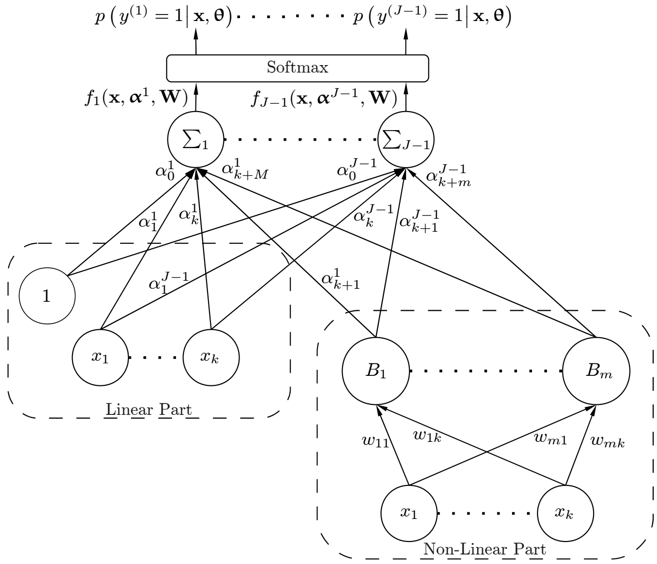

Fig. 1. Structure of Radial Basis Function Neural Networks: an input layer with k input variables, a hidden layer with m RBFs and an output layer with J nodes

IV. Hybrid Learning Algorithm

In the supervised learning context, the components of the weight vectors $\theta = (\theta_1, \theta_2, ..., \theta_{J-1})$ are estimated from the training dataset D. To perform the maximum likelihood estimation of θ, one can minimize the negative log-likelihood function:

$$\begin{aligned} L(\theta) = &-\tfrac{1}{N} \textstyle\sum_{n=1}^{N} \sum_{l=1}^{J} \left(y_n^{(l)} \log p\left(\mathbf{y}_n | \mathbf{x}_n, \theta\right)\right) = \\ &= \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \Big[-\sum_{l=1}^{J} y_n^{(l)} f_l(\mathbf{x}_n, \theta_l) + \\ &\quad + \log \textstyle\sum_{l=1}^{J} \exp f_l(\mathbf{x}_n, \theta_l)\Big], \end{aligned} \qquad (5)$$

where $f_l(\mathbf{x}, \theta_l)$ corresponds to the hybrid model defined in (2).

The error surface associated with the model is convoluted with numerous local optima. Given the nonlinearity of the model with respect to the parameters $\mathbf{w}_j$, and the indefinite character of the associated Hessian matrix of $L(\theta)$,the use of gradient-based methods to maximize the log-likelihood function is not recommended. Moreover, the optimal number of basis functions of the model (i.e. m) is unknown. Thus, the estimation of the vector parameter $\hat{\theta}$ is carried out by means of a hybrid procedure described below.

The methodology proposed is based on the combination of an Evolutionary Programming algorithm (EP) (global explorer) and a local optimization procedure (local exploiter) carried out by the standard maximum likelihood optimization method. In the first step, the EP algorithm is applied to design the structure and training of the weights of a neural network. Once the basis functions have been determined by the EP algorithm, we consider a transformation of the input space by adding these nonlinear transformations given by the RBFs of the best individual in the final generation of the EP algorithm.

The model is now linear in these new variables and the initial covariates. The remaining coefficient vector $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are calculated by the *maximum likelihood* optimization method which selects the parameters that maximize the probability of the observed data points. In the next subsection, the algorithms for obtaining this maximum likelihood solution are described. Then, the different steps of the MLRIRBF learning algorithm are described and, in the last subsection, the details of the EP algorithm are given.

A. Algorithms for Multilogistic Regression Maximum Likelihood Optimization

In this paper, two different algorithms have been considered for obtaining the maximum likelihood solution for the multilogistic regression model, both available in the WEKA machine learning workbench [20]:

1) MultiLogistic: MultiLogistic is an algorithm for building a multinomial logistic regression model with a ridge estimator to guard against overfitting by penalizing large coefficients, based on work of Le Cessie and Van Houwelingen [21].

In order to find the coefficient matrix $\boldsymbol{\theta}$ for which $L(\boldsymbol{\theta})$ is minimized, a Quasi-Newton Method is used. Specifically, the method used is the active-sets' method with Broyden-Fletcher-Goldfarb-Shanno (BFGS) update [22].

2) SimpleLogistic: This algorithm builds multinomial logistic regression models fitting them using the LogitBoost algorithm [23], which was proposed by Friedman et al. for fitting *additive logistic regression models* by maximum likelihood. These models are a generalization of the (linear) logistic regression models described above. *SimpleLogistic* algorithm is based on applying LogitBoost with simple regression functions and determining the optimum number of iterations by a five fold cross-validation: the data is equally split five times into training and test, LogitBoost is run on every training set up to a maximum number of iterations (500) and the classification error on the respective test set is logged. Afterwards, LogitBoost is run again on all data using the number of iterations that gave the smallest error on the test set averaged over the five folds. Further details about the algorithm can be found in [24].

B. Estimation of the model coefficients

The process is divided into three steps.

Step 1. We apply an EP algorithm to find the basis functions:

$$\mathbf{B}(\mathbf{x},\mathbf{W}) = \{B_1(\mathbf{x},\mathbf{w}_1), B_2(\mathbf{x},\mathbf{w}_2), ..., B_m(\mathbf{x},\mathbf{w}_m)\},$$

corresponding to the nonlinear part of $f(\mathbf{x},\boldsymbol{\theta})$. We have to determine the number of basis functions m and the weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$. To apply evolutionary neural network techniques, we consider a neural network with softmax outputs and the standard structure: an input layer with a node for every input variable; a hidden layer with several nodes; and an output layer with one node for each class minus one. There are no connections between the nodes of a layer and none between the input and output layers either.

The weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$ is estimated by means of an evolutionary neural network algorithm that optimizes the error function given by the negative log-likelihood for N observations associated with the neural network model (see (5)).

The specific details of this EP algorithm can be found in some previous works [25], [26].

Although in this step the evolutionary process obtains a concrete value for the $\boldsymbol{\beta}$ vector, we only consider the estimated weight vector $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_m)$, which builds the basis functions. The values for the $\boldsymbol{\beta}$ vector will be determined in step 3 together with those of the $\boldsymbol{\alpha}$ coefficient vector.

Step 2. We consider the following transformation of the input space by including the nonlinear basis functions obtained by the EP algorithm in step 1:

$$\begin{aligned}&H : \mathbb{R}^k \to \mathbb{R}^{k+m},\\&(x_1, x_2, ..., x_k) \to (x_1, x_2, ..., x_k, z_1, ..., z_m),\end{aligned}$$

where $z_1 = B_1(\mathbf{x}, \hat{\mathbf{w}}_1), ..., z_m = B_m(\mathbf{x}, \hat{\mathbf{w}}_m)$.

Step 3. In the third step, we minimize the negative log-likelihood function for N observations:

$$\begin{aligned}L(\boldsymbol{\alpha},\boldsymbol{\beta}) = &\frac{1}{N}\sum_{n=1}^{N}\Big[-\sum_{l=1}^{J} y_n^{(l)}(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n)+\\&+\log\sum_{l=1}^{J}\exp(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n)\Big],\end{aligned}$$

where $\mathbf{x}_n = (1, x_{1n}, ..., x_{kn})$ and $\mathbf{z}_n = (z_{1n}, ..., z_{kn})$. Now, the Hessian matrix of the negative log-likelihood in the new variables $x_1, x_2, ..., x_k, z_1, ..., z_m$ is semi-definite positive. The estimated coefficient vector $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\mathbf{W}})$ determines the model of (2) with $B_j(\mathbf{x}, \mathbf{w}_j)$ defined as (3) or (4).

In this final step, both algorithms presented in subsection IV-A have been used for obtaining the parameter matrix $\boldsymbol{\theta}$. $B_j(\mathbf{x}, \mathbf{w}_j)$ is defined by (3) or (4). Moreover, two different versions of the hybrid neuro-logistic models have been considered:

- LR models with only the non-linear part, i.e. the model does not include the initial covariates of the problem (left part of Fig. 1. The different neural networks, together with the two ML algorithms applied in Step 3, result in four different methods: MultiLogistic regression using Product Units (MLPU), SimpleLogistic regression using Product Units (SLPU), MultiLogistic regression using RBFs (MLRBF) and SimpleLogistic regression using RBFs (SLRBF).
- LR models with both the linear and the non-linear part, i.e. the models of Fig. 1. In the same way that with the previously presented models, two different logistic regression algorithms are applied (SimpleLogistic and MultiLogistic) over the two different models defined by considering (3) or (4) for $B_j(\mathbf{x}, \mathbf{w}_j)$, what finally results in four different methods: MultiLogistic regression using Initial covariates and PUs (MLIPU), SimpleLogistic regression using Initial covariates and PUs (SLIPU), MultiLogistic regression using Initial covariates and RBFs (MLIRBF) and SimpleLogistic regression using Initial covariates and RBFs (SLIRBF).

V. Experiments

The methods compared are the following: PU methods (EPUNN, MLPU, SLPU, MLIPU and SLIPU) and RBF methods (ERBF, MLRBF, SLRBF, MLIRBF and SLIRBF). Different state-of-the-art Statistical and Artificial Intelligence algorithms have been considered for the sake of comparison. Specifically, the results of the following algorithms have been compared with the soft-computing techniques presented in this paper:

1) RandomForest, an ensemble classifier that consists of many decision trees [27].
2) Naive Bayes Tree learning algorithm (NBTree) [20].
3) Logistic Model Tree (LMT) [24] classifier.
4) The k Nearest Neighbour (k-NN) classifier, adjusting the value of k using a nested 10-fold cross-validation.
5) A Gaussian Radial Basis Function Network (RBFNetwork) [20], deriving the centers and width of hidden units using k-means, and combining the outputs obtained from the hidden layer using logistic regression.
6) Both standard logistic regression algorithms presented in section IV-A: SimpleLogistic (SLogistic) and MultiLogistic (MLogistic).
7) The Naive Bayes standard learning algorithm (NaiveBayes) [20].

These algorithms have been selected for comparison since they are some of the best performing algorithms of recent literature on classification problems.

A 10-fold cross-validation has been applied and the performance has been evaluated by using the C measure defined in (1) considering the generalization set (C_{G}). When applying the algorithms proposed (PU and RBF methods), ten repetitions are performed per each fold, and when applying the rest of methods, the 10-fold process is repeated ten times, in order to obtain an average and a standard deviation of the C_{G} from the same sample size (100 models).

In order to apply logistic regression analyses, all nominal variables of the problems have been transformed to binary ones, resulting in a total of 51 variables. Moreover, the variables are scaled to the $[0.1, 0.9]$ interval for PUs and to $[-2.0, 2.0]$ for RBFs.

The results are included in Table II, where the mean and standard deviation values of C_{G} are given. The first analysis of these results (without taking into consideration the statistical significance) reveals that the best performing method is SLIPU and the second one SLIRBF. Other important observation is that PU methods generally outperform their RBF equivalents, obtaining also a lower standard deviation. Consequently, PUs are better suited for classifying permanent disability than RBFs as they are charcaterized by a higher performance and stability.

In order to ascertain the statistical significance of the observed differences between the mean C_{G} of the best models obtained for each methodology, we have applied statistical tests. First of all, a non-parametric Kolmogorov-Smirnov test (KS-test) with a signification level $\alpha = 0.05$ was used to evaluate if the C_{G} values of all methods followed a normal distribution. A normal distribution cannot be assumed because only 8 from 18 obtained a p-value lower than the critical level (i.e. only a 44.44% of the methods follows a normal distribution). As a consequence, a non-parametric Kruskal-Wallis test for independent samples was selected in order to check if the methodology applied is significantly affecting the results obtained. The test concludes that these differences are significant (with a $p-$value $= 0.00$). So, we finish the statistical analysis applying the Mann-Whitney U rank sum test for all pairs of algorithms and the results are also included in Table II. These results include, for each algorithm, the number of algorithms statistically outperformed (Wins), the number of draws (non significant differences) and the number of loses (number of algorithms that outperform the method).

TABLE II
MEAN, STANDARD DEVIATION, MAXIMUM AND MINIMUM VALUES OF THE ACCURACY RESULTS (C_{G}) FROM 100 EXECUTIONS OF A 10-FOLD CROSS VALIDATION, USING THE DIFFERENT METHODS COMPARED. NUMBER OF WINS, DRAWS AND LOSES WHEN COMPARING THE DIFFERENT METHODS USING THE MANN–WHITNEY U RANK SUM TEST WITH $\alpha = 0.05$

	$C_{\mathrm{G}}(\%)$	Mann–Whitney U test		
	Mean±SD	# Wins	# Draws	# Loses
EPUNN	85.04 ± 9.48	5	3	9
MLPU	85.00 ± 9.50	5	3	9
SLPU	85.06 ± 9.52	5	5	7
MLIPU	86.64 ± 8.93	7	5	5
SLIPU	$\mathbf{90.05 \pm 9.44}$	**16**	**1**	**0**
ERBF	79.76 ± 11.36	1	2	14
MLRBF	79.88 ± 11.20	1	2	14
SLRBF	79.56 ± 13.54	1	2	14
MLIRBF	86.39 ± 8.96	5	7	5
SLIRBF	$\mathit{89.86 \pm 9.40}$	*14*	*3*	*0*
RandomForest	88.02 ± 9.41	12	1	4
NBTree	87.07 ± 9.35	8	5	4
LMT	89.85 ± 9.41	14	2	1
k-NN	66.04 ± 8.12	0	0	17
RBFNetwork	86.75 ± 9.30	8	4	5
SLogistic	89.77 ± 9.39	14	2	1
MLogistic	86.54 ± 9.31	7	5	5
NaiveBayes	84.17 ± 9.15	4	0	13

The best result is in bold face and the second best result in italic.

From the analysis of these results, the SLIPU method has to be highlighted as the most competitive one (with only one draw), followed by SLIRBF and then by LMT and SLogistic.

A. Extracting information from the best SLIPU model

In this subsection, the main objective is to obtain some information about the importance each variable has for the classification problem considered, by analyzing the probability expression of the best SLIPU model. This model is included in Table III. The reduced number of non-linear terms and variables is notably low, which makes the process of interpreting the expression easier. Let us highlight again that all the variables are scaled to $[0.1, 0.9]$ interval.

The model is composed of two hybrid functions, f_{ND} and

f_{PD} that share the same Product Unit (PU_1).

Starting with the interpretation of the linear part in the model for "No disability" (ND), the set of variables with higher absolute coefficient is represented by x_{42}, x_{47}, x_{43}, and x_{49}. These variables are associated with "injury and poisoning" ICD9 group (x_{42}), "common disease" contingency (x_{47}), the number of diseases with low occupational repercussion (x_{43}) and "working accident" contingency (x_{49}). While (x_{42}) is the variable with higher coefficient, it does not apply to most of the cases, hence (x_{43}) is the most important variable for this model.

Analyzing the linear part in the model for "Permanent Disability" (PD), we found that the set of variables with higher absolute coefficient is represented by x_{47}, x_{41} and x_{42}. The variable x_{41} is associated with "symptoms, signs, and ill-defined conditions" ICD9 group.

With regard to the Product Unit PU_1, the non-linear part of the model, we can observe that it is a critical component for the classification. PU_1 has a high positive influence on the classification of "No disability" files and a relatively high negative influence on the classification of "Permanent Disability".

The occurrence of variables representing *low occupational repercussion* has a clear influence on the first model. It also has an indirect influence on the rest of the models based on the complementary implication: a file without diseases of low occupational repercussion contains diseases of middle or high occupational repercussion. Hence there is a high probability that the file with diseases of middle or high occupational repercussion will be classified as Permanent Disability. In this sense, the criterion established by a qualified person plays a crucial role in determining permanent disability.

TABLE III
PROBABILITY EXPRESSION OF THE BEST SLIPU MODEL, C_{G} AND TEST CONFUSSION MATRIX

Best SLIPU Permanent Disability Probability Model

$$p_{\text{ND}} = \frac{e^{f_{\text{PD}}(\mathbf{x},\boldsymbol{\theta})}}{1+e^{f_{\text{PD}}(\mathbf{x},\boldsymbol{\theta})}+e^{f_{\text{ND}}(\mathbf{x},\boldsymbol{\theta})}}$$

$$\begin{aligned} f_{\text{ND}}(\mathbf{x},\boldsymbol{\theta}) = {} & -2.48 + 2.04x_1 - 1.50x_{10} - 2.43x_{11} + 1.78x_{22} - \\ & -1.26x_{33} + 0.91x_{35} + 0.95x_{37} + 16.56x_{42} + 7.50x_{43} - \\ & -6.83x_{47} + 3.70x_{49} + 6.76PU_1 \end{aligned}$$

$$p_{\text{PD}} = \frac{e^{f_{\text{ND}}(\mathbf{x},\boldsymbol{\theta})}}{1+e^{f_{\text{PD}}(\mathbf{x},\boldsymbol{\theta})}+e^{f_{\text{ND}}(\mathbf{x},\boldsymbol{\theta})}}$$

$$\begin{aligned} f_{\text{PD}}(\mathbf{x},\boldsymbol{\theta}) = {} & 3.83 + 0.33x_2 + 1.52x_{10} - 2.43x_{11} - 0.72x_{14} + \\ & +1.78x_{22} - 0.66x_{30} - 1.26x_{33} + 1.87x_{35} - 0.92x_{40} + \\ & +6.46x_{41} + 5.78x_{42} + 0.69x_{43} - 6.83x_{47} - 1.95x_{49} - \\ & -2.94PU_1 \end{aligned}$$

$$PU_1 = x_{14}^{-0.09}\ x_{22}^{0.06}\ x_{37}^{-0.14}\ x_{40}^{-0.28}\ x_{42}^{0.39}\ x_{43}^{0.67}$$

$x_1 \leftarrow$ (age); $x_2 \leftarrow$ (sex); $x_{10} \leftarrow$ (rcno94=12); $x_{11} \leftarrow$ (rcno94=4); $x_{14} \leftarrow$ (rcno94=7); $x_{22} \leftarrow$ (sick leave time); $x_{30} \leftarrow$ (disease8); $x_{33} \leftarrow$ (disease11); $x_{35} \leftarrow$ (disease13); $x_{37} \leftarrow$ (disease15); $x_{40} \leftarrow$ (disease18); $x_{41} \leftarrow$ (disease19); $x_{42} \leftarrow$ (disease20); $x_{43} \leftarrow$ (low occupational repercussion); $x_{47} \leftarrow$ (contingency=CD); $x_{49} \leftarrow$ (contingency=WA)

$x_i \in [0.1, 0.9]$

$C_{\text{G}} = 95.24\%$

Generalization Confusion Matrix

	Predicted		
Target	NI	I	B
NI	45	3	0
I	1	32	0
B	0	0	3

VI. CONCLUSIONS

The hybrid neuro-logistic models have proved to serve as an accurate tool in the classification of permanent disability. A comparative study between an extensive collection of standard classifiers and the results of the statistical tests applied, and the hybrid neuro-logistic models shows that the latter are more precise in determinig the degree of permanent disability.

Our hybrid models include a non-linear component (from different kinds of neural networks) and a standard linear component, combining both in a logistic regression predictor. The complexity of the model and the high amount of parameters involved in these classifiers encouraged us to use a combined methodology, including an evolutionary algorithm and a standard maximum-likelihood optimization process.

Useful information has been extracted from the most accurate model, given its simple structure (number of connections and number of hidden neurons). Simple structure is one of the main advantages of the models presented.

The obtained model is not intended to be a widely used tool in the classification of permanent disability. First, it would be necessary to examine more data as the scope of the PD problem is very broad due to the high number and complexity of cases. However, our findings can be used to develop new, improved systems. For instance, an extended model could be used to create an information system, both for patients and professionals, which would provide assistance in the evaluation of permanent disability.

ACKNOWLEDGMENT

This work was partially supported by the SESAAME project, number TIN2008-06582-C03-03, and the TIN 2008-06681-C06-03 project of the MICINN, Spain, FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía", Spain.

REFERENCES

[1] B. Tulu, B. Hilton, and T. A. Horan, "Improving disability evaluation productivity: linking innovative business models with information technology," *Int. J. Healthcare Technology Management*, 2005.

[2] M. Bevilacqua, F. E. Ciarapica, and G. Giacchetta, "Data mining for occupational injury risk: a case study," *International Journal of Reliability, Quality and Safety Engineering*, vol. 17, no. 4, pp. 351–380, 2010.

[3] A. Persona, D. Battini, M. Faccio, M. bevilacqua, and F. emanuele ciarapica, "Classification of occupational injury cases using the regression tree approach," *International Journal of Reliability, Quality and Safety Engineering*, vol. 13, no. 2, pp. 171–191, 2006.

[4] C.-W. Liao, "Association rule mining for occupational injuries in the construction industry," Ph.D. dissertation, National Taiwan University of Science and Technology, 2007.

[5] F. Ciarapica and G. Giacchetta, "Classification and prediction of occupational injury risk using soft computing techniques: An italian study," *Safety Science*, vol. 47, no. 1, pp. 36 – 49, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6VF9-4S1BX0T-1/2/3c6241502a223639a9dab86cb6490ca7
[6] S. Asensio-Cuesta, J. Diego-Mas, and J. Alcaide-Marzal, "Applying generalised feedforward neural networks to classifying industrial jobs in terms of risk of low back disorders," *International Journal of Industrial Ergonomics*, vol. 40, no. 6, pp. 629 – 635, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6V31-50M3CC6-1/2/cda32180e5b39fab9dd994075de78b30
[7] M. C. Carnero and D. J. Pedregal, "Modelling and forecasting occupational accidents of different severity levels in spain," *Reliability Engineering & System Safety*, vol. 95, no. 11, pp. 1134 – 1141, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6V4T-50HP2R7-2/2/db6776aa9941d794db560396c92189d6
[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., ser. Information Science and Statistics. Springer, August 2006.
[9] R. Durbin and D. Rumelhart, "Products units: A computationally powerful and biologically plausible extension to backpropagation networks," *Neural Computation*, vol. 1, no. 1, pp. 133–142, 1989.
[10] A. Ismail and A. P. Engelbrecht, "Pruning product unit neural networks," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002, pp. 257–262.
[11] C. M. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–581, 1991.
[12] R. J. Howlett and L. C. Jain, *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*, 1st ed., ser. Studies in Fuzziness and Soft Computing. Physica-Verlag Heidelberg, 2001.
[13] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.
[14] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694–713, 1997.
[15] A. C. Martínez-Estudillo, F. J. Martínez-Estudillo, C. Hervás-Martínez, and N. García, "Evolutionary product unit based neural networks for regression," *Neural Networks*, vol. 19, no. 4, pp. 477–486, 2006.
[16] C. Hervás-Martínez and F. Martínez-Estudillo, "Logistic regression using covariates obtained by product-unit neural network models," *Pattern Recognition*, vol. 40, no. 1, pp. 52–64, 2007.
[17] P. A. Gutiérrez, F. López-Granados, J. M. Peña-Barragán, M. Jurado-Expósito, M. T. Gómez-Casero, and C. Hervás, "Mapping sunflower yield as affected by *Ridolfia segetum* patches and elevation by applying evolutionary product unit neural networks to remote sensed data," *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 122–132, 2008.
[18] C. Hervás-Martínez, F. J. Martínez-Estudillo, and M. Carbonero-Ruz, "Multilogistic regression by means of evolutionary product-unit neural networks," *Neural Networks*, vol. 21, no. 7, pp. 951–961, 2008.
[19] P. A. Gutiérrez, C. Hervás-Martínez, and F. J. Martínez-Estudillo, "Logistic regression by means of evolutionary radial basis function neural networks," *IEEE Transactions on Neural Networks*, 2011, accepted.
[20] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Data Management Systems. Morgan Kaufmann (Elsevier), 2005.
[21] S. le Cessie and J. van Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.
[22] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press, 1982.
[23] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 38, no. 2, pp. 337–374, 2000.
[24] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.
[25] F. J. Martínez-Estudillo, C. Hervás-Martínez, P. A. Gutiérrez, and A. C. Martínez-Estudillo, "Evolutionary product-unit neural networks classifiers," *Neurocomputing*, vol. 72, no. 1-2, pp. 548–561, 2008.
[26] P. A. Gutiérrez, C. Hervás-Martínez, and M. Lozano, "Designing multilayer perceptrons using a guided saw-tooth evolutionary programming algorithm," *Soft Computing*, vol. 14, no. 4, pp. 599–613, 2010.
[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.