Nonlinear Logistic Regression Model Based On Simplex Basis Function

Xia Hong, Hong Wei
Department of Computer Science
School of Mathematical and Physical Sciences
University of Reading
Reading, UK, RG6 6AY
Email: x.hong(h.wei)@reading.ac.uk

Junbin Gao
Discipline of Business Analytics
The University of Sydney Business School
The University of Sydney
NSW 2006, Australia
E-mail: junbin.gao@sydney.edu.au

***Abstract*—In this paper a novel nonlinear logistic regression model based on a simplex basis function neural network is introduced that outputs probability of categorical variables in response to multiple predictors. It is shown that since a linear combination of the simplex basis functions can be represented as a piecewise linear model, the proposed nonlinear logistic regression model retains the main advantage of linear logistic regression model, that is, allowing probabilistic interpretation of the data sets from an identified model. The associated estimation problem is treated based on the principle of maximum likelihood by alternating over two algorithms; the iteratively reweighted least squares algorithm for linear parameters, while the simplex basis functions are fixed; then nonlinear parameters in each simplex basis function are adapted in turn based on gradient descent of the negative likelihood. The proposed algorithm is then extended to estimation of nonlinear multinomial logistic model. Numerical experiments are initially carried out to illustrate the advantage of nonlinear logistic regression model versus its linear counterpart in terms of approximation capability. Then we apply the proposed method for a difficult computer vision example of land-cover real data set**

I. Introduction

Logistic regression is widely used in various fields including machine learning [1], [2], computer vision [3], medical diagnostics [4], and social science [5]. It is a popular and effective technique for classification tasks [6], [7]. Using a probabilistic framework, the logistic regression classifier predicts class posterior probabilities of input data samples. The logic regression can be generalized to the multi-class classifiers by employing a multinominal logistic function which takes into account the correlations among classes. The multinominal logistic regression is also called the maximum entropy model in the natural language processing community. Various learning algorithms have been developed for logistic regression and maximum entropy models, including iterative scaling [8], coordinate descent [9], trust region Newton method [10], etc.

Essentially the linear logistic regression models the probability of categorical variables in response to multiple predictors as a linear function. In comparison to other non-probabilistic classification methods such as support vector machine [11], the linear logistic regression model is generally worse in terms of classification performance. Hence it is natural to combine both to yield the so called nonlinear logistic regression models. For example training one or more conventional SVM classifiers followed by linear logistic regression models using SVM classifiers output as inputs [12]. However such models still lose some interpretation capability in comparison to linear logistic regression models since their parameters are not related directly to the input variables, rather to the associated nonlinear basis functions.

The use of linear functions to the system input is key to the interpretability of linear logistic regression models. Alternatively, a nonlinear system can be approximated by locally linear systems as piecewise linear systems. Various piecewise linear models exist such as lattice piecewise linear representation [13], hinging hyperplanes (HH) [14] and piecewise affine models [15]. Notably the hinging hyperplane (HH), which uses a hinge function as basis functions, is shown to be a powerful model representation for nonlinear systems since it is endowed with proven approximation capabilities to arbitrary nonlinear functions [14]. Recently a new simplex basis function model [16] has been introduced which can be viewed as a HH model and hence has the same approximation capability as HH.

In this paper we propose a novel nonlinear logistic regression model based on a simplex basis function neural network. It is analyzed that the proposed model nonlinear logistic regression model retains the main advantage of linear logistic regression model of allowing probabilistic interpretation of the data sets from an identified model, due to dual representation as a piecewise linear model. Since the model parameter estimation is a nonlinear estimation problem subject to constraints, the problem is based on an iteration of two algorithms based on the principle of maximum likelihood. The linear parameters are based on well known iteratively reweighted least squares algorithm, while the simplex basis functions are fixed. Then nonlinear parameters in each simplex basis function are adapted in turn based on gradient descent of the negative likelihood. It is shown that the algorithm is easily extendable to the multinominal logistic regression model, and the procedure is provided.

The remaining of this paper is organized as follows. Section II describes preliminaries of linear logistic model. Section III introduces the proposed nonlinear logistic model based on SBF

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

functions. The SBF's dual representation as a locally linear model input vector is analyzed for its interpretability. A novel model estimation algorithm based on maximum likelihood has been presented in Section IV. Section V introduces the estimation of nonlinear multinomial logistic model based on combining multiple binary classifiers. Numerical experiments are carried out in Section VI to illustrate the advantage of nonlinear logistic regression model versus its linear counterpart in terms of approximation capability. Finally the proposed method is applied to a difficult computer vision example of land-cover real data set.

II. Linear logistic model

Given a data sample $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \cdots x_m \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^m$ denoting the m-dimensional input vector, the logistic linear regression model calculates the class probability

$$P\big(t=1|\boldsymbol{x}\big) = y(\boldsymbol{x}) = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b) = \frac{1}{1+\exp\big(-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)\big)} \tag{1}$$

and $P\big(t=0|\boldsymbol{x}\big) = 1 - P\big(t=1|\boldsymbol{x}\big)$, where $t \in \{0,1\}$ is the class label denoting two class types. $\boldsymbol{w} = [w_1, \cdots, w_m]^{\mathrm{T}} \in \Re^m$ and $b \in \Re$ are the weights and a bias term. Note that the logistic sigmoid function is given as

$$\sigma(a) = \frac{1}{1+\exp(-a)}. \tag{2}$$

and its derivative can be conveniently expressed via itself

$$\frac{d\sigma}{da} = \sigma(1-\sigma). \tag{3}$$

The main advantage of linear logistic model is that it is able to extract probability information with respective to the input variables. Since t is binary, the logistic model leads to

$$\log\Big(\frac{P\big(t=1|\boldsymbol{x}\big)}{P\big(t=0|\boldsymbol{x}\big)}\Big) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \tag{4}$$

which models the log odds ratio between two classes as linear relationship to the system inputs. The corresponding parameters w_i have clear interpretation, i.e. for every 1-unit increase in x_i, the odds multiply by $\exp(w_i)$, hence the model is valuable to users who need to make sense of the data, e.g. in medical diagnosis application, or as a support to validate physical/biological hypothesis based on which data experiments are originally designed.

Consider a training data set $D_N = \{\boldsymbol{x}(k), t(k)\}_{k=1}^N$, in which $t(k)$ denotes the class type for each data sample $\boldsymbol{x}(k)$. We may estimate $\boldsymbol{w}, b$ by minimizing the negative log-likelihood:

$$\begin{aligned} &\min_{\boldsymbol{w},b} E(\boldsymbol{w}, b) \\ &= -\sum_{k=1}^{N} \Big(t(k)\log y(k) + \big[1-t(k)\big]\log\big[1-y(k)\big] \Big) \end{aligned} \tag{5}$$

where $y(k) = P\big(t=1|\boldsymbol{x}(k)\big)$. Unfortunately if the data exhibit high nonlinearities, the logistic linear regression model will struggle to achieve the best achievable classification performance or even fail, due to the inflexibility of the model structure, as illustrated using the following example.

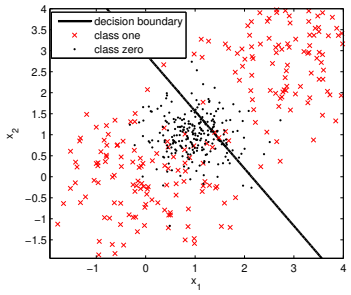


Fig. 1. Results of linear logistic model for Example 1. The model failed due to being unable to model the nonlinear classification boundary as required by the data set.

Example 1a: In this simulated example, 500 data samples $\boldsymbol{x}(k) \in \Re^2$, $k = 1, ...500$, are randomly generated. The first 250 data samples are from a mixture density given as $\frac{1}{2}N(\boldsymbol{0}, \boldsymbol{I}) + \frac{1}{2}N(3 \times \boldsymbol{1}, \boldsymbol{I})$, with output $t(k) = 0$. The second 250 data samples are drawn from $N([1,1]^{\mathrm{T}}, 0.5^2\boldsymbol{I})$, with output $t(k) = 1$, $k = 251, ..., 500$. The linear logistic model was obtained using (5). The data set was plotted in Figure 1 against the obtained linear decision boundary of $P\big(t=1|\boldsymbol{x}\big) = 0.5$, which clearly fails to separate two classes.

III. Nonlinear Logistic Model with Simplex Basis Function Neural Networks

In order to obtain a better classification capability to model data exhibiting high nonlinearities, we consider the nonlinear logistic model of

$$P\big(t=1|\boldsymbol{x}\big) = y(\boldsymbol{x}) = \frac{1}{1+\exp\big(-f(\boldsymbol{x})\big)} \tag{6}$$

and $P\big(t=0|\boldsymbol{x}\big) = 1 - P\big(t=1|\boldsymbol{x}\big)$, where $f(\bullet)$ is the unknown system mapping given by

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \theta_j \phi_j(\boldsymbol{x}) = [\boldsymbol{\phi}(\boldsymbol{x})]^{\mathrm{T}}\boldsymbol{\theta} \tag{7}$$

where θ_j are the model weights, and the regressors $\phi_j(\boldsymbol{x})$ is a predetermined basis function (with some adjustable internal parameters). M is the total number of regressors or model terms. $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 \cdots \theta_M \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{\phi} = \begin{bmatrix} \phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}) \cdots & \phi_M(\boldsymbol{x}) \end{bmatrix}^{\mathrm{T}}$.

Similarly the nonlinear logistic model leads to

$$\log\Big(\frac{P\big(t=1|\boldsymbol{x}\big)}{P\big(t=0|\boldsymbol{x}\big)}\Big) = [\boldsymbol{\phi}(\boldsymbol{x})]^{\mathrm{T}}\boldsymbol{\theta} \tag{8}$$

which models the log odds ratio between two classes as an linear relationship to the regressors $\phi_j(\boldsymbol{x})$. Note that logistic nonlinear regression model may retain some interpretation capability depending on the choice of $\phi_j(\boldsymbol{x})$. However it does not lead to linear relationship between its input $\boldsymbol{x}$ to log odds ratio, preventing the users to make sense of the significance of

system input $\boldsymbol{x}$ directly. On the other hand, any deterioration of classification performance the logistic linear regression model can offset its advantage of good interpretation. In this work we aim to offer a good compromise for both problems.

Consider $f(\boldsymbol{x})$ is specified as a simplex basis function (SBF) network model [16], in which the regressors $\phi_j(\boldsymbol{x})$ is a piecewise linear function, referred to as the simplex basis function (SBF) as

$$\phi_j(\boldsymbol{x};\boldsymbol{\mu}_j,\boldsymbol{c}_j)=\max\Big(0,1-\sum_{i=1}^{m}\mu_{i,j}|x_i-c_{i,j}|\Big) \tag{9}$$

in which $\boldsymbol{c}_j=\begin{bmatrix}c_{1,j}\ c_{2,j}\cdots c_{m,j}\end{bmatrix}^{\mathrm{T}}\in\Re^m$ is known as the center vector of the jth SBF unit which controls the location of jth SBF, and $\boldsymbol{\mu}_j=\begin{bmatrix}\mu_{1,j}\ \mu_{2,j}\cdots\mu_{m,j}\end{bmatrix}^{\mathrm{T}}\in\Re_+^m$ is the shape parameters vector that control the shape of jth SBF.

In the following, a special property is analyzed, which is referred to as the dual representation of SBF as a locally linear model (Lemma 1).

Lemma 1: The SBF model $f(\boldsymbol{x})$ can be represented as a piecewise locally linear model with respective to input $\boldsymbol{x}$ as

$$f(\boldsymbol{x})=\boldsymbol{\alpha}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{x}+\beta(\boldsymbol{x}) \tag{10}$$

where $\boldsymbol{\alpha}(\boldsymbol{x})$ and $\beta(\boldsymbol{x})$ are piecewise constants, with the properties

$$(i)\ \frac{\partial}{\partial\boldsymbol{x}}\boldsymbol{\alpha}(\boldsymbol{x})=\boldsymbol{0},\quad \frac{\partial}{\partial\boldsymbol{x}}\beta(\boldsymbol{x})=0, \tag{11}$$

$$(ii)\ \ \frac{\partial}{\partial\boldsymbol{x}}f(\boldsymbol{x})=\boldsymbol{\alpha}(\boldsymbol{x}) \tag{12}$$

Proof. Consider any given input vector $\boldsymbol{x}$, (7) can alternatively represented as

$$f(\boldsymbol{x})=\sum_{j\in S(\boldsymbol{x})}\theta_j\Big(1-\sum_{i=1}^{m}\mu_{i,j}|x_i-c_{i,j}|\Big) \tag{13}$$

where $S(\boldsymbol{x})\in[1,...,M]$ is index set of j, satisfying condition $\sum_{i=1}^{m}\mu_{i,j}|x_i-c_{i,j}|<1$. We have

$$\begin{aligned}f(\boldsymbol{x})&=\sum_{j\in S(\boldsymbol{x})}\theta_j-\sum_{j\in S(\boldsymbol{x})}\theta_j\sum_{i=1}^{m}\mu_{i,j}|x_i-c_{i,j}|\\&=\sum_{i=1}^{m}x_i\sum_{j\in S(\boldsymbol{x})}\theta_j\mu_{i,j}\mathrm{sign}\big(c_{i,j}-x_i\big)\\&+\sum_{j\in S(\boldsymbol{x})}\theta_j\big(1-\sum_{i=1}^{m}\mu_{i,j}c_{i,j}\mathrm{sign}\big(c_{i,j}-x_i\big)\big)\\&=\boldsymbol{\alpha}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{x}+\beta(\boldsymbol{x})\end{aligned} \tag{14}$$

where

$$\mathrm{sign}(s)=\begin{cases}1 & s>0\\0 & s=0\\-1 & s<0\end{cases} \tag{15}$$

and $\boldsymbol{\alpha}(\boldsymbol{x})=[\alpha_1(\boldsymbol{x}),...,\alpha_m(\boldsymbol{x})]^{\mathrm{T}}$, in which

$$\begin{aligned}\alpha_i(\boldsymbol{x})&=\sum_{j\in S(\boldsymbol{x})}\theta_j\mu_{i,j}\mathrm{sign}\big(c_{i,j}-x_i\big),i=1,...,m\\\beta(\boldsymbol{x})&=\sum_{j\in S(\boldsymbol{x})}\theta_j\big(1-\sum_{i=1}^{m}\mu_{i,j}c_{i,j}\mathrm{sign}\big(c_{i,j}-x_i\big)\big)\end{aligned} \tag{16}$$

So that we have $\frac{\partial}{\partial x_i}\alpha_i(\boldsymbol{x})=0$, $\frac{\partial}{\partial x_i}\beta(\boldsymbol{x})=0$. Hence

$$\frac{\partial}{\partial\boldsymbol{x}}f(\boldsymbol{x})=\boldsymbol{\alpha}(\boldsymbol{x}). \tag{17}$$

This concludes the proof. □

Clearly SBF's dual representation as a locally linear model input vector $\boldsymbol{x}$ is useful for extracting gradients information from an identified model in the similar way as a linear model, except that these are locally dependent. In our proposed nonlinear logistic model here, we have

$$\log\Big(\frac{P\big(t=1|\boldsymbol{x}\big)}{P\big(t=0|\boldsymbol{x}\big)}\Big)=\boldsymbol{\alpha}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{x}+\beta(\boldsymbol{x}) \tag{18}$$

Clearly the corresponding parameters $\alpha_i(\boldsymbol{x})$ have clear interpretation in the same way as linear logistic model. That is at a local point $\boldsymbol{x}$, for every 1-unit increase in x_i, the odds multiply by $\exp(\alpha_i(\boldsymbol{x}))$. Hence we can retain the advantage of linear logistic model, except that this interpretation is dependent on a local point $\boldsymbol{x}$, in contrast to linear logistic model which imposes the constraint of a global linear relationship which may not be true to the data with severe nonlinear characteristics.

IV. The Model Estimation Algorithm

Consider the parameter estimation of nonlinear logistic model from a training data set D_N, which is specified by a set of nonlinear parameters $\boldsymbol{\theta}$, $\boldsymbol{c}_j$ and $\boldsymbol{\mu}_j$ $(j=1,...M)$. Our proposed model estimation algorithm is an iterative and hybrid one with the aim of gaining computational advantage by exploiting the special model functional structure. Specifically this approach is based on a predetermined model size and the well known $k-$means clustering algorithm is applied to obtain initial simplex function centers, while all simplex functions, it is set $\mu_{i,j}=\mu$ initially. Since it is observed that if $\boldsymbol{c}_j$ and $\boldsymbol{\mu}_j$ are known, then $\boldsymbol{\phi}$ is fixed and the methods for linear logistic model algorithm can be applied for estimation of $\boldsymbol{\theta}$, hence the iteratively reweighted least squares (IRLS) is applied for $\boldsymbol{\theta}$. For $\boldsymbol{c}_j$ and $\boldsymbol{\mu}_j$, a new gradient descent algorithm is proposed also based on minimizing negative log likelihood cost. These two algorithms are alternatively applied until a final model is obtained as detailed below.

A. Iteratively reweighted least squares (IRLS) algorithm

The well known IRLS, which forms a component of the proposed identification algorithm, is presented for completeness. Consider that $\boldsymbol{c}_j,\boldsymbol{\mu}_j,\forall j$ are fixed. Over the training data

set D_N, we may estimate $\boldsymbol{\theta}$ by minimizing the negative log-likelihood of

$$E(\boldsymbol{\theta}) = -\sum_{k=1}^{N} \Big(t(k) \log y(k) + \big[1 - t(k)\big] \log \big[1 - y(k)\big] \Big) \tag{19}$$

Denote $\boldsymbol{t} = [t(1), ..., t(N)]^{\mathrm{T}}$ and $\boldsymbol{y} = [y(1), ..., y(N)]^{\mathrm{T}}$. By making use of (2) the derivative of $E(\boldsymbol{\theta})$ is given as

$$\nabla E(\boldsymbol{\theta}) = \sum_{k=1}^{N} (y(k) - t(k)) \boldsymbol{\phi}(k) \tag{20}$$

and the Hessian is given as

$$\boldsymbol{H} = \nabla\nabla E(\boldsymbol{\theta}) = \sum_{k=1}^{N} y(k)(1 - y(k)) \boldsymbol{\phi}(k) [\boldsymbol{\phi}(k)]^{\mathrm{T}} = \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Phi} \tag{21}$$

where $\boldsymbol{R} = \mathrm{diag}\{y(1)(1 - y(1)), ..., y(N)(1 - y(N))\}$. $\boldsymbol{\Phi}$ is the $N \times (m + 1)$ matrix whose kth row is $[\boldsymbol{\phi}(k)]^{\mathrm{T}}$.

The Newton-Raphson update for minimizing $E(\boldsymbol{\theta})$ is

$$\begin{aligned} \boldsymbol{\theta}^{\mathrm{new}} &= \boldsymbol{\theta}^{\mathrm{old}} - \boldsymbol{H}^{-1} \nabla E(\boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{old}} - \big(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Phi}\big)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{t}) \\ &= \big(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Phi}\big)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{z} \end{aligned} \tag{22}$$

with $\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{\theta}^{\mathrm{old}} - \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{t})$. (22) takes the form of weighted least squares algorithm, but $\boldsymbol{R}$ depends on $\boldsymbol{\theta}$, so it needs to iteratively calculated. Hence it is named as iteratively reweighted least squares (IRLS) [18]. The IRLS algorithm is presented in Algorithm 1. This algorithm is guaranteed to converge since $E(\boldsymbol{\theta})$ us a convex function with respective to $\boldsymbol{\theta}$. The total computational complexity Algorithm 1 is $O(M^3)$ for matrix inversion plus $O(MN)$ matrix vector multiplication, this has to be scaled by $Iter_1$, as $Iter1(O(M^3) + O(MN))$.

Algorithm 1 IRLS algorithm for logistic model estimation.

Require: D_N, $\boldsymbol{\Phi}$. IRLS iteration number $Iter_1$.

Ensure: For a fixed set of M regressors, to find $\boldsymbol{\theta}^{\mathrm{opt}} = \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$.

1: Set $\boldsymbol{\theta}^{\mathrm{old}}$=**0**.
2: **for** $l = 1 : Iter_1$ **do**
3: Calculate $\boldsymbol{y}$ by setting its elements to

$$y(k) = \frac{1}{1 + \exp\big(-[\boldsymbol{\phi}(\boldsymbol{x}(k))]^{\mathrm{T}} \boldsymbol{\theta}^{\mathrm{old}}\big)}, \quad k = 1, ..., N \tag{23}$$

4: Update $\boldsymbol{R} = \mathrm{diag}\{y(1)(1-y(1)), ..., y(N)(1-y(N))\}$.
5: Calculate $\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{\theta}^{\mathrm{old}} - \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{t})$.
6: Update $\boldsymbol{\theta}^{\mathrm{new}} = \big(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Phi}\big)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{z}$.
7: $\boldsymbol{\theta}^{\mathrm{old}} \leftarrow \boldsymbol{\theta}^{\mathrm{new}}$.
8: **end for**
9: Return $\boldsymbol{\theta}^{\mathrm{opt}}$ as $\boldsymbol{\theta}^{\mathrm{new}}$.

B. The proposed gradient descent algorithm based on maximum likelihood

Now consider estimating the $\boldsymbol{c}_j, \boldsymbol{\mu}_j$ associated with $\phi_j(\boldsymbol{x})$ while $\boldsymbol{\theta}$ and all other $\phi_i(\boldsymbol{x})$, $(i \neq j)$ are fixed. Write the negative log-likelihood of

$$J^{(j)}(\boldsymbol{c}_j, \boldsymbol{\mu}_j) = -\sum_{k=1}^{N} \Big(t(k) \log y(k) + \big[1 - t(k)\big] \log \big[1 - y(k)\big] \Big) \tag{24}$$

By making use of (2), we have

$$\begin{cases} \frac{\partial J^{(j)}}{\partial \mu_{i,j}} = \sum_{k=1}^{N} (y(k) - t(k)) \frac{\partial f(\boldsymbol{x}(k))}{\partial \mu_{i,j}}, & i = 1, ..., m \\ \frac{\partial J^{(j)}}{\partial c_{i,j}} = \sum_{k=1}^{N} (y(k) - t(k)) \frac{\partial f(\boldsymbol{x}(k))}{\partial c_{i,j}} & i = 1, ..., m \end{cases} \tag{25}$$

for $i = 1, ..., m$. Note that $f(\boldsymbol{x}(k))$ can be represented as

$$f(\boldsymbol{x}(k)) = \sum_{i \neq j} \theta_i \phi_i(\boldsymbol{x}(k)) + \theta_j \max\Big(0, 1 - \sum_{i=1}^{m} \mu_{i,j} |x_i - c_{i,j}|\Big) \tag{26}$$

in which the summation term is independent of $\boldsymbol{c}_j, \boldsymbol{\mu}_j$. We have

$$\frac{\partial f(\boldsymbol{x}(k))}{\partial \mu_{i,j}} = -\theta_j |x_i(k) - c_{i,j}| Id(k), \quad i = 1, ..., m \tag{27}$$

$$\frac{\partial f(\boldsymbol{x}(k))}{\partial c_{i,j}} = \theta_j \mu_{i,j} \mathrm{sign}(x_i(k) - c_{i,j}) Id(k), \quad i = 1, ..., m \tag{28}$$

where $Id(k)$ is an indication function given as

$$Id(k) = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} \mu_{i,j} |x_i(k) - c_{i,j}| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

Finally by taking into account the positive constraints for the shaping parameters $\boldsymbol{\mu}_j$, we propose the constrained normalized gradient descent algorithm, as expressed as

$$\begin{cases} c_{i,j} &= c_{i,j} - \eta \cdot \frac{\partial J^{(j)}}{\partial c_{i,j}} / \| \frac{\partial J^{(j)}}{\partial \boldsymbol{c}_j} \| \\ \tilde{\mu}_{i,j} &= \mu_{i,j} - \eta \cdot \frac{\partial J^{(j)}}{\partial \mu_{i,j}} / \| \frac{\partial J^{(j)}}{\partial \boldsymbol{\mu}_j} \| \\ \mu_{i,j} &= \max\big(0, \tilde{\mu}_{i,j}\big) \end{cases} \tag{30}$$

for $i = 1, ...m$, where $\eta > 0$ is a preset smaller learning rate.

Equation (30) is applied to M regressors ($j = 1, ..., M$) in turn while fixing other regressors, as presented in Algorithm 2. The computational complexity of the gradient descent algorithm is $O(N)$ for each regressor, hence the total computational complexity Algorithm 2 is $O(NM)$.

Algorithm 2 Maximal likelihood estimation using normalized gradient descent for simplex basis functions.

Require: D_N, M, $\boldsymbol{\theta}$, current $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, learning rate η.

Ensure: $\boldsymbol{c}_j, \boldsymbol{\mu}_j$ are adjusted to reduce negative log likelihood $J^{(j)}$, $j = 1, ..., M$.

1: **for** $j = 1, ..., M$ **do**
2: Update $\boldsymbol{c}_j$ and $\boldsymbol{\mu}_j$ using Equations (25)-(30).
3: **end for**
4: Return $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ..., M$.

C. Initialization of simplex basis functions

The proposed algorithm needs to be initialized with a predetermined model size M and an initial design matrix $\boldsymbol{\Phi}$, which is based on preset values of $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ...M$. Clustering algorithms can be used to initialize the centers $\boldsymbol{c}_j$, which accurately reflects the distribution of the data points. We preset $\boldsymbol{\mu}_j = \mu \mathbf{1}$, where $\mu > 0$ is a predetermined constant. From N data points $\boldsymbol{x}(k)$, $k = 1, \cdots, N$, the k-means algorithm [19] seeks to partition the data points in M disjoint subset S_j, each containing N_j data points, so as to minimize the sum-of-squares clustering function given by

$$J = \sum_{j=1}^{M} \sum_{\boldsymbol{x}(k) \in S_j} \|\boldsymbol{x}(k) - \boldsymbol{c}_j\|^2 \tag{31}$$

where $\in$ denotes belongs to. J is minimized when

$$\boldsymbol{c}_j = \frac{1}{N_j} \sum_{\boldsymbol{x}(k) \in S_j} \boldsymbol{x}(k) \tag{32}$$

D. Summary of the proposed hybrid estimation algorithm

Our proposed hybrid identification algorithm is summarized in Algorithm 3. The total computational complexity Algorithm 3 is therefore $Iter{*}O(NM){+}Iter{*}Iter1{*}O(M^3)$, meaning that the algorithm scales very well for large sized data set, and is very fast for a moderate sized M.

Algorithm 3 The proposed model estimation algorithm based on simplex basis function.

Require: D_N, M, μ, $Iter$.

Ensure: Maximal likelihood estimator are obtained for all parameters ($\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ...M$, and $\boldsymbol{\theta}$).

1: Apply the $k-$means clustering algorithm to initialize $\boldsymbol{c}_j$, $j = 1, ...M$. Set all $\mu_{i,j}$ as μ.
2: **for** $l = 1, ..., Iter$ **do**
3: Form $\boldsymbol{\Phi}$ from D_N based on $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ...M$.
4: Apply **Algorithm 1** to adjust $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^{\text{opt}}$.
5: Apply **Algorithm 2** to adjust $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ...M$ while $\boldsymbol{\theta}$ is fixed as a constant vector.
6: Calculate $J(Iter) = -\sum_{k=1}^{N} \Big(t(k) \log y(k) + \big[1 - t(k)\big] \log \big[1 - y(k)\big] \Big)$
7: **end for**
8: Return $\boldsymbol{c}_j, \boldsymbol{\mu}_j$, $j = 1, ...M$, and $\boldsymbol{\theta}$ and J.

V. Nonlinear Multinomial Logistic Model with Simplex Basis Function Neural Networks

Consider the multiclass classification problem where the training data set $D_N = \{\boldsymbol{x}(k), t(k)\}_{k=1}^{N}$, in which $t(k)$ denotes the class type for each data sample $\boldsymbol{x}(k)$, but $t(k) \in \{1, ..., L\}$ is the class label denoting $L > 2$ class types. A multiclass logistic model can be estimated based on the concept of cross-entropy error using softmax transformation of the SBF model. Given a data sample $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \cdots x_m \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^m$ denoting the m-dimensional input vector, the multiclass logistic linear regression model calculates the class probability

$$P\big(\boldsymbol{t} = \boldsymbol{t}_l | \boldsymbol{x}\big) = y(\boldsymbol{x}) = \frac{\exp(f^{(l)}(\boldsymbol{x}))}{\sum_j \exp(f^{(j)}(\boldsymbol{x}))} \tag{33}$$

where $\boldsymbol{t}_l = [t_1, ...t_L]^{\mathrm{T}}$, represent one-of-$L$ classes, is a binary vector with all elements zero, except element l, which equals one, and

$$f^{(l)}(\boldsymbol{x}) = \sum_{j=1}^{M} \theta_j^{(l)} \phi_j^{(l)}(\boldsymbol{x}) = [\boldsymbol{\phi}^{(l)}(\boldsymbol{x})]^{\mathrm{T}} \boldsymbol{\theta}^{(l)} \tag{34}$$

The negative log-likelihood function is named as cross-entropy function for the multiclass problem, given as

$$E = -\sum_{k=1}^{N} \sum_{l=1}^{L} t_{kl} \log y_{kl} \tag{35}$$

where $y_{kl} = P\big(\boldsymbol{t} = \boldsymbol{t}_l | \boldsymbol{x}(k)\big)$.

Since jointly minimizing E with respective to the set of all parameters in the case of multi-class problem concerning L SBF models is a difficult one, here we propose a pragmatical way to extend the above proposed binary nonlinear logistic model and estimation algorithm by combining $(L-1)$ nonlinear binary logistic models, which are obtained by using a pivot class e.g. $t = i$ against the other $(L-1)$ classes respectively. The proposed algorithm is repeated applied based on $(L-1)$ sub-data sets, each consisting of data samples belonging to two classes as $t(k) \in \{i, l\}$, for $l \neq i$, respectively. Specifically, we obtain $(L-1)$ nonlinear logistic models given as

$$P^{(i)}\big(t = l | \boldsymbol{x}\big) = \frac{1}{1 + \exp\big(-f^{(l)}(\boldsymbol{x})\big)}, \quad l \in \{1, ..., L\} \backslash i \tag{36}$$

with

$$f^{(l)}(\boldsymbol{x}) = \sum_{j=1}^{M} \theta_j^{(l)} \phi_j(\boldsymbol{x}; \boldsymbol{\mu}_j^{(l)}, \boldsymbol{c}_j^{(l)}) \tag{37}$$

where the superscript $^{(l)}$ labels the lth SBF model, $P^{(i)}\big(t = i | \boldsymbol{x}\big) = 1 - P^{(i)}\big(t = l | \boldsymbol{x}\big)$, hence we have

$$\log\Big(\frac{P^{(i)}\big(t = l | \boldsymbol{x}\big)}{P^{(i)}\big(t = i | \boldsymbol{x}\big)}\Big) = f^{(l)}(\boldsymbol{x}), \quad l \in \{1, ..., L\} \backslash i \tag{38}$$

Algorithm 4 Algorithm for nonlinear multinomial logistic model based on SBF using pivot i.

Require: D_N, M, μ, $Iter$.

Ensure: The SBF based multinomial logistic model is obtained by combing $(L-1)$ binary classifiers.

1: From D_N, construct $(L-1)$ sub-data sets $D_N^{(l)}$, $l \in \{1, ..., L\} \backslash i$, with $t(k) \in \{l, i\}$
2: **for** $l \in \{1, ..., L\} \backslash i$ **do**
3: Apply **Algorithm 4** to $D_N^{(l)}$.
4: **return** lth maximal likelihood estimators ($\boldsymbol{c}_j^{(l)}, \boldsymbol{\mu}_j^{(l)}$), $j = 1, ...M$, and $\boldsymbol{\theta}^{(l)}$.
5: **end for**
6: **return** Combining $(L-1)$ models using (39)-(40).

Using the fact that all K of the probabilities must sum to one, it can be verified that

$$P^{(i)}\big(t=l|\boldsymbol{x}\big)=\frac{\exp(f^{(l)}(\boldsymbol{x}))}{1+\sum_{l,l\neq i}\exp(f^{(l)}(\boldsymbol{x}))}, l\in\{1,...,L\}\backslash i \tag{39}$$

and

$$P^{(i)}\big(t=i|\boldsymbol{x}\big)=\frac{1}{1+\sum_{l,l\neq i}\exp(f^{(l)}(\boldsymbol{x}))} \tag{40}$$

Algorithm 4 summarizes the above estimation procedure for nonlinear multinomial logistic model based on pivot label i. However these are inconsistencies when different class is used as pivot. We proposed to average the results for each class being set as pivot i, to yield the final predicted class label for a new data sample $\boldsymbol{x}$, which is given as

$$\hat{l}(\boldsymbol{x})=\arg\max_{l}\{\frac{1}{L}\sum_{i=1}^{L}P^{(i)}\big(t=l|\boldsymbol{x}\big)\} \tag{41}$$

Note that only a total $\frac{L(L-1)}{2}$ pairwise binary classifiers need to trained.

VI. Experimental Studies

A. Comparison with linear logistic model

Example 1b is devoted to comparison with its linear counterpart, the linear logistic model Example 1a in Section II. This example is used to illustrate the advantage of the proposed model of being capable of modeling nonlinear decision boundary whereby the linear logistic model fails.

Example 1b: In order to demonstrate model properties and advantages of the proposed nonlinear logistic model in comparison with the linear logistic model, we revisit the same data set of *Example 1a* which is failed by linear logistic model. We preset the model size $M=4$, $\mu=0.2$, $Iter=100$ for the proposed model estimation algorithm based on simplex basis function to be applied. We also set the learning rate $\eta=0.005$, the iteration of IRLS as three. The model results are shown in Figure 2. The data set was plotted in Figure 2(a) against the obtained nonlinear decision boundary of $P\big(t=1|\boldsymbol{x}\big)=0.5$, which is able to separate two classes, as well as the final SBF centers $\boldsymbol{c}_j$, which are initialized by $k-$means clustering algorithm, and then adjusted together with the $\boldsymbol{\mu}_j$ using the proposed gradient descent algorithm. The evolution of the negative log likelihood of the proposed algorithm is plotted in Figure 2(b) showing it converges. Based on the obtained nonlinear logistic model, Figure 2(c)&(d) plot the predicted class probabilities for the data region and the Log odds ratio respectively. Since the model has a good classification performance. These results are meaningful for interpretation purpose, with much of the predicted class probabilities close to one when $t=0$. The local linearity of the model of log odds ratio is shown in Figure 1(d), which can be very useful in explaining the data. The overall comparison between linear logistic model and the proposed nonlinear logistic model for Example 1 over the training data set is summarized in Table I.

TABLE I
Comparison between linear logistic model and the proposed nonlinear logistic model for Example 1 over the training data set.

	Misclassification rate (%)	Negative log likelihood value
Linear logistic model	36	334.89
Proposed algorithm	9	119.23

B. Application to the land-cover image data set

Remotely sensed data are provided in six images as shown in Figure 3. Each of six bands is in the size of 211×356 measured images of colored images of red (R), green(G), blue (B), LIDAR first echo (FE), last echo (LE) and Near infrared (NIR). The ground-truth information are given in four classes of building, vegetation, car and ground, as shown in Figure 4(a), and the number of data samples are given in Table II, it can be seen that the class distribution is balanced except for car class, which is very imbalanced. We aim to construct a nonlinear SBF based multinomial logistic model to predict land covers as one of these four classes.

We start with generating input features using the six band image data imported as matrices. Feature selection is important in computer vision tasks. Known physical properties should be utilized in constructing discriminant features if possible. As such two new artificial images are generated [22]. The normalized difference vegetation index (NDVI) is defined as [22]

$$NDVI=\frac{NIR-R}{NIR+R} \tag{42}$$

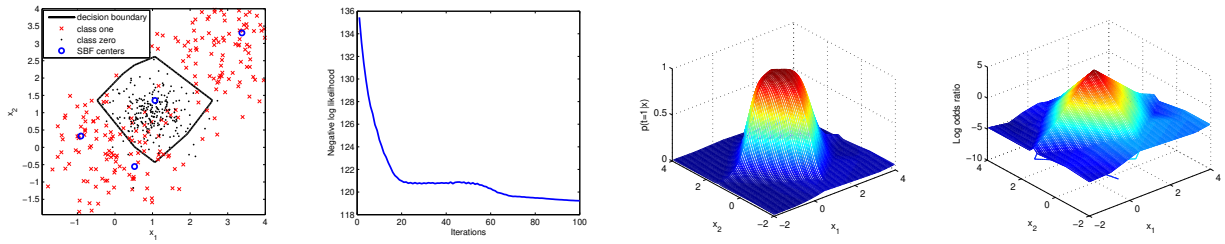
which is created from Red (R) and Near infrared (NIR) images, which is capable of distinguish vegetation from other objects. An additional derived feature image is the height difference (HD) defined as

$$HD=FE-LE \tag{43}$$

created from LIDAR first echo (FE) and last echo (LE), which is used to distinguish trees from other objects [22]. The six original images and the two derived images are normalized by dividing its maximum value in the original images. Denote each normalized pixel value of six original images and the two derived images as $x_1^{(0)},....x_8^{(0)}$ as R,G,B, FE, LE, NIR, NDVI and HD respectively.

In this computer vision task we should also include spatial contexture information among pixels, as neighborhood pixels are mostly likely belong to the same class. In order to incorporate spatial information, for each original $x_i^{(0)}$, $i=1,...,8$, we calculated statistical means as $x_i^{(j)}$, $j=1,...,9$, representing mean values of a series of squares with size $(2j+1)\times(2j+1)$, centered at $x_i^{(0)}$, respectively. There are a total number of 88 features in our experiment as

$$\boldsymbol{x}=[x_1^{(0)},...,x_8^{(0)},...,x_1^{(9)},...,x_8^{(9)}]^{\mathrm{T}}\in\Re^{88} \tag{44}$$



(a) (b) (c) (d)

Fig. 2. Results of nonlinear logistic model for Example 1.

We set the size of our training data samples as fully balanced, using the same number of training samples, 500 and 1000 data samples per class, respectively which are randomly drawn. In order to generate a four class nonlinear SBF based multinomial logistic model, 12 binary SBF classifiers are trained between all combinations of class pairs. We preset the model size $M = 20$, $\mu = 0.2$, $Iter = 100$, the learning rate $\eta = 0.005$, the iteration of IRLS is as three. The classification results of the whole image are reported in two cases as shown in III, since 94.7 % and 97.34% data points in the image are test data in two cases, except for car class the proportion of test data points are 55% and 9.5% respectively, which explains why car class seems to have best results. The average true positives for all classes are 87.5% and 89% when 500 and 1000 data points are used in training. The modeling results can be visualized in Figure 4(b) and (c), which shows slight improvement of using more training data samples at a higher computational cost. This example clearly demonstrate that the proposed model and learning algorithm is capable of extracting land cover type information form a small number of registered data points, and can be extended to other computer vision applications.

TABLE II
DESCRIPTION OF CLASSES FOR LAND-COVER IMAGE DATA SET.

Class	Data points	Percentage (%)
Building	21573	28.72%
Vegetation	24144	32.14%
Car	1105	1.47%
Ground	28294	37.67%
Total	75116	100%

VII. CONCLUSIONS

In this paper we have introduced a novel nonlinear logistic regression model based on a simplex basis function neural network. Since a linear combination of the simplex basis functions can be represented as a piecewise linear model, the proposed model nonlinear logistic regression model retains the main advantage of linear logistic regression model of not only predicting the probability of categorical variables in response to multiple predictors, but also the change of odd ratio with respect to the input variables, allowing probabilistic interpretation of the data sets from an identified model. Based on the principle of maximum likelihood, we proposed a composite estimation algorithm by iterating over two sub-algorithms (i) the iteratively reweighted least squares algorithm for linear parameters, while the simplex basis functions are fixed and (ii)the gradient descent algorithm for nonlinear parameters in each simplex basis function, which are adapted in turn based on minimizing negative likelihood. It is shown that the proposed algorithm is extendable to nonlinear multinomial logistic model. In order to demonstrate the effectiveness of the proposed approaches, numerical experiments are designed so as to illustrate the advantage of nonlinear logistic regression model versus its linear counterpart in terms of approximation capability and its application for multiclass classification using nonlinear multinomial logistic model based on real land-cover data set in computer vision.

REFERENCES

[1] W. S. Lee and B. Liu, "Learning with positive and unlabelled examples using weighted logistic regression," In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, USA, 2003.

[2] X. Hong and J. B. Gao, "Manifold optimization for nonnegative coefficient logistic regression", In *Proceedings of 2016 International Joint Conference on Neural Networks (IJCNN)*, pp1762-1766, Vancouver, Canada, 2016.

[3] H. Khurshid and M. F. Khan, "Segmentation and classification using logistic regression in remote sensing imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(1), pp224-231, 2015.

[4] S. C. Bagley, H. White, B. A. Golomb, "Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain," *Journal of Clinical Epidemiology*, 54, pp979-985, 2001.

[5] I. Theodossiou, "The effects of low-pay and unemployment of psychological well-being: a logistical regression approach," *Journal of Health Economics*, 17(1), pp85-104, 1998.

[6] A. Agresti, *Categorical Data Analysis*, New York: Wiley-Interscience, 2002.

[7] T. Hastie, R. Tishirani, and J. Friedman,*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, New York, 2001.

[8] J. Goodman, "Sequential conditional generalized iterative scaling," In *Proceedings of the 40th Annual Meeting of the Association of Computationnal Linguistics (ACL)*, pp9-16,2002.

[9] F.L Huang, C.J. Hsieh, K.W Chan, and C.J Lin, "Iterative Scaling and Coordinate Descent Methods for Maximum Entropy Models", *J. Mach. Learn. Res.* 11, pp815-848, 2010.

[10] C. J. Lin, R. C. Weng and S. S. Keerthi, "Trust region Newton method for large scale logistic regression", *J. Mach. Learn. Res.* 9, pp627-650, 2008.

[11] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

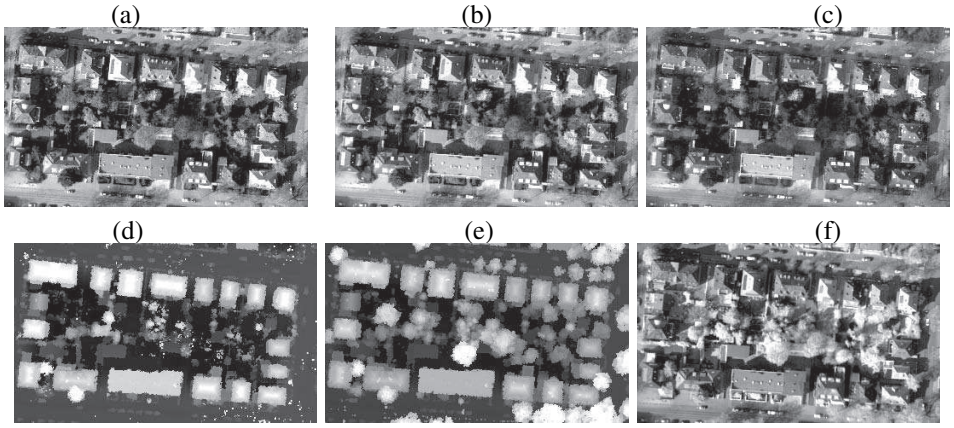


Fig. 3. Land cover image data set as six channels of images; (a) Red (b) Green (c) Blue (d) LIDAR first echo (e) LIDAR last echo and (f) Near infrared.

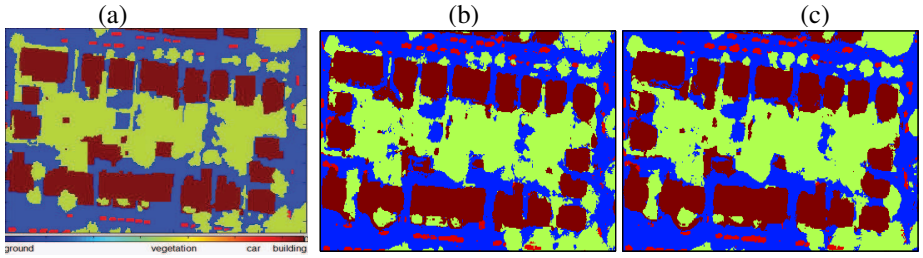


Fig. 4. Results of land cover image data set; (a) the ground truth of four class labels (ground, vegetation, car and building); (b) Predicted class labels using 500 data points per class for training and (c) Predicted class labels using 1000 data points per class for training.

TABLE III

RESULTS OF LAND COVER IMAGE DATA SET, WHERE (%) IS WITH RESPECT TO ACTUAL CLASS; (A) 500 DATA POINTS PER CLASS FOR TRAINING AND (B) 1000 DATA POINTS PER CLASS FOR TRAINING.

(a)

Actual Class	Predicted Class			
	Ground	Vegetation	Car	Building
Building	18909 (87.65%)	712 (3.3%)	165 (0.76%)	1767 (8.26%)
Vegetation	1082 (4.48%)	21490 (89.0%)	73 (0.3%)	1499(6.21%)
Car	0 (0%)	26 (0.63 %)	969 (92.22%)	110 (7.15%)
Ground	1924 (6.66%)	1813 (9.24%)	1151 (2.99%)	23406 (81.11%)

(b)

Predicted Class	Actual Class			
	Building	Vegetation	Car	Ground
Building	18160 (89.42%)	611 (2.83%)	0 (0%)	1733 (7.25%)
Vegetation	924 (3.83%)	21865 (90.6%)	68 (0.28%)	1287 (5.33%)
Car	0 (0%)	17 (1.54%)	1053 (95.29%)	35 (3.17%)
Ground	2109 (7.45%)	2578 (9.11%)	772 (2.73%)	22835 (80.71%)

[12] M. K. Elbashir, J. Wang and F. Wu, "A hybrid approach of support vector machines with logistic regression for β-turn prediction", In *Proceedings of 2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, Philadelphia, PA, USA, 4-7 Oct, 2012.

[13] J. Xu, T. J. J. van den Boom, B. D. Schutter and S. Wang, "Irredundant lattice representations of continunous piecewise affine functions", *Automatica* 70, pp109-120, 2016.

[14] L. Breiman, "Hinging Hyperplanes for regression, classification and function approximation", *IEEE Trans. on Information Theory* 39(3), pp999-1013, 1993.

[15] J. Roll, A. Bemporad and L. Ljung, "Identification of piecewise affine systems via mixed-integer programming", *Automatica* 40(1),pp37-50, 2004.

[16] J. Yu, S. Wang and L. Li, "Incremental Design of Simplex Basis Function Model for Dynamic System Identification," *IEEE Trans. on Neural Networks and Learning Systems*, In Press.

[17] R. S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewoods Cliffs and New Jersey, 2007.

[18] D. B. Rubin, "Iteratively reweighted least squares," In *Encyclopedia of Statistical Sciences*, Vol 4, ed. S. Kotz, N. L. Johnsone and C.B. Read, 272-275. New York: Wiley.

[19] S. Haykin, *Neural Networks and Learning Machines*. Pearson Education Inc, 2009.

[20] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[21] *http://www.fml.tuebingen.mpg.de/members/raetsch/benchmark*

[22] Y. Cao, H. Wei, H. Zhao and N. Li, "An effective approach for land-cover classification from airborne lidar fused with co-registered data," *International Journal of Remote Sensing*, vol. 33, no. 18, pp. 5927–5953, 2012.