Neurocomputing 73 (2010) 2799–2808

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An efficient multi-objective learning algorithm for RBF neural network

Illya Kokshenev *, Antonio Padua Braga

Universidade Federal de Minas Gerais, Depto. Engenharia Eletrônica Av. Antônio Carlos, 6.627 - Campus UFMG Pampulha 30, 161-970 Belo Horizonte, MG, Brazil

ARTICLE INFO

Available online 22 August 2010

Keywords:
Multi-objective learning
Radial-basis functions
Pareto-optimality
Model selection
Regularization

ABSTRACT

Most of modern multi-objective machine learning methods are based on evolutionary optimization algorithms. They are known to be global convergent, however, usually deliver nondeterministic results. In this work we propose the deterministic global solution to a multi-objective problem of supervised learning with the methodology of nonlinear programming. As the result, the proposed multi-objective algorithm performs a global search of Pareto-optimal hypotheses in the space of RBF networks, determining their weights and basis functions. In combination with the Akaike and Bayesian information criteria, the algorithm demonstrates a high generalization efficiency on several synthetic and real-world benchmark problems.

1. Introduction

Many tasks of intelligent data analysis are covered by the field of machine learning. As known, solutions to common problems of machine learning, such as pattern recognition, regression, and categorization (clustering) always result into trade-offs among several concurrent objectives of learning. For instance in supervised learning, the trade-off between the empirical risk (training error) and capacity of a hypotheses class (model complexity) is depicted by the paradigms of Statistical Learning Theory (SLT) [1] and the bias-variance dilemma [2], playing essential role in the performance of a learning machine. Namely, the principle of structural risk minimization (SRM) [3] states that the error and complexity must be minimized maintaining a certain balance in order to achieve a solution to the learning problem, characterized by good generalization properties.

The principle of SRM is usually implemented by means of error minimization while controlling the complexity of the model. Such approach is employed in many learning machines, such as neural networks with weight decay or pruning, regularization networks, and support vector machines (SVM). They minimize both error and complexity as a single loss function, whereas the point of balance is pre-determined by one or several hyperparameters (e.g., regularization and kernel parameters). Each choice of hyperparameters provides only a particular solution (learning hypothesis), which is not necessary efficient since not all choices of hyperparameters represent the trade-off between the error and complexity. In contrast, the principle of Pareto-optimality permits one to express the complete set of efficient solutions through the multi-criteria formulation of the learning problem. This approach led to a development of the multi-objective machine learning (MOML) [4].

A direct application of the Pareto-optimality principle to a general set of hypotheses usually results in non-convex problems, whose global solutions are required. Due to NP-complexity of such problems and difficulties of finding their solutions analytically, the arsenal of MOML methods went to the field of rapidly developing evolutionary multi-objective optimization (EMO) [5,6], as witnessed by the recent review on the subject [7]. In particular, most MOML algorithms (e.g., [8–12]) emerge from the genetic population-based approach. As an alternative, applications of nonlinear programming methods are demonstrated in [13–15], where the MOML problem of finding Pareto-optimal hypotheses in the domain of multilayer perceptrons (MLP) is approached with the so-called MOBJ algorithms.

The MOBJ algorithms are deterministic. However, they rely on the locally convergent optimization directly applied to generally non-convex problems, suffering from the problem of local minima. Hence, the Pareto-optimality is not guaranteed. On the other hand, the EMO algorithms are based on heuristics, providing the nondeterministic approximations of Pareto sets with populations of nondominated elements, which are unable to reach Pareto-optimality within a guaranteed time.

Despite of high capabilities of EMO, certain multi-objective problems can be efficiently solved in a deterministic way, taking advantages of nonlinear programming. In particular, the earlier proposed in [16] idea of decomposition of the multi-objective problem into a set of convex subproblems led to a development of the MOBJ algorithm for finding Pareto-optimal solutions within a small class of hypotheses of RBF networks. Such an approach allows to approximate Pareto sets arbitrary well with the numbers of exact solutions of convex subproblems.

In this work, we provide a deeper study of the previous results [16] and extend their application to larger classes of hypotheses.

* Corresponding author.
E-mail addresses: illya.kokshenev@gmail.com (I. Kokshenev), apbraga@cpdee.ufmg.br (A. Padua Braga).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.06.022

Specifically, we show the possibility of finding Pareto-optimal hypothesis within the class of RBF networks of arbitrary structures. The proposed MOBJ algorithm determines the weights, widths, and centers of the basis functions as well as their quantity. Also, a special attention is payed to the problem of selection of the final solution (model selection) from the wide spectrum of Pareto-optimal hypotheses.

2. Multi-objective view on supervised learning

Let Ω be the set of learning hypotheses and $\phi : \Omega \to \mathbb{R}^r$, $r \in \mathbb{N}$ be the vector-function of learning objectives. Without loss of generality, we assume that all $r \geq 2$ components of ϕ are aimed for minimization under Ω. When there exists such a hypothesis $f \in \Omega$ that simultaneously turns all components of ϕ into their global extrema, the solution to the minimization problem is, obviously, f. Otherwise, the solution to the multi-objective problem is the set

$$\mathcal{P}(\Omega) := \{f \in \Omega | \forall f' \neq f \in \Omega (f \prec f')\} \quad (1)$$

of nondominated hypothesis, also known as Pareto set. Here, for two hypothesis $f \in \Omega$ and $f' \in \Omega$ we denote $f \prec f'$ with the meaning "f strictly dominates f'". In our minimization setting, $f \prec f'$ is true *iff* $\phi(f') \neq \phi(f)$ and all components of the difference vector $\phi(f') - \phi(f)$ are non-negative. In other words, $f \prec f'$ is true when the hypothesis f is not worse than f', but also is better with respect to at least one of the objectives. Hence, the nondominated elements represent a set of solutions which cannot be improved any further, thus, they are optimal, i.e., Pareto-optimal. The Pareto set can be viewed as the lower bound of Ω under the strict partial order relation $\prec$, whereas its geometry can be studied in $\mathbb{R}^r$ from its image under ϕ, denoted as $\rho(\Omega) := \phi(\mathcal{P}(\Omega))$, also known as Pareto front.

In particular, we consider the case when Ω is the set of input–output mapping functions, corresponding to a certain class of neural networks. When $\phi_e : \Omega \to \mathbb{R}$ and $\phi_c : \Omega \to \mathbb{R}$ are the empirical risk and model complexity functionals, respectively, the bi-criteria minimization problem

$$\min_{f \in \Omega} \phi(f) = (\phi_e(f), \phi_c(f))^T \quad (2)$$

corresponds to the supervised learning in its multi-objective formulation, referred to as MOBJ [13]. When Ω is an uncountable set, the Pareto front $\rho(\Omega)$ is a non-increasing curve in $\mathbb{R}^2$ and for some arbitrary ϕ may contain non-convex intervals and discontinuities, as illustrated in Fig. 1.

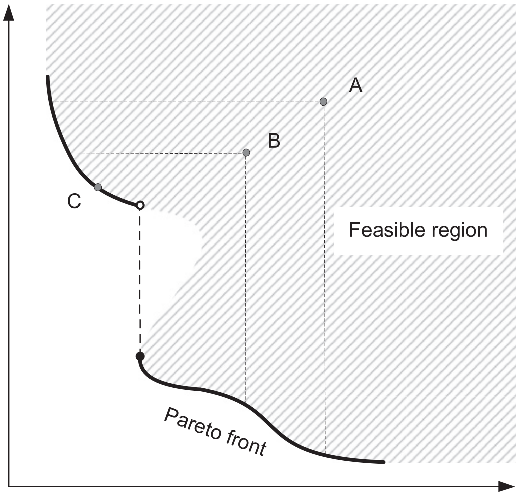

Fig. 1. Illustration of the Pareto optimality principle: the hypotheses A, B, and Pareto-optimal C are related as $C \prec B \prec A$.

The Pareto-set $\mathcal{P}(\Omega)$ usually contain infinite number of elements, equivalently efficient with respect to ϕ. Thus, it is required to make a decision towards the final hypothesis from $\mathcal{P}(\Omega)$, via application of a certain *posteriori* model selection criterion.

3. Approximations of the Pareto set

For generally non-convex objective functions, finding all Pareto-optimal hypotheses requires a global optimization, addressing the MOML to a class of NP-complete problems. Thus, approximate solutions are common in practice. In the evolutionary approach to MOML with genetic algorithms (GA) (e.g., [8,9]), the Pareto set $\mathcal{P}(\Omega)$ is approximated by a finite population of hypotheses which are getting closer to $\mathcal{P}(\Omega)$ after each evolution step. However, the elements of $\mathcal{P}(\Omega)$ can be analytically expressed as solutions of the single-objective optimization problems by means of the so-called scalarization techniques. For instance, the well-known $\varepsilon-$constraint [17] method determines the Pareto-optimal hypothesis of the MOBJ problem (2) as a solution of the constrained error minimization problem

$$\begin{aligned} &\min_{f \in \Omega} \quad \phi_e(f) \\ &\text{s.t.} \quad \phi_c(f) \leq \varepsilon_i. \end{aligned} \quad (3)$$

The set of solutions of (3), corresponding to a finite sequence of restriction parameters $(\varepsilon_i)_i$, is the subset of $\mathcal{P}(\Omega)$, and, thus, is its finite-set approximation $\tilde{\mathcal{P}}(\Omega) \subseteq \mathcal{P}(\Omega)$.

Another traditional scalarization method is the weighted-sum. Namely, when ϕ_e and ϕ_c are strictly convex on Ω, the minimization of their convex combination

$$\min_{f \in \Omega} \phi_e(f) + \lambda_i \phi_c(f) \quad (4)$$

is equivalent to (3) and draws Pareto-optimal elements from (2) (see e.g., [18, Chapter 3] and [19]).

Noteworthy, the commonly known learning schemes can be recognized from both (3) and (4). When ϕ_c is the measure of learning capacity of the model associated with f, the $\varepsilon-$constraint (3) solutions for the sequence of parameters $0 < \varepsilon_1 < \varepsilon_2 < \ldots < \infty$ minimize empirical risk ϕ_e within the structure $\emptyset \subset \Omega_1 \subset \Omega_2 \subset \ldots \subset \Omega$ of the nested subsets $\Omega_i = \{f \in \Omega | \phi_c(f) < \varepsilon_i\}$, explicitly implementing the principle of SRM. On the other hand, when (4) is minimized with $\phi_c(f)$, being a certain smoothness measure of f, one recovers a certain form of the regularization [20,21] in Ω. However, the latter requires a strict convexity of (4) for holding its equivalence to (2).

Usually, one is interested in Ω to be a class of universal approximators, e.g., neural networks of all possible topologies up to a certain size. Obviously, in this case ϕ_e contains multiple local minima and, consequently, is non-convex on Ω. Hence, due to the convexity limitations of the weighted-sum the Pareto set $\mathcal{P}(\Omega)$ cannot be entirely approximated with (4), whereas application of (3) requires globally convergent optimization procedures. Instead, following the earlier ideas from [16], we propose the decomposition of the problem domain by the union

$$\Omega = \bigcup_i \Omega_i.$$

Given that $\mathcal{P}(\mathcal{P}(\Omega)) = \mathcal{P}(\Omega)$, one can infer that $\mathcal{P}(A \cup B) = \mathcal{P}(\mathcal{P}(A) \cup \mathcal{P}(B))$ and thereby find the Pareto set from the relation

$$\mathcal{P}(\Omega) = \mathcal{P}\left(\bigcup_i \mathcal{P}(\Omega_i)\right). \quad (5)$$

When the subsets Ω_i are such that ϕ_e and ϕ_c are strictly convex under Ω_i, the elements of $\mathcal{P}(\Omega_i)$ in (5) can be efficiently found by solving the convex subproblems (4) or (3) on the corresponding subsets Ω_i.

4. The Pareto set of RBF networks

Consider the hypothesis class of all possible RBF neural networks

$$F := \left\{ f : \mathbb{R}^n \rightarrow \mathbb{R} | f(x) = \sum_{i=1}^{m} k_\sigma(x, c_i) w_i,\ m \in \mathbb{N} \right\}. \tag{6}$$

Here we treat the n-dimensional Gaussian neurons $k_\sigma(x,c_i) = \exp(-\|x - c_i\|^2 / 2\sigma^2)$ with their widths $\sigma \in \mathbb{R}^+$ and centers (prototypes) $c_i \in \mathbb{R}^n$. Initially, the class F contains all RBF networks, starting with the empty $m=0$ and finishing up by infinitely large topologies. Given the training set $\{(x_i,y_i)\}_{i=1}^N$ of N input–output pairs, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, we consider the empirical risk

$$\phi_e(f) := \sum_{i=1}^{N} (y_i - f(x_i))^2$$

of the squared loss. It is evidently that $\phi_e(f)$ is convex on the subset of hypotheses

$$F_{\sigma_j, C_m} := \{ f \in F | \sigma = \sigma_j, c_i \in C_m \},$$

corresponding to networks of fixed topology with m neurons, σ_j width and prototype matrix C_m. Since the weights w_i are the only free parameters within F_{σ_j, C_m}, f is linear with respect to them. Then, the empirical risk $\phi_e(f)$ is the quadratic surface in the space $\mathbb{R}^m$ of weights and thus is strictly convex on F_{σ_j, C_m}. Assuming also ϕ_c to be strictly convex on F_{σ_j, C_m}, the complete hypothesis set can be represented as the union

$$F = \bigcup_{\sigma_j \in \mathbb{R}^+} \bigcup_{m \in \mathbb{N}} \bigcup_{C_m \in \mathbb{R}^{m \times n}} F_{\sigma_j, C_m} \tag{7}$$

of subsets on which both objectives are convex.

Even though the problem of finding $\mathcal{P}(F)$ can be decomposed into a number of smaller and convex subproblems by their finite-set approximation, the direct application of (5) still requires to carry out the union over the infinite number of elements. Consequently, in order to take a practical advantage of (5), one needs a certain "refining" of the hypothesis set (7).

4.1. Refining sets

Consider the subset of hypotheses

$$F_{\sigma_j} := \bigcup_{m \in \mathbb{N}} \bigcup_{C_m \in \mathbb{R}^{m \times n}} F_{\sigma_j, C_m}$$

of all RBF networks of all structures with the same width parameter σ_j. Despite being a union of the infinite number of convex elements F_{σ_j, C_m}, the search for Pareto-optimal elements within F_{σ_j} can be significantly reduced by explicit elimination of non-efficient (dominated) elements.

Taking into consideration the known case of RBF interpolation [22], one can conclude that RBF networks of $m > N$ basis functions are not efficient, since the networks with the basis functions corresponding to all data patterns ($m=N$) are sufficient to interpolate the training set. Next, consider the sets of RBF networks with the number of basis functions $m < N$ less than data patterns. It is straightforward to conclude that such hypotheses sets are contained in the larger set of RBF networks of $m=N$ basis functions, since any smaller hidden layer can be viewed as a degeneracy of the case of N basis functions by a certain number of zero weights. Also, when there are only M distinct of N input patterns, the class of RBF networks corresponding to $m=M$ should be considered instead. Therefore, we reduce the search by passing from F_{σ_j} into a smaller subset

$$\tilde{F}_{\sigma_j} := \bigcup\nolimits_{C_M \in \mathbb{R}^{M \times n}} F_{\sigma_j, C_M}$$

while still maintaining the representability of the original problem domain, i.e.,

$$\mathcal{P}(F_{\sigma_j}) = \mathcal{P}(\tilde{F}_{\sigma_j}).$$

The next significant reduction can be made by approximation of C_M with the set of distinct input patterns

$$X = \{ x_i | \forall i,\ j = 1 \ldots N,\ i \neq j (x_j \neq x_i) \}.$$

It is obvious that a selection of the centers of basis functions far outside of the convex hull of data patterns is inefficient, leading to an increase in the model complexity without reduction in error. Moreover, from the regularization point of view [23] X are the centers associated with the optimal solution to (4) when ϕ_c is a certain regularizer[1] associated with k_σ. For a general convex ϕ_c, we state that a selection of centers from X will result in a representative subset of nondominated elements of F_{σ_j}, i.e.,

$$\mathcal{P}(F_{\sigma_j, X}) \subseteq \mathcal{P}(F_{\sigma_j}).$$

Finally, the application of decomposition (5) allows one to approximate the complete Pareto set $\mathcal{P}(F)$ with

$$\tilde{\mathcal{P}}(F) := \mathcal{P}\left(\bigcup_{\sigma_j} \mathcal{P}(F_{\sigma_j, X}) \right).$$

The the elements of $\mathcal{P}(F_{\sigma_j, X})$ can be therefore found by means of convex optimization. The values of σ_j can be arranged on a finite grid $(\sigma_j)_j$, whose size (number of elements) controls the approximation quality.

4.2. Complexity measure and L_1 constraint

In our previous work [16], the complexity measure

$$\phi_c(f) = \frac{\|w\|_1}{\sigma}$$

was suggested for Gaussian RBF networks, where $\|w\|_1 = \sum_i^m |w_i|$ is the 1-norm of the weight vector $w = (w_1, w_2, \ldots, w_m)^T$ of the RBF network and σ is the width of the basis functions.

Due to the evident strict convexity of both the $\phi_c(f)$ and $\phi_e(f)$ functions on $F_{\sigma_j, X}$, it is straightforward to show from (3) and (4) that the entire nondominated set $\mathcal{P}(F_{\sigma_j, X})$ can be exactly drawn from the solutions of

$$\begin{aligned} \min_{w \in \mathbb{R}^M} \quad & \|Y - Hw\|^2 \\ \text{s.t.} \quad & \|w\|_1 \leq \sigma_j \varepsilon, \end{aligned} \tag{8}$$

or its equivalent

$$\min_{w \in \mathbb{R}^M} L(w) = \|Y - Hw\|^2 + \lambda \|w\|_1,$$

also known as LASSO regression [24]. Here $Y = (y_1, y_2, \ldots, y_N)^T$ is the desired response vector from the training set and $H = \{ k_{\sigma_j}(x_i, c_l) \}$ is the $N \times M$ design matrix of the radial basis layer.

The LASSO is a form of linear regression with 1-norm regularizer. In contrast to the common ridge regression with the Euclidean norm, also known as Tikhonov's regularization [20], the solutions of LASSO are sparse. Geometrically, its behavior is

[1] Namely, $\phi_c(f) = \|Df\|^2$, where D is the linear differential operator such that k_σ is Green's function to $\tilde{D}D$ ($\tilde{D}$ is adjoint to D).

explained by the optimal solutions resting in the edges of the 1-norm restriction polytope. When the restriction strength increases resulting in smaller polytope, more weights are shrinked to zero. From the statistical point of view, the sparsity is due to a certain subset selection process [25], in which the most correlated regressors (columns of the design matrix H) are chosen while the others are shrinked.

4.3. Regularization path

The nondominated elements $\mathcal{P}(F_{\sigma_j,X})$, belong to a regularization path of the LASSO problem (8). As known, such path is a piecewise-linear curve in $\mathbb{R}^M$ [26] that can be entirely retrieved by the computationally efficient LARS algorithm [25].

For a given design matrix H and response vector Y, the LARS algorithm generates the sequence of vectors $(p_i \in \mathbb{R}^M)_{i=0}^{q}$, originating from $p_0=0$ until reaching the ordinary least squares (OLS) [27] solution $p_{ols}=(H^TH)^{-1}H^TY$. Then, the weights corresponding to the elements of $\mathcal{P}(F_{\sigma_j,X})$ are found exactly from the linear interpolation between the corresponding pairs of nodes $(p_0,p_1),(p_1,p_2),\ldots,(p_{q-1},p_{ols})$ of the path. Since ϕ_c grows linearly with $|w_i|$, while ϕ_e grows quadratically, the Pareto-front $\rho(F_{\sigma_j,X})$ is a piecewise-quadratic curve, whose segments are also uniquely determined by the results of the LARS algorithm. Hence, any point of the curve $\rho(F_{\sigma_j,X})$ can be found directly from a certain quadratic interpolation between a pair of nodes of the path without need of explicit computation of ϕ_e and ϕ_c. The whole picture of the LASSO regularization path and its corresponding Pareto front is shown schematically in Fig. 2.

4.4. Treating the bias parameter

The class of generalized RBF networks (6) was introduced without the bias parameter $(+b)$, whose importance should not be ignored. Since the complexity measure ϕ_c is based on smoothness, there is no place for the bias within the vector of weights in (8). Hence, the bias parameter must be estimated separately.

As known from linear regression, no bias needed for the centered data. Suppose the regression (8) is carried out for centered design matrix $\overline{H}=H-U\mu(H)$ and output vector $\overline{Y}=Y-U\mu(Y)$. Here, $\mu(H)$ is $1\times M$ the mean row vector of H, $\mu(Y)$ is the scalar mean of Y, and $U=(1,1,\ldots,1)^T$ is the $N\times 1$ vector of units. For arbitrary w, the squared loss in (8) can be rewritten as

$$\phi_e(f)=\|\overline{Y}-\overline{H}w\|^2=\|Y-Hw-U\mu(Y)+U\mu(H)w\|^2$$

that yields an expression for the bias parameter

$$b=\mu(Y)-\mu(H)w. \tag{9}$$

Consequently, when the centered matrices $\overline{H}$ and $\overline{Y}$ are passed to the LARS algorithm and the bias parameter is restored using (9), the obtained regularization path corresponds to the solutions of

$$\min_{w\in\mathbb{R}^M,b\in\mathbb{R}} \quad \|Y-Hw-b\|^2$$
$$\text{s.t.} \quad \|w\|_1\le\sigma_j\varepsilon.$$

4.5. MOBJ algorithm

Before combining the above results into an algorithm, several parameters should be introduced. First, one should define the grid for a search along σ, that can be an arbitrary finite discrete sequence $(\sigma_j)_j$. In particular, we suggest the grid $\sigma_j=(\sigma_{\min}^{0.5}+j/R_\sigma(\sigma_{\max}^{0.5}-\sigma_{\min}^{0.5})^2$ for $j=0\ldots R_\sigma-1$, where R_σ is the grid size and $\sigma_{\min}$ and $\sigma_{\max}$ determines its range. While R_σ determines only a resolution of the approximation, both $\sigma_{\min}$ and $\sigma_{\max}$ must be determined carefully, in order to cover the whole range of efficient solutions. We suggest to select $\sigma_{\min}$ and $\sigma_{\max}$ in accordance with the minimum and maximum distances between the patterns of X, respectively. However, other heuristics can also be used. For example, one can consider choosing range of width from the percentile distribution of distances between patterns in the training set.

The second useful parameter is the maximum complexity limit $\phi_c^{\max}$. The regularization path of LASSO regression spreads from the empty solution until reaching the most complex one, OLS solution, which is not regularized and may result in infinitely large wights. In order to prevent numerical instability and also reduce computational burden, we suggest to stop the LARS algorithm when the solutions $\phi_c(f)>\phi_c^{\max}$ are reached.

Aiming for application of the central idea of the decomposition (5) in a computationally efficient way, we are interested in a certain rendering of the elements of $\mathcal{P}(F_{\sigma_j,X})$, whose complexities ϕ_c lie at the common grid. Since the Pareto front corresponding to $\mathcal{P}(F_{\sigma_j,X})$ is a smooth monotonic curve which can be calculated at arbitrary point at almost no computational cost, it is sufficient to consider a linear grid in rage $[0,\phi_c^{\max}]$ with sufficiently large number of elements R_c. Then, the search for the nondominated elements $\tilde{\mathcal{P}}(F)$ within the solutions of convex subproblems can be carried out by choosing the hypotheses with the smallest value of ϕ_e among each set of equicomplex elements.

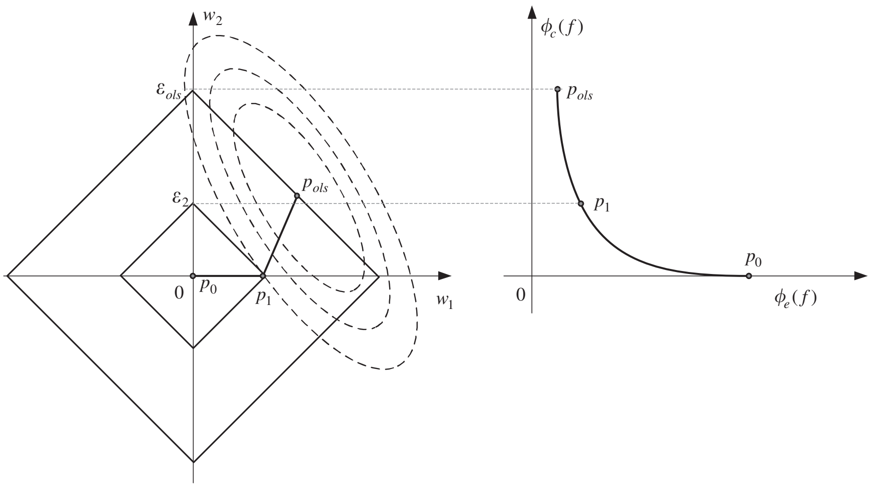

Fig. 2. The LASSO regularization path and its corresponding Pareto front in two-dimensional case.

Finally, the MOBJ algorithm can be stated as follows:

1. Initialization:
   (a) Given N training samples $\{x_i,y_i\}_{i=1}^{N}$, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, and the parameters $\phi_c^{\max}$, R_c, and R_σ.
   (b) Find the $(M \times n)$-matrix of distinct prototypes X from x_i, $(M \leq N)$.
   (c) Determine $\sigma_{\min}$ and $\sigma_{\max}$ using certain heuristic rules, e.g., $\sigma_{\min} = \min_{i \neq j}\|x_i - x_j\|$ and $\sigma_{\max} = \max_{i \neq j}\|x_i - x_j\|$.
2. For each $j = 0 \ldots R_\sigma - 1$, calculate the corresponding element of the grid σ_j and find $\mathcal{P}(F_{\sigma_j},X)$ by solving the corresponding subproblem:
   (a) Calculate $(N \times M)$-design matrix H for the N input patterns from the training set, using the centroid matrix X and width σ_j of the Gaussian RBF functions.
   (b) Find the sequence of nodes $P_j = \{p_k \in \mathbb{R}^M\}$ of the piecewise-linear LASSO regularization path, until reaching the complexity limit $\|p_k\|_1 > \sigma_j \phi_c^{\max}$. This can be done by passing the centered H and Y to the LARS algorithm.
   (c) Find the piecewise-quadratic Pareto front curve $\rho_j = \rho(F_{\sigma_j},X)$ calculating the parameters of its segments (since the curve bi-dimensional, it will take three scalar parameters per segment).
3. Combine solutions of the convex subproblems. For each $i = 0 \ldots R_c - 1$ and the corresponding complexity magnitudes $c_i = (\phi_c^{\max}/R_c)i$ on the grid do:
   (a) Find the hypothesis $f_{ij} \in \mathcal{P}(F_{\sigma_j},X)$ such that $\phi_c(f_{ij}) = c_i$ for all $j = 0 \ldots R_\sigma - 1$. This is simply done by linear interpolation between a certain pair of vectors in P_j.
   (b) Find the nondominated element $f_i = \operatorname{argmin}_{j=0 \ldots R_\sigma - 1} \phi_e(f_{ij})$. Here $\phi_e(f_{ij})$ can be found from the corresponding segment of the Pareto front ρ_j, without the need to compute $\|\overline{Y} - \overline{H}w\|^2$.

   The nondominated elements f_i, $i = 0 \ldots R_c - 1$ are the sought approximation of the Pareto set of the learning problem.
4. Restore the bias parameter using (9) for the resulted f_i elements and apply the model selection criterion to determine the final solution.

5. Model selection

The proposed MOBJ algorithm approximates the Pareto set with a wide complexity spectrum of RBF networks: from zero up to M basis functions (since the LASSO solutions are expected to be sparse at low complexities), with different width of Gaussian functions. Therefore, generalization properties of the final solution substantially depend on model selection criterion, which must determine the equilibrium point between the overfitting and underfitting parts of the approximated Pareto set.

When there is no *a priori* knowledge about which model to choose, the techniques of validation are commonly used. For example, the final hypothesis could be chosen in accordance with the minimum of the validation error (MVE) criterion

$$MVE(f) = \frac{1}{N_v}\sum_{i=1}^{N_v}(y_i - f(x_i))^2,$$

usually calculated on an additional validation data-set. However, the MVE criterion may suffer from a loss of representability against a wide range of models when the validation set is not large enough. Hence, we rely on model selection methods based on the information already available form the Pareto set: values of the training error, complexity, and model parameters (w_i and σ).

In [7], the heuristic Pareto-based selection approach [28] is discussed relying on selection of solutions from the corner point of the Pareto front, closest to the origin. Indeed, such an approach implies that selected solutions are likely to generalize well to unseen data. However, the selection (point of balance between the error and complexity) is highly dependent on the scale of the objective functions, whereas their adequate normalization may not be possible since the complexity objective is usually unbounded from above.

The application of another heuristics for model selection is demonstrated in [29], which emerges from the Pareto-ranking approach [30]. In case of selection by Pareto-ranking the scale of the objectives is irrelevant, however, the results are highly dependent on the layout and quantity of nondominated solutions of the certain rank. In our context, the domain of the problem is continuous and the proposed MOBJ algorithm can approximate the Pareto set with arbitrary amount of elements and their precision.

In the field of system identification and later statistics, the Akaike (AIC) [31] and Bayesian (BIC) [32] information criteria turn to be widely used model selection tools. Being independent on the way how the candidate solutions are obtained, such criteria establish the balance between the error and complexity of a model by matching the amount of unexplained information (held in the residual) with the number of degrees of freedom. In particular, one seeks the hypothesis minimizing

$$AIC(f) := 2\mathrm{df}(f) - 2\ln L(f),$$

or

$$BIC(f) := \ln(N)\mathrm{df}(f) - 2\ln L(f),$$

within the set of competitive solutions. Here df(f) is the effective number of degrees of freedom and $L(f)$ is the maximized likelihood function for the given model f. As suggested in [33], the second-order correction

$$AIC(f) = AIC(f) + \frac{2\mathrm{df}(f)(\mathrm{df}(f)+1)}{N - \mathrm{df}(f) - 1}$$

is necessary for preventing the AIC from overfitting on short data-sets (e.g., $N/\mathrm{df}(f) < 40$).

In the ridge regression model, the unbiased estimate of df(f) corresponds to the trace of the covariance matrix. In case of LASSO regression, the unbiased estimate $\hat{\mathrm{df}}(f) = m_r$ for df(f) is the number of non-zero weights m_r, as proved in [34]. Since the spectrum of Pareto-optimal RBF networks is expected to end up with the large models of $m_r = M$ basis functions and, thus, $N/\mathrm{df}(f) \approx 1$, the AIC should be always used with a correction.

For our setting, we introduce a generalized formula

$$\gamma(f,\tau) = \tau m_r - 2\ln L(f), \tag{10}$$

for the information criteria, and hereafter define $AIC(f) = \gamma(f, 2N/(N - m_r - 1))$ (correction is included) and $BIC(f) = \gamma(f, \ln(N))$. In (10), the estimation of likelihood $L(f)$ depends on particular settings of the learning problem and, thus, should be treated separately in regression and classification cases.

5.1. Regression case

In regression tasks, one usually assumes the additive noise model

$$y_i = f^0(x_i) + \varepsilon_i, \tag{11}$$

where $f^0(x_i)$ is the true unknown hypothesis (regression function) and y_i is the output observation from the training data-set, corresponding to x_i. Here, the noise component $\varepsilon_i \sim (0, \sigma_{ns}^2)$ is assumed to be random, independent, and normally distributed with the variance σ_{ns}^2. The maximized log-likelihood function

$$\ln L(f) = -\frac{1}{2} N \ln(2\pi\sigma_{ns}^2) - \frac{1}{2\sigma_{ns}^2}\sum_{i=1}^{N} e_i^2$$

is associated with a minimum of the squared sum of errors $e_i = y_i - f(x_i)$, for a given candidate hypothesis f.

The substitution of ln $L(f)$ in (10) yields the criterion

$$\gamma_{reg}(f,\tau) = \tau m_r + N\ln(2\pi\sigma_{ns}^2) + \frac{1}{\sigma_{ns}^2}\sum_{i=1}^{N} e_i^2,$$

or

$$\gamma_{reg}(f,\tau) = \tau\sigma_{ns}^2 m_r + \phi_e(f),$$

after multiplication by σ_{ns}^2 and removing the constant part.

The noise variance σ_{ns}^2 is usually unknown, however, its unbiased estimate $\hat{\sigma}_{ns}^2 = (1/(N-\mathrm{df}f))\sum_{i=1}^{N} e_i^2 = \phi_e(f)/(N-m_r)$ can be used. The information criterion based on this estimate is

$$\gamma'_{reg}(f,\tau) = \left(\frac{\tau m_r}{N-m_r} + 1\right)\phi_e(f).$$

5.2. Classification case

Consider the binary classification task with the labels $\{-1,+1\}$. Both the true hypothesis $f^0(x_i)$ and the observed output responses y_i now receive only binary values. Then, the additive noise component $\varepsilon_i = y_i - f^0(x_i)$ should be distributed within three possible states: $\varepsilon_i \in \{-2,0,2\}$. As seen, in this case ε_i cannot be independent, otherwise, the observed y_i would be able to receive the values $\{-3,-1,0,1,+3\}$. Consequently, the assumption (11) is not valid for classification. In other words, the noise is not additive.

In fact, the likelihood function of misclassification can be explicitly written as

$$L(f) = \eta^{E_c(f)}(1-\eta)^{N-E_c(f)},$$

where η is the probability of incorrectly labeled due to noise observations and $E_c(f)$ is the apparent number of misclassifications produced by the candidate hypothesis f. Substitution of the likelihood function into (10) provides the criterion

$$\gamma_{cls}(f,\tau) = \tau m_r - 2E_c(f)\ln(\eta) - 2(N-E_c(f))\ln(1-\eta).$$

Similarly to the above regression case, one passes from the unknown probability η to its unbiased estimate $\hat{\eta} = E_c(f)/(N-m_r)$, which, after simplification and removing a constant term, yields the criterion

$$\gamma'_{cls}(f,\tau) = \tau m_r + 2E_c(f)\ln\left(\frac{N-m_r}{E_c(f)} - 1\right).$$

6. Experiments

In this section we demonstrate the proposed MOBJ algorithm in practice on several benchmark problems. In its first part, we study behavior of the MOBJ algorithm on synthetic data-sets. The last part shows the results on real-world data-sets.

6.1. Twin spiral

The *twin spiral* is the classical two-dimensional benchmark problem for learning machines [35]. The goal of this experiment is to empirically qualify candidate solutions and their properties generated by the MOBJ algorithm. The data-set consists of 194 training patterns of the two highly non-linearly separable classes, each containing 97 points. The fragment of the Pareto front rendered by the MOBJ algorithm is shown in Fig. 3 for $\sigma = 0.12\ldots13.0$, with the widths resolution $R_\sigma = 150$ and up to complexity $\phi_c^{\max} = 500$. The minima of AIC and BIC are pointed out the same final model (Fig. 4). The final solution is compared with the overfitting and underfitting solutions manually selected from the Pareto set. The numerical results are shown in Table 1, where the column m_r represents a number of the basis functions

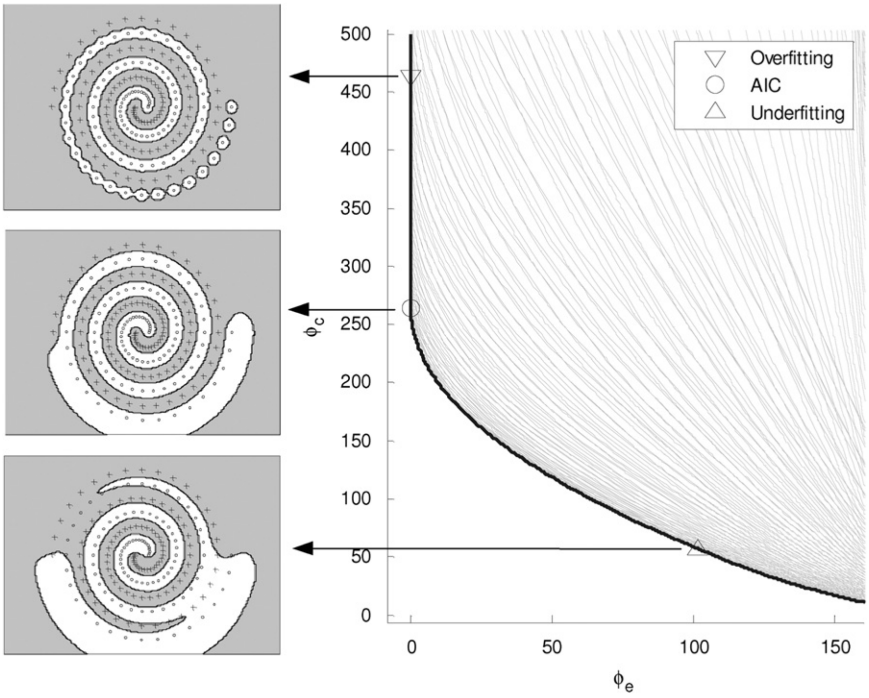

Fig. 3. *Twin spiral*: the minimum AIC solution compared with two other Pareto-optimal solutions.

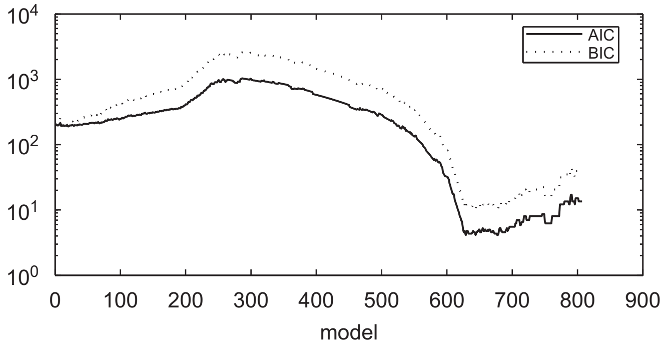

Fig. 4. *Twin spiral*: the AIC and BIC plots along the Pareto-optimal set.

Table 1
Twin-spiral benchmark results.

Solution	Properties				RMSE
	m_r	ϕ_c	σ	$\lVert w \rVert_1$	
AIC, BIC	162	263.8	0.558	147.3	0.0159
Underfitting	157	58.6	0.558	32.7	0.7171
Overfitting	187	468.8	0.377	176.8	0.0004

Table 2
Results for the *sinc* regression benchmark: test NRMSE × 10^2 (mean) and its standard deviation (std.).

σ_{ns}	Method	$N=50$		$N=100$		$N=200$	
		Mean	Std.	Mean	Std.	Mean	Std.
0.1	MOBJ-AIC	2.75	0.74	1.86	0.63	1.26	0.37
	MOBJ-BIC	4.06	2.58	**1.8**	0.56	1.31	0.38
	MOBJ-MVE	**2.56**	0.68	1.83	0.55	**1.2**	0.33
	ROFS-GCV	8.22	3.84	6.32	2.91	4.75	1.86
	ROFS-BIC	8.03	4.07	4.73	3.61	1.43	0.4
0.2	MOBJ-AIC	4.21	1.41	2.89	1.05	1.97	0.63
	MOBJ-BIC	5.07	2.62	**2.76**	0.77	1.95	0.55
	MOBJ-MVE	**3.99**	1.15	**2.76**	0.86	**1.8**	0.51
	ROFS-GCV	11.1	3.01	9.62	1.84	8.21	1.2
	ROFS-BIC	10.8	3.42	7.1	3.46	2.82	1.75
0.4	MOBJ-AIC	5.68	1.87	3.8	1.31	2.54	1.01
	MOBJ-BIC	5.81	2.37	**3.53**	1.14	2.40	0.77
	MOBJ-MVE	**5.20**	1.75	3.55	1.10	**2.30**	0.68
	ROFS-GCV	11.9	2.26	10.4	1.26	9.36	0.86
	ROFS-BIC	11.6	2.80	9.01	2.93	5.03	2.64

(size of the RBF network); ϕ_c is the corresponding complexity, σ is the width of Gaussian functions, and $\lVert w \rVert_1$ is the "size" of the weights. The root mean squared error (RMSE) is calculated for the training set.

As seen, the final solutions indicated by both the AIC and BIC coincide and achieve high generalization while separating all 194 patterns correctly with the large margin.

6.2. Noised sinc regression

The idea of this experiment is to demonstrate a behavior of the MOBJ algorithm with different model selection criteria on several data-set sizes and under noise conditions.

In the experiment, we consider the non-overlapping training and validation sets of the same size N uniformly sampled within the range $x \in (0,4\pi]$ from

$$y_i = \frac{\sin(\pi x_i)}{\pi x_i} + \varepsilon_i. \tag{12}$$

Here $\varepsilon_i \sim (0,\sigma_n^2 s)$ is the independent, normally distributed noise disturbance.

For each combination of tree data-set sizes $N \in \{50,100,200\}$ and three noise variances $\sigma_{ns} \in \{0.1,0.2,0.4\}$, we have generated 100 training and validation sets corresponding to distinct noise realizations. The normalized root mean squared error (NRMSE) of the final solutions was evaluated against the test set of 1000 samples taken without noise.

For comparison, we choose the regularized orthogonal forward selection (ROFS) algorithm [36]. Although the ROFS is based on the single-objective least-squares method, it acts similar to the MOBJ algorithm: model weights are determined by the forward selection, width and regularization parameters are determined by the grid search in accordance with the model selection criterion.

In the MOBJ algorithm, the settings $R_\sigma = 100$, $R_c = 1000$ and $\phi_c^{\max} = 150$ were used for all tests. The ranges of widths estimated from the data-sets were approximately [0.05,6].

In our experiments, we compare the MOBJ algorithm, endowed with the criteria AIC, BIC and MVE, to the solutions found by the ROFS with BIC and generalized cross validation (GCV). The validation sets were only used in the MVE criterion (MOBJ-MVE), but only the training sets were required for the rest of the solutions (MOBJ-AIC, MOBJ-BIC, ROFS-BIC and ROFS-GCV).

In Table 2, averaged over 100 noise realizations, the values of regression performance and their dispersion measures are shown for all data-set sizes and noise variances (best values are bold). In Fig. 5, we demonstrate the Pareto fronts and the final solutions associated with some particular noise realizations, corresponding to $\sigma_{ns} = 0.1$ and sample sizes $N=50$ and $N=100$.

6.3. Wisconsin breast cancer

Another well-known benchmark is based on real data obtained the from microscopic examination of patients. The Wisconsin breast cancer data-set [37] contains 699 patterns describing via 9 real-valued attributes the benign and malignant (cancer) tissue classes. In order to maintain the comparability and reproductivity of the results, we use the data-set partitions from the Proben1 benchmark set [38] selecting 350 training, 175 validation, and 174 test samples.

The classification results were obtained for each one of three different data partitions: *cancer1*, *cancer2*, and *cancer3*. In the experiment, the parameter values $\sigma = 0.05 \ldots 2.25$, $R_\sigma = 100$, $\phi_c^{\max} = 150$, and $R_c = 500$ were used. Correspondingly, 251, 247, and 239 distinct prototypes were found from the training sets. The results are shown in Table 3, where the classification accuracies measured are presented by the numbers of false-positive (F_p) and false-negative (F_n) classifications of the test sets, along with the total correct classification rates.

6.4. Abalone data-set

The *abalone* data-set represents the benchmark problem of mid-to-large scale (4177 patterns) available from [37]. The problem consists in prediction of the age of abalone from its 8 physical measurements (7 scalar, and 1 categorial). The age of abalone is represented by integers from 1 to 29, therefore the problem can be viewed as the regression task.

In this experiment, we are going to compare the solutions found by the MOBJ algorithm with the corresponding benchmark results of the SVM with Gaussian RBF kernel in [39]. For comparability of the results, we reproduce the benchmark

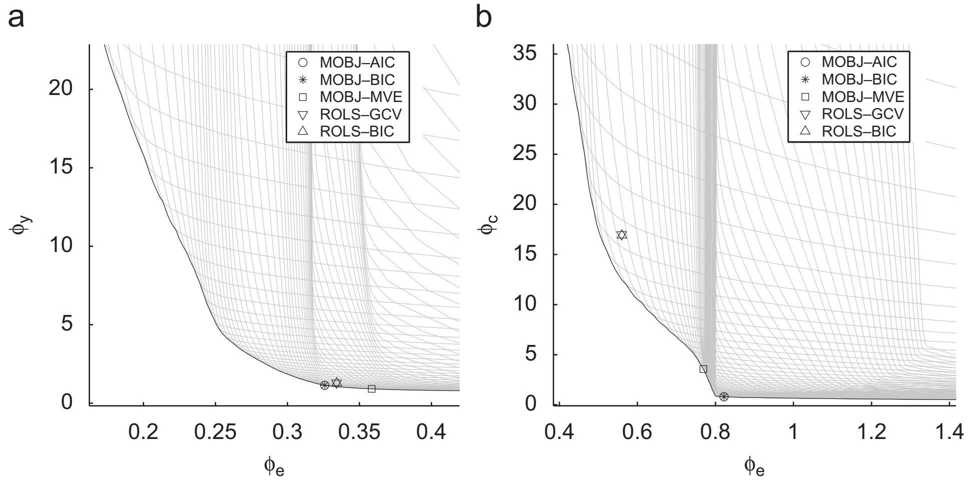

Fig. 5. The fragments of the Pareto fronts from the *sinc* regression experiment with $\sigma_{ns}=0.1$ and training sets of 50 (a) and 100 (b) samples.

Table 3
Wisconsin breast cancer benchmark results.

Part.	Method	Properties				Accuracy		
		m_r	ϕ_c	σ	$\Vert w \Vert_1$	F_p	F_n	Total (%)
cancer 1	MOBJ-AIC	3	2.71	0.900	2.44	4	0	97.8
	MOBJ-BIC	1	2.13	1.024	2.19	7	0	96.0
	MOBJ-MVE	72	149.7	0.312	46.71	0	3	**98.3**
	ROFS-GCV	132	4400	0.05	220	0	50	71.3
	ROFS-BIC	141	2139	0.07	165.3	0	26	85.0
cancer 2	MOBJ-AIC,BIC	2	2.62	0.87	2.29	2	5	**96.0**
	MOBJ-MVE	16	6.43	0.74	4.758	3	4	**96.0**
	ROFS-GCV	166	1250	0.1	131.6	0	27	84.5
	ROFS-BIC	158	1323	0.1	132.31	0	27	84.5
cancer 3	MOBJ-AIC	3	2.83	0.85	2.41	5	2	**96.0**
	MOBJ-BIC	2	2.18	1.02	2.24	8	0	95.4
	MOBJ-MVE	10	3.75	0.81	3.0333	5	2	**96.0**
	ROFS-GCV	154	1379	0.1	138	0	24	86.2
	ROFS-GCV	153	1379	0.1	138	0	24	86.2

Table 4
Abalone data-set results: median values of solution parameters and test RMSE.

Method	Parameters				Train RMSE	Test RMSE
	m_r	ϕ_c	σ	$\Vert w \Vert_1$		Median (mean)
MOBJ-AIC	158	3280	0.144	427	3.94	4.49 (4.55)
MOBJ-BIC	52	882	0.245	232	4.37	4.62 (4.62)
MOBJ-MVE	130.5	2528	0.156	396	4.01	**4.47** (**4.50**)
SVM [39]	–	–	–	–	–	4.48 (4.51)

settings from [39] and use exactly the same 100 data-set partitions of the Abalone data-set, available at http://www.ci.tuwien.ac.at/~meyer/benchdata. Each partition provides 90% of samples for training and 10% for test. Again, we consider three model selection criteria within the MOBJ algorithm: AIC, BIC, and MVE. Following the benchmark settings described in [39], the validation criterion is evaluated on $\frac{1}{3}$ of the training data after training the model on the rest of $\frac{2}{3}$ of samples. The resolution parameters of the MOBJ were set to $R_\sigma=30$ and $R_c=500$. The range of width estimated from the training set was approximately [0.003, 1.3].

The median average values of solution parameters, training and test errors obtained from the MOBJ algorithm are presented in Table 4 in comparison with the corresponding test error measurements of SVM [39]. Since [39] also provides mean values of the test error, we additionally include corresponding mean averages in the parenthesis.

7. Discussion

As demonstrated by the `twin spiral` experiment (Fig. 3), the MOBJ algorithm generates an extremely broad spectrum of solutions, however, the information criteria are capable to indicate solutions with the best generalization properties.

As seen from Fig. 5 of the *sinc* regression benchmark, the irregularities of the Pareto-fronts emerge under conditions of the noised data-sets. The analysis of the randomized experiments (Table 2) reveals the overall tendencies of utilized model selection criteria in the proposed algorithm. The solutions of the MOBJ-MVE are the most stable and outperform all others, in most cases. This was expected, since the validation sets were independently generated to be representative, and, thus, they provide unbiased estimates of true generalization errors. The MOBJ-AIC solutions are remarkably stable and precise even for smaller data-sets, while the MOBJ-BIC solutions gain their stability and performance on larger data-sets. Under almost all benchmark conditions, the MOBJ algorithm endowed with the information criteria demonstrates a competitive performance and good stability. In the situation, when a half of the available data-set is used for training instead of validation, the AIC and BIC demonstrate their great advantage over MVE (e.g., the results of MOBJ-AIC and MOBJ-BIC for $N=100$ should be compared with the results of MOBJ-MVE for $N=50$, since the rest 50 of 100 samples are used for validation).

On the real-world Wisconsin breast cancer benchmark (Table 3), the MOBJ-AIC and MOBJ-BIC show good classification rates and also compete with MOBJ-MVE. Moreover, the MVE solutions are characterized with much higher complexities and numbers of basis functions, whereas the AIC and BIC provide extremely small solutions with only 1–3 basis functions.

The ROFS algorithm demonstrated in all cases inferior performance and, usually, oversized solutions. A comparison with the classical SVM implementation [40], benchmarked [39] on the *abalone* data-set, also show slightly superior generalization performance of the proposed MOBJ algorithm. According to [39], the MOBJ-MVE solutions were selected by the same procedure used for selection of hyperparameters of SVM, which involves retraining by the cross-validation scheme. Both the mean and median performances rank MOBJ-MVE at the first place. On the other hand, the MOBJ-AIC showed performance close to MOBJ-MVE, while its application does not require any re-training, and, thus, reduce overall computational costs.

The selection of SVM hyperparameters by means of the grid search (at least in two dimensions) and k-fold cross validation is a common practice. The proposed MOBJ algorithm, viewed from this perspective, naturally incorporates two-dimensional grid search, however, the model selection criterion, in contrast to the SVM case, is applied only to the nondominated elements of the grid. The search over nondominated elements instead of all, corresponds to the uncertainly reduction and also provides significant reduction of the computational costs when the cross-validation is used.

As known, the LARS algorithm requires $\mathcal{O}(N^3)$ [34] operations for obtaining the complete regularization path, which is too high for applications to large data-sets, say, $N > 5000$. Noteworthy, same complexity order is required for a single fit by least-squares. As known from the Gaussian process theory, both the least squares and LASSO regressions cannot be solved exactly in a time shorter than $\mathcal{O}(N^3)$. Thus, switching to approximate but faster algorithms might be a good solution on practice. Indeed, in the context of the MOBJ algorithm, its computational complexity can also be controlled by cutting unnecessary parts of regularization pathes out by choosing a proper limit $\phi_c^{\max}$, which determines the maximum size of the model $m_r^{\max}$. After reaching $m_r^{\max}$ number of regression parameters, the LARS procedure stops at $\mathcal{O}(m_r^{\max}N^2)$ time. Aiming at generalization to a large data-set, one should expect $m_r/N \ll 1$ and consequently, the appropriate choice of $\phi_c^{\max}$ reduces the computational time $\mathcal{O}(N^2)$. Also, when the training patterns of a large data-set are dense, the reduction of their distinct set by means of clustering also implies the reduction of time to $\mathcal{O}(N^2)$, when the number of clusters M is small, i.e., $M/N \ll 1$.

While the resolution parameter R_σ directly determines how many regularization pathes will be constructed, the complexity resolution R_c affects mainly the number of evaluations of the model selection criterion. In case when the AIC or BIC are employed, the choice of R_c has no strong influence to the overall time of calculation. For instance, the MOBJ algorithm and all three model selection criteria took below one minute for the *cancer1* data-set and about 20 minutes for the *abalone* data-set, implemented in MATLAB on conventional desktop PC. Note, that the latter experiment involved MVE in the cross-validation scheme.

In contrast with the reported in [9,29,7] results of the evolutionary MOML algorithms, the Pareto front approximations of the MOBJ algorithm consist of smooth curves instead of populations of scattered points. Such approach to the multi-objective problem overcomes the difficulties of the uniform distribution of solutions along the Pareto front, usually occurring in population-based EMO algorithms. Also, it is possible to control the approximation quality directly by increasing the resolution R_σ. Moreover, with slight modifications of the proposed algorithm one can increase the approximation quality progressively in the neighborhood of the region of interest, or for the entire Pareto front.

8. Conclusion

The analytical approach to the multi-objective problem shows the possibilities of its global and deterministic solution by means of decomposition into convex elements. In contrast to nondeterministic evolutionary MOML algorithms, the proposed MOBJ algorithm is capable to approximate the Pareto set arbitrary well within a guaranteed time. Despite of the algorithmic similarity to previous [16], the new MOBJ algorithm performs an extremely broader search within the spaces of RBF networks while efficiently rendering a wide spectrum of candidate solutions.

The MOBJ algorithm, endowed with the information criteria for model selection, demonstrates the possibility of reaching good generalizations without time- and data-consuming validation steps. The parameters of the algorithm only determine the quality of Pareto set approximation with the resolution and range of the search, not affecting the generalization properties of final solutions.

According to a high performance and stability of the solutions demonstrated by the proposed algorithm on the synthetic and real-world benchmarks, we propose the algorithm as effective and, therefore, competitive MOML tool for supervised learning.

Acknowledgements

This research is financially supported by the National Council of Scientific and Technological Development (CNPq) of Brazil.

References

[1] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, New York, 1995.

[2] S. Geman, E. Bienenstock, R. Doursat, Neural network and the bias/variance dilemma, Neural Computation 4 (1992) 1–58.

[3] V.N. Vapnik, A.Y. Chervonenkis, Theory of Pattern Recognition, Nauka, Moscow, 1974 (in Russian).

[4] Y. Jin (Ed.), Multi-Objective Machine Learning, Series: Studies in Computational Intelligence, vol. 16, Springer Verlag, Heidelberg, 2006.

[5] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation 3 (1) (1995) 1–16.

[6] T. Hanne, Global multiobjective optimization using evolutionary algorithms, Journal of Heuristics 6 (3) (2000) 347–360.

[7] Y. Jin, B. Sendhoff, Pareto-based multi-objective machine learning: an overview and case studies, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38 (3) (2008) 397–415.

[8] G. Liu, V. Kadirkamanathan, Learning with multi-objective criteria, in: Fourth International Conference on Artificial Neural Networks, 1995, pp. 53–58.

[9] Y. Jin, T. Okabe, B. Sendhoff, Neural network regularization and ensembling using multi-objective evolutionary algorithms, Evolutionary Computation, CEC2004 1 (2004) 1–8.

[10] G.G. Yen, Multi-objective evolutionary algorithm for radial basis function neural network design, in: Y. Jin (Ed.), Multi-Objective Machine Learning, Studies in Computational Intelligence, vol. 16, Springer2006, pp. 221–239.

[11] V. Bevilacqua, G. Mastronardi, F. Menolascina, P. Pannarale, A. Pedone, A novel multi-objective genetic algorithm approach to artificial neural network topology optimisation: the breast cancer classification problem, in: International Joint Conference on Neural Networks (IJCNN'06), Vancouver, 2006, pp. 1958–1965.

[12] N. Kondo, T. Hatanaka, K. Uosaki, Pattern classification via multi-objective evolutionary RBF networks ensemble, in: International Joint Conference on SICE-ICASE 2006, 2006, pp. 137–142 DOI: 10.1109/SICE.2006.315388.

[13] R.A. Teixeira, A.P. Braga, R.H.C. Takahashi, R.R. Saldanha, Improving generalization of MLPs with multi-objective optimization, Neurocomputing 35 (2000) 189–194.

[14] M.A. Costa, A. Braga, B.R. Menezes, R.A. Teixeira, G.G. Parma, Training neural networks with a multi-objective sliding mode control algorithm, Neurocomputing 51 (2003) 467–473.

[15] A.P. Braga, R.H.C. Takahashi, M.A. Costa, R.A. Teixeira, Multi-objective algorithms for neural-networks learning, in: Y. Jin (Ed.), Multi-Objective Machine Learning, Series: Studies in Computational Intelligence, vol. 16, Springer Verlag, Heidelberg2006, pp. 151–171.

[16] I. Kokshenev, A.P. Braga, A multi-objective approach to RBF network learning, Neurocomputing 71 (7–9) (2008) 1203–1209.

[17] V. Chankong, Y. Haimes, Multiobjective Decision Making: Theory and Methodology, Elsevier Science, New York, 1983.

[18] M. Ehrgott, Multicriteria Optimization, Springer, Berlin, Heidelberg, 2005.

[19] A. Geoffrion, Proper efficiency and the theory of vector maximization, Journal of Mathematical Analysis and Applications 22 (1968) 618–630.

[20] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill-Posed Problems, Wiley, New York, 1977.

[21] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural Computation 7 (2) (1995) 219–269.

[22] M.D. Buhmann, M.J.D. Powell, Radial basis function interpolation on an infinite regular grid, in: M. Cox, J. Mason (Eds.), Algorithms for Approximation II, Clarendon Press, Oxford1988, pp. 146–169.

[23] T. Poggio, F. Girosi, Networks for approximation and learning, in: Proceedings of the IEEE, 1990.

[24] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society 58 (1) (1996) 267–288.

[25] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2004) 407–499.

[26] M.Y. Park, T. Hastie, L1-regularization path algorithm for generalized linear models, Journal of the Royal Statistical Society 69 (1) (2007) 659–677.

[27] A. Bjorck, Numerical Methods for Least Squares Problems, SIAM, 1996.

[28] J. Handl, J. Knowles, Exploiting the tradeoff—the benefits of multiple objectives in data clustering, in: Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 3410, Springer-Verlag, New York, 2005, pp. 547–560.

[29] J. Gonzalez, I. Rojas, J. Ortega, H., F.J. Fernandez, A.F. Diaz, Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation, IEEE Transactions on Neural Networks 14 (6) (2003) 1478–1495.

[30] S. Forrest (Ed.), Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, 1993.

[31] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.

[32] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (1978) 461–464.

[33] K.P. Burnham, R.D. Anderson, Model Selection and Inference: A Practical Information Theoretic Approach, Springer-Verlag, New York, 1998.

[34] H. Zou, T. Hastie, R. Tibshirani, On the "degrees of freedom" of the LASSO, Annals of Statistics 35 (5) (2007) 2173–2192.

[35] K. Lang, M. Witbrock, Learning to tell two spirals apart, in: Proceedings of the Connectionist Models Summer School, Morgan Kaufmann, 1988.

[36] M.J.L. Orr, Recent advances in radial basis function networks, Technical Report ⟨www.ed.ac.uk/m̃jo/papers/recad.ps⟩, Institute for Adaptive and Neural Computation, 1999.

[37] A. Asuncion, D. Newman, UCI machine learning repository, URL: ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩, 2007.

[38] L. Prechelt, Proben1: A set of neural network benchmark problems and benchmarking rules, Technical Report 21/94, URL: ⟨http://citeseer.ist.psu.edu/prechelt94proben.html⟩, 1994.

[39] D. Meyer, F. Leisch, K. Hornik, The support vector machine under test, Neurocomputing 55 (1–2) (2003) 169–186.

[40] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, Software available at: ⟨http://www.csie.ntu.edu.tw/c̃jlin/libsvm⟩, 2001.

Illya Kokshenev received his M.S. degree in Computer Science from Kharkov National University of Radio-electronics, Ukraine, in 2004 and his Ph.D. in Electrical Engineering from Federal University of Minas Gerais, Brazil in 2010. His current research interests cover neural, hybrid, and kernel models within classical and multi-objective aspects of machine learning and operational research.

Antonio Padua Braga was born in Brazil in 1963. He obtained his degree in Electrical Engineering and his Master's in Computer Science, both from Federal University of Minas Gerais, Brazil. The main subject of his Ph.D., which was obtained from University of London, England, was Storage Capacity of Boolean Neural Systems. After finishing his Ph.D., he returned to his position as an Adjunct Professor at Department of Electronics Engineering at Federal University of Minas Gerais, where he has now a full professorship. He has published several papers on international journals, conferences and has written two books in neural networks. He is the Head of the Computational Intelligence Laboratory and was founder Co-Editor in Chief of the International Journal of Computational Intelligence and Applications (IJCIA), published by Imperial College Press. His current research interest areas are learning algorithms and applications of artificial neural networks, Bioinformatics and Pattern Recognition.