Embedded local Feature Selection within Mixture of Experts

Billy Peralta[a,*], Alvaro Soto[a]

[a]*Department of Computer Science,Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, 6904411, Santiago, Chile.*

Abstract

A useful strategy to deal with complex classification scenarios is the "divide and conquer" approach. The mixture of experts (MoE) technique makes use of this strategy by jointly training a set of classifiers, or experts, that are specialized in different regions of the input space. A global model, or gate function, complements the experts by learning a function that weighs their relevance in different parts of the input space. Local feature selection appears as an attractive alternative to improve the specialization of experts and gate function, particularly, in the case of high dimensional data. In general, subsets of dimensions, or subspaces, are usually more appropriate to classify instances located in different regions of the input space. Accordingly, this work contributes with a regularized variant of MoE that incorporates an embedded process for local feature selection using L_1 regularization. Experiments using artificial and real-world datasets provide evidence that the proposed method improves the classical MoE technique, in terms of accuracy and sparseness of the solution. Furthermore, our results indicate that the advantages of the proposed technique increase with the dimensionality of the data.

Keywords: Mixture of experts, local feature selection, embedded feature selection, regularization.

*Corresponding author, Telephone: (56 2) 354 4712, Fax: (56 2) 354 4444
Email addresses: bmperalt@uc.cl (Billy Peralta), asoto@ing.puc.cl (Alvaro Soto)

1. Introduction

Performing classification in scenarios with large and complex intra and inter-class variation is a challenging task for most classification methods. In these cases, different subsets of instances might respond to different patterns and, even more, these patterns might arise in different subsets of dimensions. As an example, in visual recognition, changes in illumination or pose conditions usually produce drastic variations in the visual appearance of relevant objects, affecting the discriminative properties of different visual features [35]. As a further example, in gene function prediction, the expression level of particular genes can change substantially under different experimental conditions, affecting the discriminative properties of different co-expression patterns that usually arise on subsets of experiments [15].

A useful strategy to deal with complex classification scenarios is the "divide and conquer" approach. Under this strategy, a complex problem is divided into multiple simpler problems. Decision trees (DTs) are one of the oldest and most widely used classification techniques based on this strategy [36]. This technique consists of building a tree using a partitioning scheme that recursively divides the input space and adjusts local classifiers within each partition. Interestingly, each branch of the resulting tree is in charge of classifying a different subset of instances. Furthermore, classification in each branch is performed using a particular subset of dimensions. We believe that this double "divide and conquer" strategy, that adaptively adjusts each branch of the tree to deal with a selected subsets of instances and dimensions, is one of the main reasons to explain the good performance shown by DTs and their later extensions based on ensemble strategies [8]. Unfortunately, the representational space and usual learning strategies used by DTs impose relevant limitations that affect their abilities to deal with complex classification scenarios. In particular, a DT embeds a hypothesis space given by a disjunction of conjunctions of constraints. These constraints are usually based on single [36] or low dimensional [31] partitions of the input space. Furthermore, common training strategies are based on greedy schemes that can lead to suboptimal

solutions. As an example, the greedy decision at the root node of the tree constrains the conjunctions embedded by all the branches of the tree.

A probabilistic approach to the "divide and conquer" strategy is the mixture of experts (MoE) technique [19]. In contrast to DTs, this technique uses a probabilistic framework that is advantageous in managing the intrinsic uncertainty in the data. MoE divides the data into multiple regions where each region has its own classifier or expert [19]. Each expert is specified by a probability distribution that is conditioned on class values. In the mixture, predictions of experts are weighed using a global model known as gate function. This function adaptively estimates the relevance or weight assigned to each expert for the classification of each input instance.

Both, DTs and MoE, use a "divide and conquer" strategy that divides the input space to perform classification, using a hard partitioning in the case of a DT and a probabilistic, or soft, partitioning in the case of MoE. A relevant difference arises in terms of how each technique handle the dimensionality of each instance: while a DT incorporates an embedded feature selection scheme, MoE does not. We believe that a suitable embedded feature selection scheme can be a useful tool to boost the performance of the MoE technique. In particular, in our experiments for the case of high dimensional datasets we notice that the traditional MoE technique has serious difficulties to learn adequate models. Also, as the number of parameters increases with the number of dimensions, the resulting MoE models become complex usually leading to overfitting problems.

This work contributes with a MoE model that incorporates embedded local feature selection using L_1 regularization. Our main intuition is that particular subsets of dimensions, or subspaces, are usually more appropriate to classify certain input instances. Consequently, we expect to improve the accuracy of traditional MoE models by introducing a technique that adaptively selects subsets of dimensions to train each expert in the mixture.

This paper is organized as follows. Section 2 presents background information about feature selection methods, in particular L_1 regularization. Section 3 describes relevant

previous works. Section 4 presents the proposed approach. Section 5 presents and discusses the results of our experiments. Finally, Section 6 presents our main conclusions and future avenues of research.

2. Background

2.1. Feature selection

In classification problems, the goal corresponds to learn a mapping from an input vector x to an output value y, where x is a vector with D dimensions and y takes categorical values. If vector x is high-dimensional, one can usually improve classification accuracy by discarding irrelevant and redundant features [14, 22]. This process is known as feature or variable selection. In general, there are three main methods for feature selection:

- *Filter methods* rank each input feature x_j in relation to predicting y using a metric of goodness, such as mutual information [3], Pearson correlation [16], Fisher score [10], and chi-square statistic [26], among others [14]. Next, the features are selected according to ranking results. These methods can be incorporated in a sequential forward selection in order to find a subset of discriminant dimensions [16]. Generally, the chosen metric is independent of the final classification model [14]. Filter methods are usually fast and simple, in comparison to alternative techniques.

- *Wrapper methods* search the feature space looking for possible subsets that improve performance. For each subset, these methods execute the classification model and evaluate its resulting predictive power [22], usually using accuracy or F-measure [46]. Then, the subset of features with greatest predictive power is chosen. Main issues associated with these methods are difficulties defining the best metric to measure predictive power, as well as the computational complexity associated to the evaluation of a large number of subsets of features, $2^D - 1$ subsets in the worst case (exhaustive search).

- *Embedded methods* combine feature selection and model fitting into a single optimization problem. DT [36] and Adaboost [12] can be considered embedded techniques, although an explicit criterium for feature minimization is not included during the training process. Two popular techniques to embed feature selection inside a classification algorithm are L_1-regularization [42] and automatic relevancy determination [27].

This paper concentrates on embedded models, specifically L_1-regularization, due to their computational tractability and formal soundness. Although filter methods are faster than alternative techniques, they are usually less effective, as they use a metric that is independent of the final classification scheme. On the other hand, wrapper methods are generally more reliable, as they can take advantage of robust classification algorithms. Nevertheless, these methods are slow due to the usually large number of subsets to explore, and the complexity associated to repeatedly training a robust classifier [22]. Embedded models are attractive because they use a reliable measure of goodness, similar to wrapper methods, but they avoid retraining a predictor for each feature subset explored.

2.2. L_1 regularization

Consider the context of linear models given by the expression $y = w^T x + b$, where $x \in \Re^D$ is the input vector, $y \in \Re$ is the output value, $w \in \Re^D$ is the vector of coefficients, and $b \in \Re$ is the bias [4]. Selecting features by means of regularization fits a vector w of parameters, and at the same time maximizes the number of coefficients w_i of w that takes value equal to zero. . As a consequence, for each coefficient $w_i = 0$, the associated dimension x_i of x can be ignored.

The details of L_1 regularization are derived in the context of probabilistic models [42]. Specifically, the optimization for simultaneous fitting and regularization of the likelihood function of a probabilistic model can be expressed as:

$$\hat{L_0} \;=\; L(w) \;+\; \lambda \left\| w \right\|_0 , \tag{1}$$

where L is the likelihood function that depends on the parameter w. In this case, $\|w\|_0 = \sum_{j=1}^{D} I(|w_j| > 0)$ counts the number of nonzero elements of vector w. $\lambda > 0$ is a trade-off constant

that balances between model fitting and regularization. The direct maximization of Equation (1) is in general unfeasible due to the discrete nature of $I(|w_j|)$. Considering a relaxation of the previous objective function, one has:

$$\hat{L_1} \;=\; L(w) \;+\; \lambda \|w\|_1 \,, \tag{2}$$

where $\|w\|_1 = \sum_{j=1}^{d} |w_j|$. This results in a sparse weight vector w, which means that many of its elements are zero. The nonzero elements correspond to relevant features. This method is known in the statistics community as Lasso (least absolute shrinkage and selection operator), or L_1 regularization [42]. Equation (2) can be rewritten as:

$$\begin{aligned} \hat{w} \;&=\; \arg\max_{w} L(w) \\ \text{subject to} \quad & ||w||_1 \leq t, \end{aligned}$$

where t is an upper bound on the L_1 norm of weights. More precisely, a tight bound t is equivalent to a heavy regularization λ, whereas a loose bound t corresponds to a small value of λ. Lasso can be interpreted similarly to a quadratic cost function with linear constraints and is thus a convex quadratic problem, which has efficient algorithms to solve it [6].

3. Related work

In a seminal work, Jacobs et al. [19] introduce the MoE technique. They divide the space of data into several separate models, where each model has its own supervised classifier. Then, they use a gradient approach to learn parameters which, in this case, is a vector of weights. Finally, they apply this model to multi-speaker vocal recognition.

They base their work on Hampshire et al. [17], who combine the outputs of local experts but without considering localization.

Jordan and Jacobs [20] extend the MoE formulation to a hierarchical case. They treat this model as a conditional mixture model where the distribution of outputs is given by a mixture of component distributions referred as experts. These experts and mixing coefficients are conditioned on input features. Also, the mixing coefficients are controlled by gating distributions. They use the EM algorithm [9] to learn model parameters through a maximum likelihood scheme. They experiment with a robot dynamic problem where they obtain results that are comparable to a neural network trained with the Backpropagation algorithm, but with greater speed. A comparison of MoE with other ensemble methods is given by Vogdrup [47], where a variant of MoE outperforms Adaboost, Bagging, and hierarchical MoE techniques in terms of classification performance.

Titsias and Likas [43] present a three-level hierarchical mixture model for classification where each mixture component has an independent class-conditional mixture. The model estimates the posterior probability of class membership using a similar scheme to the MoE classifier. Model parameters are learned using maximum likelihood. Their results indicate that the final model improves classification with respect to the case where class-conditional information is not considered in each mixture component.

Bishop and Svensén [5] propose a full Bayesian treatment of the hierarchical mixture of experts combining local and global variational methods. For doing this, they establish a lower bound on the marginal probability of data under the model. The greatest difficulty with this Bayesian approach comes from the resulting gating distribution that do not admit a conjugate prior. They use a variational approximation for the logistic function and approximate the joint distribution of the model parameters by a factorized distribution. They apply this method to a kinematics problem, outperforming the results of a hierarchical MoE.

In the context of nonparametric Bayesian models, Rasmussen and Ghahramani [37] present a nonparametric extension to MoE models, where they use Gaussian processes

to model the experts. Also, they use an input-dependent adaptation of the Dirichlet Process to implement a gating network for an infinite number of experts. Inference is performed using Gibbs sampling. This model adjusts the covariance function according to inputs. Simulations show the viability of their approach, however, this model is complex, as it depends on many hyperparameters where interpretability is not natural.

Meeds and Osindero propose an alternative infinite mixture of experts where each expert comprises a multivariate Gaussian distribution to model its inputs, and a Gaussian Process to model its outputs [29]. They use a full generative model over input and output spaces. This approach presents some advantages related to conditional models due to its capability to deal with incomplete data, however, as in [37], this work requires fitting a large number of hyperparameters.

Additionally, some variations in the form of gating and expert functions have been proposed. Xu et al. [53] suggest to replace the usual multinomial logit model with Gaussian basis functions, where each expert is modeled by a Gaussian function. This idea adds flexibility to model the local covariance of the data. Nguyen et al. [32] propose a variation to the classical MoE by using an evolutionary algorithm to learn the model. The overall model is an ensemble, where each component is a mixture of experts. In the context of regression, Lima et al. [24] combine MoE with support vector machines in a probabilistic framework, where the gate functions are represented by a normalized kernel function and the experts correspond to support vector machines.

On the other hand, there have been recently some interesting applications of MoE. Saragih et al. [39] apply MoE to a deformable model fitting problem. Ebrahimpour and Jafarlou [11] apply a hierarchical MoE to view-independent face recognition. They use principal component analysis to find a suitable representation of the data and neural networks to model experts and the gate function.

Closely related to our approach, in terms of embedded local feature selection in a mixture model, Pan and Shen [34] propose the use of L_1 regularization for selecting features in the unsupervised case of model-based clustering. In experiments with real datasets, as they state, they do not attain good results for the case of classification,

most likely due to a mismatch between true labels and resulting clusters. Wang and Zhu [50] also apply regularization over clustering but using L_∞ norm. They propose to use quadratic programming to solve equations related to a constrained optimization.

As it can be seen from the our review, a common issue among previous works on MoE for classification is the fact that they do not consider an explicit feature selection scheme. Notable exceptions are [34] and [50] where an embedded feature selection step is applied, but in the context of clustering and not of classification. Also, in the context of regression and closely related to our work, Khalili [21] presents a MoE model that includes an embedded feature selection approach based on a regularization scheme and Gaussian models. Similarly, we propose a regularization scheme to add feature selection to the MoE model, however, we formulate our model in the context of classification using multinomial logit functions. As a consequence, our domain of applications, mathematical formulation, and optimization solution are highly different from the one proposed in [21]. In particular, we use an iterative optimization scheme similar to the one used in [23], while [21] uses a local quadratic approximation for the regularization term.

In the context of alternative techniques to MoE, there are also interesting works on approaches to combine classifiers, and to perform embedded feature selection. Xiao et al. [52] propose to combine classifiers by jointly maximizing accuracy and ensemble diversity using a neural network architecture. Ulas et al. [45] combine classifiers by employing the most informative components of the eigenvectors corresponding to the correlation matrix among classifier outputs. Ñanculef et al. [33] propose to learn an ensemble of regressors using a sequential scheme and a score minimizing classification error and ensemble diversity. All these works do not include an embedded feature selection mechanism.

In terms of classification schemes that include an embedded feature selection process. Wu et al. [51] propose to select groups of features for image classification arguing that, usually, subsets of visual features are related to specific group of instances. Consequently, they incorporate a group Lasso regularizer inside a logistic regression classifier,

solving the resulting optimization problem using a co-ordinate descent method. In the context of object recognition, Yang et al. [54] boost a standard object classifier based on parts, by adding supplementary parts. These additional parts are selected using a classification scheme that includes Lasso regularization over the selected parts. Maldonado et al. [28] perform feature selection inside a support vector machine considering a penalization score over the features used by the kernel functions. In contrast to our approach, [51] and [28] consider a global feature selection and [54] performs a global part selection scheme, while our work considers local feature selection for each expert classifier.

The idea of adding a regularizer over weights that are used to integrate multiple models has been explored in previous works. Hua et al. [48] propose a framework to annotate videos considering different aspects such as low level features and temporal consistency, where each aspect is represented by multiple instance graphs. In particular, they develop a procedure to find a weight for each graph by jointly optimizing all graphs and a regularization score over the weights. Geng et al. [13] develop a method to learn the intrinsic ensemble of manifolds for unlabeled data in a semi-supervised scenario. Their method begins with a guess for initial manifolds, iterating then to find suitable weights that increase the smoothing and discriminative power of the manifolds. Wang et al. [49] also present a model that learns the weights of a graph ensemble. In particular, their method re-ranks web images, constrained to be near the outputs of a textual search. A relevance score for each graph is learned jointly with the weight of each graph by constraining that visually similar images should have similar relevance scores. In this way, these previous works regularize the parameters of the respective models in order to smooth the labels of a graph [48][13], increase discrimination [13][49], or construct a suitable similarity metric between pairs of instances [49]. Our work can be considered complementary to these previous works, in the sense that it focuses on sparsely selecting the features used by the local models of a MoE scheme.

4. Proposed Approach

In this section we present our main contribution, RMoE, a regularized version of MoE technique that incorporates a local feature selection scheme inside each expert and gate function. Our main intuition is that, in the context of classification, different partitions of the input data can be best represented by specific subsets of features. This is particularly relevant in the case of high dimensional spaces, where the common presence of noisy or irrelevant features might obscure the detection of particular class patterns. Specifically, our approach takes advantage of the linear nature of each local expert and gate function in the classical MoE formulation [19], meaning that L_1 regularization can be directly applied. Below, we first briefly describe the classical MoE formulation for classification. Afterwards, we discuss the proposed modification to the MoE model that provides embedded feature selection.

4.1. Mixture of Experts

In the context of supervised classification, there is available a set of N training examples, or instance-label pairs (x_n, y_n), representative of the domain data (x, y), where $x_n \in \Re^D$ and $y_n \in C$. Here C is a discrete set of Q class labels $\{c_1, ..., c_Q\}$. The goal is to use training data to find a function f that minimizes a loss function which scores the capacity of f to predict the true underlying relation between x and y. From a probabilistic point of view [4], a useful approach to find f is to use a conditional probability formulation:

$$f(x) \;=\; \arg\max_{c_i \in C} \; p(y = c_i|x).$$

In the general case of complex relations between x and y, a useful strategy consists of approximating f through a mixture of local functions. This is similar to the case of modeling a mixture distribution [40] and it leads to the MoE model.

We decompose the conditional likelihood $p(y|x)$ as:

$$p(y|x) \;=\; \sum_{i=1}^{K} p(y, m_i|x) \;=\; \sum_{i=1}^{K} p(y|m_i, x)\, p(m_i|x), \tag{3}$$

where Equation (3) represents a MoE model with K experts m_i. Figure 1 shows a schematic diagram of the MoE approach. The main idea is to obtain local models in such a way that they are specialized in a particular region of the data. In Figure 1, x corresponds to the input instance, $p(y|m_i, x)$ is the expert function, $p(m_i|x)$ is the gating function, and $p(y|x)$ is the weighted sum of experts. Note that the output of each expert model is weighed by the gating function. This weight can be interpreted as the *relevance* of expert m_i for the classification of input instance x. Also note that the gate function has K outputs, one for each expert. There are K expert functions that have Q components, one for each class.

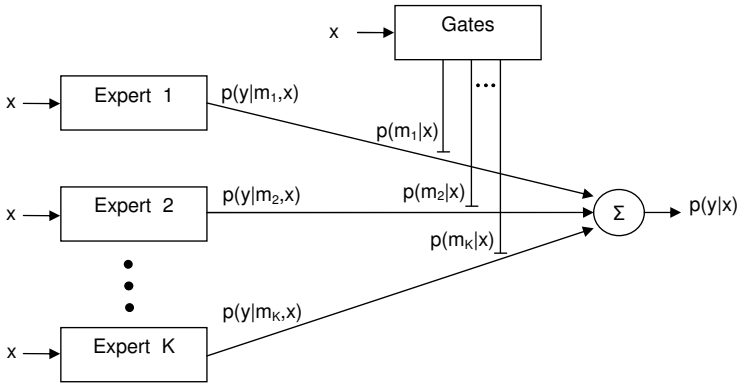

Figure 1: Mixture of experts scheme.

The traditional MoE technique uses multinomial logit models, also known as soft-max functions [4], to represent the gate and expert functions. An important characteristic of this model is that it forces competition among its components. In MoE, such components are expert functions for the gates and class-conditional functions for the experts. The competition in soft-max functions enforces the especialization of experts in different areas of the input space [55].

Using multinomial logit models, a gate function is defined as:

$$p(m_i|x) = \frac{exp(\nu_i^T x)}{\sum_{j=1}^{K} exp(\nu_j^T x)} \tag{4}$$

where $i \in \{1, \ldots, K\}$ indexes expert i and $\nu_i \in \Re^D$ is a vector of model parameters. Component ν_{ij} of vector ν_i models the relation between the gate and dimension j of input instance x.

Similarly, an expert function is defined as:

$$p(y = c_l|x, m_i) = \frac{exp(\omega_{li}^T x)}{\sum_{j=1}^{M} exp(\omega_{ji}^T x)} \tag{5}$$

where ω_{li} depends on class label c_l and expert i. In this way, there are a total of $Q \times K$ vectors ω_{li}. Component ω_{lij} of vector ω_{li} models the relation between expert function i and dimension j of input instance x.

There are several methods to find the value of the hidden parameters ν_{ij} and ω_{lij} [30]. An attractive alternative is to use the EM algorithm. In the case of MoE, the EM formulation augments the model by introducing a set of latent variables, or *responsibilities*, indicating the expert that generates each instance. Accordingly, EM iterations consider an expectation step that estimates expected values for *responsibilities*, and a maximization step that updates the values of parameters ν_{ij} and ω_{lij}. Specifically, the posterior probability of the *responsibility* R_{in} assigned by the gate function to expert m_i for an instance x_n is given by [30]:

$$\begin{aligned} R_{in} &= p(m_i|x_n, y_n) \\ &= \frac{p(y_n|x_n, m_i)\, p(m_i|x_n)}{\sum_{j=1}^{K} p(y_n|x_n, m_j)\, p(m_j|x_n)} \end{aligned} \tag{6}$$

Considering these responsibilities and Equation (3), the expected complete log-likelihood $\langle L_c \rangle$ used in EM iterations is [30]:

$$\langle L_c \rangle = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}\ [log\ p(y_n|x_n, m_i)\ + log\ p(m_i|x_n)] \tag{7}$$

4.2. Regularized Mixture of Experts (RMoE)

To embed a feature selection process in the MoE approach, we use the fact that in Equations (4) and (5), the multinomial logit models for gate and experts functions contain linear relations in the relevant parameters. This linearity can be straightforwardly used in feature selection by considering that a null parameter component ν_{ij} or ω_{lij} implies that dimension j is irrelevant for gate function $p(m_i|x)$, or expert model $p(y|m_i, x)$, respectively. Consequently, we propose to penalize complex models using L_1 regularization. A similar consideration is used in [34], but in the context of unsupervised learning. The idea is to maximize the likelihood of the data while simultaneously minimizing the number of non-null parameter components ν_{ij} and ω_{lij}. Considering that there are Q classes, K experts, and D dimensions, the expected L_1 regularized log-likelihood $\left\langle L_c^R \right\rangle$ is given by:

$$\left\langle L_c^R \right\rangle = \langle L_c \rangle - \lambda_\nu \sum_{i=1}^{K} \sum_{j=1}^{D} |\nu_{ij}| - \lambda_\omega \sum_{l=1}^{Q} \sum_{i=1}^{K} \sum_{j=1}^{D} |\omega_{lij}| \; . \tag{8}$$

To maximize Equation (8) with respect to model parameters, we first use the standard fact that the likelihood function in Equation (7) can be decomposed in terms of independent expressions for gate and expert models [30]. In this way, the maximization step of the EM based solution can be performed independently for gate and expert parameters [30]. In our problem, each of these optimizations has an additional term given by the respective regularization term in Equation (8). To handle this case, we observe that each of these optimizations is equivalent to a regularized logistic regression [23]. As shown in [23], this problem can be solved by using a coordinate ascent optimization strategy [44] given by a sequential two-step approach that first models the problem as an unregularized logistic regression and afterwards incorporates the regularization constraints.

In summary, we handle Equation (8) by using an EM based strategy that, at each step, solves the maximization with respect to model parameters by decomposing the

problem in terms of gate and expert parameters. Each of these problems is, in turn, solved using the strategy proposed in [23]. Next, we provide details of this procedure.

Optimization of the unregularized log-likelihood

In this case, we solve the unconstrained maximization of the log-likelihood given by Equation (7). First, we optimize the log-likelihood with respect to vector ω_{li}. The maximization of the expected log-likelihood $\langle L_c \rangle$ implies differentiating Equation (7) with respect to ω_{li}:

$$\frac{\partial \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in} \; [log \; p(y_n|x_n, m_i) \,]}{\partial \omega_{li}} \;\; = \;\; 0 \tag{9}$$

which is equivalent to:

$$-\sum_{n=1}^{N} R_{in} \left(p(y_n|x_n, m_i) - y_n\right) x_n \;\; = \;\; 0. \tag{10}$$

In this case, the classical technique of least-squares cannot be directly applied because of the soft-max function in $p(y_n|x_n, m_i)$. Fortunately, as described in [20] and later in [30], Equation (10) can be approximated by using a transformation that implies inverting the soft-max function. Using this transformation, Equation (10) is equivalent to an optimization problem that can be solved using a weighted least squares technique [4]:

$$\min_{\omega_{li}} \;\; \sum_{n=1}^{N} R_{in} \left(\omega_{li}^T x_n - log \; y_n\right)^2 \tag{11}$$

A similar derivation can be performed with respect to vectors ν_i. Again differentiating Equation (7), in this case with respect to parameters ν_{ij} and applying the transformation suggested in [20], we obtain:

$$\min_{\nu_i} \;\; \sum_{n=1}^{N} \left(\nu_i^T x_n - log R_{in}\right)^2 \tag{12}$$

Optimization of the regularized likelihood

Following the procedure in [23], we add the regularization term to the optimization problem given by Equation (11), obtaining an expression that can be solved by any standard algorithm for Lasso resolution [42]:

$$\begin{aligned} \min_{\omega_{li}} \quad & \sum_{n=1}^{N} R_{in} \left(log\, y_n - \omega_{li}^T x_n \right)^2 \\ \text{subject to:} \quad & ||\omega_{li}||_1 \leq \lambda_\omega \end{aligned} \tag{13}$$

Similarly, we can also obtain a standard Lasso optimization problem to find parameters ν_{ij} :

$$\begin{aligned} \min_{\nu_i} \quad & \sum_{n=1}^{N} \left(logR_{in} - \nu_i^T x_n \right)^2 \\ \text{subject: to} \quad & ||\nu_i||_1 \leq \lambda_\nu \end{aligned} \tag{14}$$

Specifically, in the case of T iterations, there are a total of $T * K * (Q + 1)$ Lasso optimization problems related to the maximization step of the EM algorithm. To further reduce this computational load, we slightly modify this maximization by applying the following two-steps scheme:

- Step-1: Solve K Lasso optimization problems to find gate parameters ν_{ij} assuming that each expert uses all the available dimensions. In this case, there are $T - 1$ iterations.
- Step-2: Solve $K * (Q + 1)$ Lasso optimization problems to find expert parameters ω_{lij} applying the feature selection process. In this case, there is a single iteration.

Using the previous scheme we reduce from $T*K*(Q+1)$ to $K*(T-1)+K*(Q+1)$ the number of Lasso optimization problems that we need to solve in the maximization step of the EM algorithm. In our experiments, we do not notice a drop in performance by using this simplification, but we are able to increase processing speed in one order of magnitude.

In summary, starting by assigning random values to the relevant parameters ν_{ij} and ω_{lij}, our EM implementation consists of iterating the following two steps:

- Expectation: estimating responsabilities for each expert using Equation (6), and then estimating the outputs of gate and experts using Equations (4) and (5).
- Maximization: updating the values of parameters ν_{ij} and ω_{lij} in Equations (13) and (14) by solving $K*(T-1)+K*(Q+1)$ Lasso optimization problems according to the approximation described above in Step-1 and Step-2.

5. Experiments

In this section, we use synthetic and real datasets to analyze the performance of RMoE. In particular, we compare its performance against the traditional MoE technique. Furthermore, we analyze RMoE in terms of classification accuracy and dimensionality reduction under different parameter configurations. RMoE is oriented to classification tasks, therefore, it requires categorical class variables. Finally, we compare the performance of RMoE against three popular classification algorithms that also consider embedded feature selection: Random Subspace (RS) [18], Decision Trees (DT) [36], and Adaboost (AB) [12]. In the case of Adaboost, we use decision stumps as weak classifiers.

All results reported in this section are obtained by averaging performances on 30 hold-out folds. In each case, classification performance is measured by using an independent test set that is not used in the determination of any of the parameters associated to each method. For each case, the estimation of parameters is performed by using training sets corresponding to 50% of the available data. In particular, in the case of RMoE, we first use the training set to apply a 3-fold cross-validation procedure to obtain suitable values for the regularization parameters λ_ν and λ_ω. Specifically, we test a total of 24 combinations of parameter values for λ_ν and λ_ω. For parameter λ_ω we test the set of values: $\{20, 10, 5, 2, 1, 0.5, 0.2, 0.1\}$ and, for each of these cases, we select the corresponding λ_ν by multiplying λ_ω by the factors in: $\{2, 1, 0.5\}$. We choose the

combination of λ_ν and λ_ω with highest accuracy, according to the 3-fold cross validation. After the values of λ_ν and λ_ω are selected, we use the complete training set to estimate parameters for experts and gate function. For the cases of Random Subspace, Decision Trees, and Adaboost, main parameters are the number of trees in the forest, the minimum number of records in leaf nodes, and the number of decision stumps (weak classifiers), respectively. These parameters are obtained from the best model according to 2-fold cross-validation inside the training set of each hold-out fold. For Random Subspace, we experiment using from 5 to 50 trees in the forest with a step size of 5. For Decision Trees, we test using from 1 to 10 records in leaf nodes with a step size of 1. For Adaboost, we test using from 10 to 100 decision stumps with a step size of 10.

5.1. Synthetic datasets

We generate 6 synthetic datasets, each consisting of two equiprobable classes. We define relevant patterns for each class using a subset of the total number of dimensions. Specifically, relevant dimensions for each class are represented by a multivariate Gaussian distribution using a randomly selected subset consisting of 4 to 6 dimensions. For each training instance the remaining dimensions are filled using samples from an Uniform distribution. As shown in Table 1, we vary the total number of dimensions in the datasets from 200 to 1200 dimensions. In terms of parameters, Gaussian distributions are selected in such a way that their central parts do not overlap. Means vectors are randomly selected within the range $[0, 20]$, while covariance matrices are diagonal with non-zero values randomly selected within the range $[0, 1]$. In case of Uniform distributions, they are defined within the range $[0, 20]$.

Table 2 shows that, in terms of classification accuracy, RMoE outperforms MoE technique in all the tested datasets. We can observe that the level of improvement of RMoE with respect to MoE fluctuates among the different datasets. For example, in Dataset 1 RMoE improves the performance of MoE by 40%, while in Dataset 8 the improvement is 46%.

Table 3 compares RMoE and MoE in terms of parameter dimensionality for the

Table 1: Synthetic datasets used for experiments.

Dataset name	#Instances	#Dimensions	#Relevant dimensions {class-1,class-2}
Dataset 1	400	200	$\{4, 4\}$
Dataset 2	400	400	$\{4, 4\}$
Dataset 3	500	600	$\{5, 5\}$
Dataset 4	500	800	$\{5, 5\}$
Dataset 5	600	1000	$\{6, 6\}$
Dataset 6	600	1200	$\{6, 6\}$

Table 2: Accuracy on synthetic datasets using 30 hold-out partitions. RMoE(λ_ν^*,λ_ω^*) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair ($\lambda_\nu^{'}$,$\lambda_\omega^{'}$) show the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details).

Dataset name	MoE	RMoE($\lambda_\nu^{'}$,$\lambda_\omega^{'}$)	(λ_ν^*,λ_ω^*)
Dataset 1	52.9(4.6)	**93.1(2.0)**	(5,10)
Dataset 2	54.4(4.0)	**98.3(1.3)**	(10,10)
Dataset 3	53.4(2.9)	**97.0(1.2)**	(5,5)
Dataset 4	51.7(3.0)	**96.4(1.1)**	(10,10)
Dataset 5	52.6(4.0)	**97.4(1.0)**	(7.5,10)
Dataset 6	51.9(2.8)	**98.2(0.9)**	(10,5)

classification results shown in Table 2. The main observation is that, as expected, for these types of high dimensional datasets RMoE provides sparse models.

Finally, we evaluate the performance of the feature selection step included in RMoE. In this case, we use the fact that for the synthetic datasets we know the true dimensions used to generate the class pattern behind each instance. Specifically, we define the following score that quantifies the relevance assigned by RMoE to dimension j for the classification of input instance x:

Table 3: Average parameter dimensionality for results shown in Table 2.

Dataset name	MoE	RMoE(λ_ν^*,λ_ω^*)	Feature reduction
Dataset 1	200	33	83.5%
Dataset 2	400	23	94.2%
Dataset 3	600	38	93.7%
Dataset 4	800	22	97.2%
Dataset 5	1000	30	97.0%
Dataset 6	1200	39	96.7%

$$\varphi(j;x) \;=\; \sum_{i=1}^{K} p(m_i|x) \times |\omega_{y^*ij}|, \tag{15}$$

where K is the number of experts, y^* is the class label provided by RMoE to input instance x, ω_{y^*ij} is the parameter associated to dimension j when expert i is applied to class y^*, and $p(m_i|x)$ is the posterior probability assigned by the gate to expert i given input x. In short, this score evaluates the relevance of each data dimension considering both: the weight assigned to the data dimension by each expert and the weight assigned by the gate to the respective expert. Equation (15) considers absolute values for parameters ω_{y^*ij} because, according to the regularization, only weight values near zero imply that the corresponding feature is irrelevant for the mixture.

Given scores $\varphi(j;x)$ for each instance x, we construct a feature relevance ranking, by sorting these scores in descending order. Afterwards, we analyze the positions reached in the ranking by the true dimensions used to build each data instance. Table 4 indicates position in the ranking under which, in average, it is possible to find a given percentage of the relevant dimensions. We use average percentage because each data instance has a different ranking of relevance. As an example, Table 4 shows that in Dataset 3, according to the ranking, on average of 90% of the relevant features are among the first 10 dimensions with highest score.

In general, results differ from one dataset to another. For example, for Dataset 2

all relevant dimensions appear among the first 18 positions of the ranking, while for Dataset 1 all relevant dimensions appear among the first 75 dimensions of the ranking. These positions correspond to approximately 4.5% and 37.5% of the total number of dimensions, respectively. The presence of irrelevant features at the top of rankings can be related to the work of [14], that unexpectedly, shows that redundant features can improve classification.

Table 4: Relative relevance assigned by RMoE to features used to generate class patterns in synthetic datasets using the score in Equation (15). Each column indicates the position in the ranking where, on average, it is possible to find a given percentage (column header) of the relevant dimensions.

Dataset name	60%	70%	80%	90%	100%	Total Dimensions
Dataset 1	15	72	73	74	75	200
Dataset 2	12	13	16	17	18	400
Dataset 3	7	8	9	10	46	600
Dataset 4	10	25	30	31	33	800
Dataset 5	8	9	15	16	17	1000
Dataset 6	6	8	50	51	52	1200

5.2. Real datasets

We test performance of RMoE using 13 real datasets. Table 5 describes the main characteristics of each of these datasets. Arrhythmia, Ionosphere, Musk-1, Secom, Semeion, Spectf, and Sonar datasets are taken from UCI Machine Learning Repository [2]. Leukemia, Lymphoma, Colon, and Dataset-C are biological datasets taken from [1]. BrainTumor is a biological dataset taken from [41]. PIE10P is a face recognition dataset taken from [25]. In the case of PIE10P and Leukemia datasets, we select the top 1000 and 1500 features, respectively, according to the Fisher score filter [10]. We reduce the number of features in these two datasets to obtain a pool of datasets with a highly diverse number of features, as shown in Figure 2. Finally, in all cases we removed

the variables with zero variance.

Table 5: Real datasets used for experiments.

Dataset name	#Objects	#Dimensions	#Classes
Ionosphere	351	33	2
Spectf	267	44	2
Sonar	208	61	2
Musk-1	486	168	2
Semeion	1593	256	10
Arrhythmia	452	279	16
Secom	1567	471	2
PIE10P	210	1000	10
Leukemia	75	1500	2
Colon	62	2001	2
Lymphoma	45	4027	2
BrainTumor	90	5921	5
Dataset-C	60	7130	2

We test RMoE using the same combinations of parameter values for λ_ν and λ_ω considered in the case of synthetic datasets. Table 6 shows the accuracy of the best RMoE model obtained for each dataset (RMoE(λ_ν^*,λ_ω^*)).

Regarding the results in Table 6, RMoE outperforms the traditional MoE technique in all the tested datasets. The increase in performance depends on each particular dataset but, as expected, the advantages of RMoE with respect to MoE increase with the dimensionality of the dataset. This is the case of datasets such as Secom, PIE10P, Leukemia, Lymphoma, Colon, BrainTumor, and Dataset-C.

To check if these results are statistically significant, we run a paired Student's t-test (Behrens-Fisher problem [38]) to compare the results of RMoE against the results of each of the alternative techniques. When comparing RMoE to MoE, RMoE has greater accuracy than MoE with over 95% of confidence in all but one of the datasets where

Table 6: Accuracy on real datasets using 30 hold-out partitions. RMoE(λ_ν^*,λ_ω^*) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair ($\lambda_\nu^{'}$,$\lambda_\omega^{'}$) shows the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details).

Dataset name	MoE	RMoE(λ_ν^*,λ_ω^*)	($\lambda_\nu^{'}$,$\lambda_\omega^{'}$)	RS	DT	AB
Ionosphere	82.7(3.0)	84.1(2.6)	(0.25,0.5)	**93.0 (1.6)**	87.8 (2.2)	91.3 (1.7)
Spectf	70.1(3.7)	76.6(3.4)	(10,20)	**80.2(1.5)**	74.7(3.9)	79.5(2.6)
Sonar	64.1(5.6)	74.1(4.2)	(2.25,2)	**79.1(4.0)**	70.8(4.9)	**79.1(3.6)**
Musk-1	67.2(4.9)	80.0(2.0)	(1,0.75)	**85.5(2.4)**	75.3(3.4)	82.1(2.7)
Semeion	66.1(2.3)	85.1(1.5)	(2.5,2)	**91.7(0.9)**	66.6(1.9)	60.9(2.4)
Arrhythmia	45.0(10.9)	66.0(2.2)	(1,2)	**69.8(1.9)**	65.2(3.2)	82.1(2.7)
Secom	59.4(7.3)	73.1(1.6)	(10,10)	**74.6(1.3)**	66.3(4.0)	71.9(2.4)
PIE10P	32.9(11.0)	**99.4(1.1)**	(4,2)	95.2(1.8)	78.8(5.1)	76.5(6.5)
Leukemia	59.0(13.1)	93.1(5.6)	(0.5,1)	**95.9(3.1)**	90.5(4.3)	80.8(16.3)
Colon	54.7(11.8)	**82.0(5.2)**	(1.5,1.5)	59.7(4.9)	57.7(8.2)	60.0(9.3)
Lymphoma	53.3(10.0)	**88.8(4.7)**	(3.25,3.5)	85.1(7.1)	72.2(9.5)	68.3(19.9)
BrainTumor	36.9(7.1)	**83.8(4.1)**	(1.5,2)	79.3(3.3)	65.7(5.2)	74.5(6.1)
Dataset-C	51.2(8.3)	**62.7(9.0)**	(1,1)	59.7(4.9)	57.7(8.2)	60.0(9.3)

the confidence drops to 93% (Ionosphere). In terms of the other alternative techniques under test and all six high-dimensional datasets under evaluation (over 500 dimensions), RMoE also shows superior performance with over 95% of confidence. Exceptions are some cases of the Random Subspace (RS) technique, where for the datasets Lymphoma and Dataset-C, RMoE has better accuracy than RS with a 83% and 82% of confidence, respectively, and for the case of Leukemia dataset, where RS has better accuracy than RMoE with a 96% of confidence.

Figure 2 shows the accuracy achieved by the methods under consideration as a function of the dimensionality of the dataset. We observe that, for datasets with low dimensionality ($<$ 500 dimensions), the performance of RMoE is slightly lower than

classifiers such as Random Forest and Adaboost. However, in the case of datasets with high dimensionality, RMoE shows comparable and, in most cases, superior performances than the alternative techniques under evaluation. This confirms our intuition about the relevance of a suitable embedded feature selection scheme when dealing with high dimensional data. A complementary advantage of our method is that it provides a sound probabilistic framework for classification.

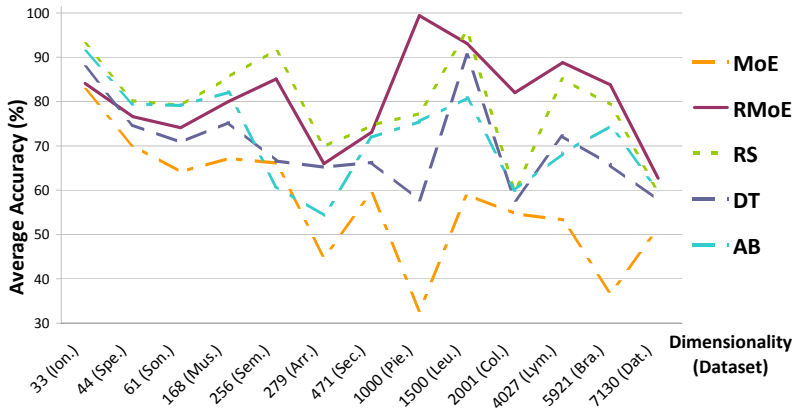

Figure 2: Average accuracy on real datasets for all tested algorithms versus number of dimensions of the datasets.

Table 7 shows average parameter dimensionality as well as percentage of features reduction provided by RMoE with respect to MoE. As in the case of synthetic datasets, Table 7 shows that RMoE favors sparse solutions with a competitive or superior accuracy than the traditional MoE technique. In general, results are variable in terms of sparsity. For example, for datasets Colon, Lymphoma, BrainTumor, and Dataset-C, the best models provided by RMoE use less than 1% of the available dimensions. On the other hand, for the dataset Ionosphere, RMoE uses 78.1% of all dimensions. In general, when the dataset has few dimensions, the difference between RMoE and MoE is less noticeable. Therefore, as expected, feature selection tends to be more useful when datasets have more dimensions.

Finally, we analyze execution times of MoE and RMoE. By considering n records and d dimensions, with $d > n$, in the case of MoE the main parameters are obtained by solving a weighted least square optimization that is usually dominated by complexity

Table 7: Average dimensionality of parameters in real datasets.

Dataset name	MoE	RMoE(λ_ν^*,λ_ω^*)	Dimensionality reduction of RMoE
Ionosphere	32	25	21.9%
Spectf	43	5	88.4%
Sonar	60	17	71.7%
Musk-1	167	34	79.6%
Semeion	256	77	70.0%
Arrhythmia	279	18	93.5%
Secom	471	12	97.5%
PIE10P	1000	20	98.0%
Leukemia	1500	23	98.5%
Colon	2000	19	99.1%
Lymphoma	4026	13	99.7%
BrainTumor	5921	24	99.6%
Dataset-C	7129	14	99.8%

$O(d^3)$, while in the case of RMoE such step depends on the method used to solve a Lasso optimization problem. In the case of RMoE, we use the iterative solution proposed by [7]. Using $\lambda_\nu = \lambda_\omega = 1$, Table 8 shows that RMoE is usually slower than MoE in the case of few dimensions. However, in high dimensional cases, RMoE is able to run faster than MoE by taking advantage of the sparsity given by Lasso optimization.

Table 8: Average execution time (in miliseconds) of MoE and RMoE for different datasets using 100 independent executions in each case.

Dataset name	MoE	RMoE
Ionosphere	80	110
Spectf	40	310
Sonar	40	450
Musk-1	170	850
Semeion	2270	5130
Arrhythmia	920	2110
Secom	430	1550
PIE10P	1220	2700
Leukemia	3160	420
Colon	5160	2260
Lymphoma	2.6E4	0.1E4
BrainTumor	43.2E4	1.8E4
Dataset-C	13.5E4	0.4E4

6. Conclusions

This paper introduces RMoE, a regularized variant of mixture of experts, where local feature selection is performed on experts and gate function using L_1 regularization. Our experiments provide evidence that the proposed technique improves classical mixture of experts in terms of accuracy and sparseness of the solution. In particular, using a diverse set of synthetic and real datasets, RMoE is able to find classification models that provide not only greater accuracy but also use less than 5% of the available features. In this respect, as expected, the proposed technique has demonstrated greater utility for the case of high dimensional datasets. In the case of low dimensional datasets, there is no significant difference in terms of classification accuracy when comparing RMoE to the classical MoE model. As for goodness of feature selection, in the case of synthetic

datasets, where there is ground truth information about the process used to generate the data, the proposed method is able to recover most of the true relevant dimensions. In terms of the performance of RMoE with respect to popular alternative techniques that also uses embedded feature selection, we also observe that RMoE shows a superior classification accuracy for the case of datasets with a high number of dimensions. As future work, we believe that an important constraint of RMoE is the assumption that the conditional distributions for each expert follows a logistic regression. We plan to explore alternative and more flexible distributions to model gate and expert functions. Another avenue of future research is to explore the incorporation of an embedded feature selection scheme for the case of hierarchical mixture of experts.

7. Acknowledgements

This work was partially funded by FONDECYT grant 1095140.

References

[1] J. Aguilar, Dataset repository in arff, http://www.upo.es/eps/aguilar/datasets.html, 2008.

[2] A. Asuncion, D. Newman, UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[3] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (1994) 537–550.

[4] C. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, New York, USA, 2nd edition, 2007.

[5] C. Bishop, M. Svensén, Bayesian hierarchical mixtures of experts, in: Conference on Uncertainty in Artificial Intelligence, pp. (2003) 57–64.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, United Kingdom, 2004.

[7] J. Bradley, A. Kyrola, D. Bickson,C. Guestrin, Parallel Coordinate Descent for L1-Regularized Loss Minimization, International Conference on Machine Learning (2011) 321–328.

[8] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[9] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1977) 1–38.

[10] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley-Interscience, USA, second edition, 2001.

[11] R. Ebrahimpour, F.M. Jafarlou, View-independent face recognition with hierarchical mixture of experts using global eigenspaces, Journal of Communication and Computer 7 (2010) 1103–1107.

[12] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the European Conference on Computational Learning Theory, Springer-Verlag, London, UK, 1995, pp. 23–37.

[13] B. Geng, D. Tao, T. Xu, L. Yang, X. Hua, Ensemble manifold regularization, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 1227–1233.

[14] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[15] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Journal of Machine Learning 46 (2002) 389–422.

[16] M. Hall, Correlation-based Feature Selection for Machine Learning, Ph.D. thesis, University of Waikato, 1999.

[17] J. Hampshire, A. Waibel, The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 751–769.

[18] T. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844.

[19] R. Jacobs, M. Jordan, S. Nowlan, G. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1991) 79–87.

[20] M. Jordan, R. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6 (1994) 181–214.

[21] A. Khalili, New estimation and feature selection methods in mixture-of-experts models, Canadian Journal of Statistics 38 (2010) 519–539.

[22] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.

[23] S.I. Lee, H. Lee, P. Abbeel, A.Y. Ng, Efficient L1 regularized logistic regression, in: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI) (2006) 401–408.

[24] C. Lima, A. Coelho, F. Von Zuben, Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification, Information Sciences 177 (2007) 2049–2074.

[25] H. Liu, Arizona state university: Feature selection datasets, http://featureselection.asu.edu/datasets.php, 2012.

[26] H. Liu, R. Setiono, Chi2: Feature selection and discretization of numeric attributes, in: J. Vassilopoulos (Ed.), Proceedings of the International Conference on Tools with Artificial Intelligence, IEEE Computer Society, Herndon, Virginia, 1995, pp. 388–391.

[27] D. MacKay, Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks, Network: Computation in Neural Systems 6 (1995) 469–505.

[28] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Information Sciences 181 (2011) 115–128.

[29] E. Meeds, S. Osindero, An alternative infinite mixture of Gaussian process experts, in: Advances In Neural Information Processing Systems, (2005) pp. 883–890.

[30] P. Moerland, Some methods for training mixtures of experts, Technical Report, IDIAP Research Institute, 1997.

[31] S.K. Murthy, S. Kasif, S. Salzberg, A system for induction of oblique decision trees, Journal of Artificial Intelligence Research 2 (1994) 1–32.

[32] M. Nguyen, H. Abbass, R. McKay, A novel mixture of experts model based on cooperative coevolution, Neurocomputing 70 (2006) 155–163.

[33] R. Ñanculef, C. Valle, H. Allende, C. Moraga, Training regression ensembles by sequential target correction and resampling, Information Sciences 195 (2012) 154–174.

[34] W. Pan, X. Shen, Penalized model-based clustering with application to variable selection, Journal of Machine Learning Research 8 (2007) 1145–1164.

[35] N. Pinto, D. Cox, J. DiCarlo, Why is real-world visual object recognition hard?, PLoS Computational Biology 4 (2008) 151–156.

[36] J. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc., California, USA, 1993.

[37] C. Rasmussen, Z. Ghahramani, Infinite mixtures of gaussian process experts, in: Advances in Neural Information Processing Systems, pp. (2001) 881–888.

[38] J. Rice, Mathematical Statistics and Data Analysis, Duxbury Press, 2nd edition, 1994.

[39] J. Saragih, S. Lucey, J. Cohn, Deformable model fitting with a mixture of local experts, International Conference on Computer Vision (2009) 2248–2255.

[40] D. Scott, S. Sain, Multi-dimensional density estimation, Elsevier, Amsterdam, 2004, pp. 229–263.

[41] A. Statnikov, I. Tsamardinos, Y. Dosbayev, C. Aliferis, GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression Data, International Journal of Medical Informatics 74 (2005) 491–503.

[42] R. Tibshirani, Regression shrinkage and selection via the Lasso, Journal of the Royal Statistical Society (Series B) 58 (1996) 267–288.

[43] M. Titsias, A. Likas, Mixture of experts classification using a hierarchical mixture model, Neural Computation 14 (2002) 2221–2244.

[44] P. Tseng, Convergence of block coordinate descent method for nondifferentiable maximization, Journal of Optimization Theory and Applications 109 (2001) 475–494.

[45] A. Ulas, O. Taner, E. Alpaydin, Eigenclassifiers for combining correlated classifiers, Information Sciences 187 (2012) 109–120.

[46] C. Van-Rijsbergen, Information Retrieval, Butterworth-Heinemann, London, UK, 2nd edition, 1979.

[47] J. Vogdrup, Combining predictors: comparison of five meta machine learning methods, Information Sciences 119 (1999) 91–105.

[48] M. Wang, X. Hua, R. Hong, J. Tang, J. Guo-Jung, Y. Song, Unified video annotation via multigraph learning, IEEE Transactions on Circuits and Systems for Video Technology 19 (2009) 733–746.

[49] M. Wang, H. Li, D. Tao, K. Lu, X. Wu, Multimodal graph-based reranking for web image search, IEEE Transactions on Image Processing 21 (2012) 4649–4661.

[50] S. Wang, J. Zhu, Variable selection for model-based high dimensional clustering and its application to microarray data, Biometrics 64 (2008) 440–448.

[51] F. Wu, Y. Yuan, Y. Zhuang, Heterogeneous feature selection by group Lasso with logistic regression, International Conference on Multimedia (2010) 983–986.

[52] J. Xiao, C. He, X. Jiang, D. Liu, A dynamic classifier ensemble selection approach for noise data, Information Sciences 180 (2010) 3402–3221.

[53] L. Xu, M. Jordan, G. Hinton, An alternative model for mixtures of experts, in: Advances in Neural Information Processing Systems, pp. (1994) 633–640.

[54] K. Yang, M. Wang, X. Hua, S. Yan, H. Zhang, Assemble new object detector with few examples, IEEE Transactions on Image Processing 20 (2011) 3341–3349.

[55] A. Yuille, D. Geiger, Winner-take-all mechanisms, in: M.A. Arbib (Ed.), The handbook of brain theory and neural networks, MIT Press, Cambridge, MA, USA, 1998, p. 1056.

8. Vitae

Billy Peralta Received the M.S.degree from Pontificia Universidad Catolica de Chile, Chile, in 2007. He is currently pursuing the Ph.D. degree at Pontificia Universidad Catolica de Chile, Chile. His research interests include computer vision and clustering.

Alvaro Soto Received the Ph.D. degree in Robotics from the Robotics Institute at Carnegie Mellon University in 2002. He also received a M.Sc. in Electrical and Computer Engineering from Louisiana State University in 1997. He joined the Computer Science Department at Pontificia Universidad Catolica de Chile, where he became Associate Professor in 2007. His main research interests are in statistical machine learning.