Geometry-Based Ensembles: Toward a Structural Characterization of the Classification Boundary

Oriol Pujol and David Masip

Abstract—This paper introduces a novel binary discriminative learning technique based on the approximation of the nonlinear decision boundary by a piecewise linear smooth additive model. The decision border is geometrically defined by means of the *characterizing boundary points*—points that belong to the optimal boundary under a certain notion of robustness. Based on these points, a set of locally robust linear classifiers is defined and assembled by means of a Tikhonov regularized optimization procedure in an additive model to create a final λ-smooth decision rule. As a result, a very simple and robust classifier with a strong geometrical meaning and nonlinear behavior is obtained. The simplicity of the method allows its extension to cope with some of today's machine learning challenges, such as online learning, large-scale learning or parallelization, with linear computational complexity. We validate our approach on the UCI database, comparing with several state-of-the-art classification techniques. Finally, we apply our technique in online and large-scale scenarios and in six real-life computer vision and pattern recognition problems: gender recognition based on face images, intravascular ultrasound tissue classification, speed traffic sign detection, Chagas' disease myocardial damage severity detection, old musical scores clef classification, and action recognition using 3D accelerometer data from a wearable device. The results are promising and this paper opens a line of research that deserves further attention.

Index Terms—Classification, ensemble of classifiers, Gabriel neighboring rule, visual object recognition.

1 Introduction

In the last decade, there have been several successful attempts relating geometry-based reasoning with the design of learning machines. Most of the efforts in this line are restricted to editing rules for the nearest neighbor technique [3], [4], [7]. However, several researchers have recently focused on the relationship between large margin classifiers and geometry. In [5], the authors use a graph-based technique on the training data set to locate possible support vectors while editing the rest of the set. But, it is the work of Bennett et al. [11], [12] that clearly changes the perspective for approaching large-margin classifiers from the statistical learning theory to geometric reasoning. In their work, the authors deduce the primal and dual formulations for the support vector machine using the geometric concept of the convex hull.

The intuitive geometric reasoning behind the well-known support vector machines technique comes from the maximization of the margin, defined as the minimum distance from the closest data points to the boundary. Although it is simple to understand this concept when the optimal separation is a hyperplane, it becomes much more complex in front of nonlinear boundaries. The most well-known strategy to deal with this issue is the kernel approach that represents a change in the metric space when computing the margin. However, besides this last approach, these concepts are not enough to fully define a general nonlinear boundary since there is much uncertainty due to the not so close points—points that define a classification boundary but are far from the support region with minimum margin. In the toy example of Fig. 1a, any decision boundary lying in the shadowed area satisfies the maximum minimum margin property[1] (the margin is shown as d in Fig. 1). Therefore, other considerations must be taken into account to define a unique solution. When creating a new algorithm, we require it to be able to handle noisy data and identify approximate patterns. Therefore, it seems reasonable to use this notion of robustness to noise to define a solution.

In this paper, we propose the concept of *characteristic boundary points* (CBP)—points that belong to the optimal boundary following certain definitions of robustness and margin. Fig. 1b shows an example of the CBP (black dots). Intuitively, an hyperspherical "area of influence" is laid around each data point, characterizing the amount of noise that a point is able to handle without ambiguity. The spatial locations where those "areas of influence" from different classes collide define the characteristic dichotomizing points. Observe that a robust boundary is found joining all the CBP. We approximate this boundary by a piecewise linear function. Based on the CBPs, we create a set of hyperplanes that are locally optimal from the point of view of the margin. These are assembled linearly into the final decision boundary using a Tikhonov regularization framework. The use of the Tikhonov framework is a conservative approach to the solution in which we desire to give the maximum expressivity (nonlinear behavior) to the final border while being able to control the robustness in front of noise with a parameter λ. In this context, Tikhonov regularization smoothes the classification boundaries defined by the piecewise ensemble, yielding a classification rule less prone to overfitting. Due to the local scope of each individual linear classifier and the simplicity of the method, we propose an algorithm that allows to linearize the computational complexity of the technique and cope with large scale and online learning problems.

The layout of the paper is the following: Section 2 describes the CBPs and proposes an algorithm to compute them by means of the proximity neighboring rules (in particular, we use the Gabriel neighboring rule [6]). The definition of the CBPs allows for the creation of a set of robust, locally optimal, linear classifiers that are combined into a λ-smooth additive model by means of the Tikhonov regularization framework (see Section 3). In Section 4, the algorithm for linearizing the computational complexity of the method is presented. Section 5 shows the validation of the technique on the UCI database, comparing our approach with several state-of-the-art classifiers like Adaboost [1] or kernel Support Vector Machines [2]. Additionally, results on an online scenario, large-scale data sets, and on six real object recognition problems are provided. Finally, Section 6 concludes the paper and future research lines are proposed.

2 On the Definition of the Nonlinear Decision Boundary

In the previous section, we commented that, in general, achieving a maximum minimum margin is not enough to define a unique solution to the classification problem. Therefore, informally, the desired boundary can be defined as the one that simultaneously maximizes the margin of the *closer* points to the boundary and its robustness to noise.

1. Notice that, in the case of kernel-based SVM, the kernel parameterization fully defines one solution in the shadowed area.

- *O. Pujol is with the Departament de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Edifici Històric, Gran Via de les Corts Catalanes 585, 08007 Barcelona, Spain. E-mail: oriol_pujol@ub.edu.*
- *D. Masip is with the Department of Computer Science, Multimedia, and Telecomunications, Universitat Oberta de Catalunya, Rambla del Poblenou 156, 08018 Barcelona, Spain. E-mail: dmasipr@uoc.edu.*

Manuscript received 23 Oct. 2008; revised 22 Jan. 2009; accepted 22 Jan. 2009; published online 27 Jan. 2009.
Recommended for acceptance by A. Martinez.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-10-0735.
Digital Object Identifier no. 10.1109/TPAMI.2009.31.

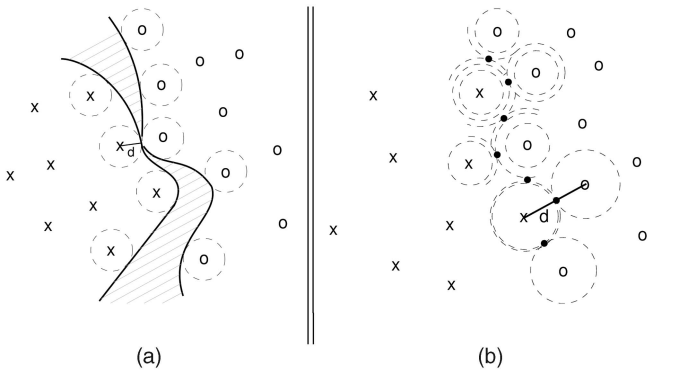

Fig. 1. Illustrative examples. (a) Any decision boundary in the shadowed area has minimum maximum margin. (b) Intuitive idea of the characterizing boundary points.

2.1 The Characteristic Boundary Points

Given a labeled training set of M points $S = \{(x_i, l_i)\}$, where $x_i \in \mathbb{R}^d$ belonging to class $l_i = l(x_i) \in \{+1, -1\}$, $i = 1 \dots M$, our **starting hypothesis** is that a characteristic boundary point $x_{cp}^{i,j} \in \mathbb{R}^d$ is defined between any two training points (x_i, x_j) that fulfill the following conditions:

- **Necessary condition:** Both points (x_i, x_j) are from different classes

$$x_i \in S \mid l_i = +1 \quad \text{and} \quad x_j \in S \mid l_j = -1.$$

- **Answer to the definition of the proximity to the optimal boundary:** There is no closer example to the candidate boundary point x_{cp} than the ones that define it, namely, (x_i, x_j).

  Given any point $p : \{p \in \mathbb{R}^d \mid (p, l) \in S\}$, then

$$\|x_i - x_{cp}\| \leq \|p - x_{cp}\|, \tag{1}$$

$$\|x_j - x_{cp}\| \leq \|p - x_{cp}\|. \tag{2}$$

- **Answer to the definition of the robustness to noise notion:** Assuming that no geometrical information about the expected noise is considered, the candidate boundary point is located at the maximum distance from both generating points (the middle point between x_i and x_j)

$$x_{cp}^{i,j} = \frac{1}{2}(x_i + x_j). \tag{3}$$

  The last property defines the robustness policy in front of new data—it gives to each data point in the training set the maximum margin locally. Alternatively, this can be seen as giving to each point in the data set a maximum "area of influence."

2.2 Locating the Closest Points to the Boundary

Let us take two pairs of points that have one of them in common, $\{(x_i, x_j), (x_k, x_j)\}$, as shown in Fig. 2. By definition, $x_{cp}^{i,j} = (x_i + x_j)/2$ is a characteristic boundary point only if $\|x_i - x_{cp}^{i,j}\| \leq \|x_k - x_{cp}^{i,j}\|$ (1). Note that the locus of the space in which $\|x_i - x_{cp}^{i,j}\| = \|x_k - x_{cp}^{i,j}\|$ defines a hypercircle centered on $x_{cp}^{i,j}$ with radius $\|x_i - x_j\|/2$. Then, $x_{cp}^{i,j}$ is a characteristic boundary point and the inequality will only hold if x_k is further than $\|x_i - x_j\|/2$ from $x_{cp}^{i,j}$. That is, there are no closer points to the candidate x_{cp} than x_i or x_j. This rule is known as the *Gabriel neighboring rule* [6].

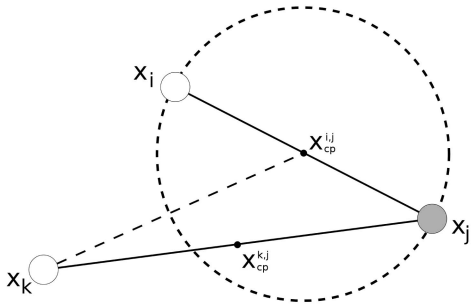

Fig. 2. Illustration of the definition of characteristic boundary points. $x_{cp}^{i,j}$ is a CBP since there is no other point inside the hypersphere centered in it with radius $\|x_i - x_j\|/2$.

Given a pair of samples x_i and x_j that form a candidate CBP ($x_{cp}^{i,j}$), the algorithm for computing the CBP must check that there does not exist any other sample x_k that is closer to $x_{cp}^{i,j}$ than x_i and x_j. The naive approach for solving this problem has a complexity of $\mathcal{O}(dM^3)$. However, the problem of computing the Gabriel neighboring rule can be easily solved without explicitly computing the point $x_{cp}^{i,j}$ given the distances among x_i, x_j, and x_k. Thus, one can first compute the distances for each pair of samples in $\mathcal{O}(dM^2)$ and then search for $x_{cp}^{i,j}$. This reduces the complexity of the algorithm to $\mathcal{O}(M^3 + dM^2)$. Algorithm 1 shows the explicit computation of the CBP using this method.

Input: Set of data points $\mathcal{S} = \{x_i, l_i\}$, $x_i \in \mathbb{R}^d$ belonging to class $l_i \in \{+1, -1\}$
Output: Set of pairs of indexes $\{(i, j)\}$ corresponding to the data pair that defines the CBP
Compute the Euclidean distance $d_{i,j}$ from each point x_i to x_j ;
Initialize the set of indexes that define the CBP, $E = \{\}$;
foreach $x_i | l_i = +1$ **do**
  foreach $x_j | l_j = -1$ **do**
    foreach $x_k | (x_k, l_k) \in \mathcal{S}$ **do**
      Compute the distance $d_{k,m}$ from the sample x_k to the central point $x_m = \frac{(x_i + x_j)}{2}$ as follows:

$$d_{k,m} = \sqrt{d_a + d_b}$$

$$d_a = d_{i,k}^2 \Big(1 - \frac{d_{j,k}^2 - d_{i,k}^2 - d_{i,j}^2}{-2 d_{i,k} d_{i,j}}\Big)$$

$$d_b = \Big(d_{i,k} \big(\frac{d_{j,k}^2 - d_{i,k}^2 - d_{i,j}^2}{-2 d_{i,k} d_{i,j}}\big) - \frac{d_{i,j}}{2}\Big)^2$$

      if $\nexists x_k | d_{k,m} < \frac{d_{i,j}}{2}$ **then**
        add the pair of indexes corresponding to $\{x_i, x_j\}$ to the set E, $E = E \bigcup (i, j)$
      end
    end
  end
end

3 Geometry-Based Ensembles

According to the bias and variance error decomposition, the definition of a classification scheme should enforce two properties: expressivity—most arrangements of sample points in the feature space can be approximated—and generalization—it captures the underlying model from the data source so that, given new unseen input examples, they are correctly labeled.

A classifier designed to find a boundary using the characterizing boundary points, x_{cp}, is by its own definition expressive since each CBP locally captures the variability of the training set. Therefore, the combination approach selected must ensure the generalization performance. One of the simplest ways of modeling the nonlinear boundary from the set of CBP is a piecewise linear ensemble solution. The ensemble is created by means of an

TABLE 1
Detailed Process for Computing the Geometry-Based Ensemble

- Compute the CBP on the training set, $\{x_{cp}^{i,j}\}, \quad \forall i,j \in 1 \ldots M, \quad i \neq j$
- Create a set of base classifiers based on $\{x_{cp}^{i,j}\}$,

$$\pi_{x_{cp}^{i,j}}(x) = (x - x_{cp}^{i,j})\vec{n_j}, \quad x_{cp}^{i,j} = \frac{x_i + x_j}{2}, \quad n_{\vec{x_{cp}^{i,j}}} = \frac{x_i - x_j}{\|x_i - x_j\|}$$

- Evaluate the base models on the data set to find matrix A,

$$A(k,i) = \text{sign}(\pi_k(x_i)), k \in \{1 \ldots |\{x_{cp}\}|\}, i \in \{1 \ldots M\}$$

- Find the $\lambda - solution$ of the weights solving the following optimization problem,

$$\alpha_\lambda = \arg\min_\alpha (\|A\alpha - l\|_2^2 + \lambda^2 \|(\alpha - \alpha^*)\|_2^2)$$

additive model $\Pi : \mathbb{R}^d \rightarrow \mathbb{R}$ of base classifiers $\pi_k(x)$ related with the characterizing boundary points

$$\Pi(x) = \sum_{k=1}^{N} \alpha_k \, \text{sign}(\pi_k(x)), \tag{4}$$

where N is the number of CBP ($N = \left|\{x_{cp}^{i,j}\}\right| \leq M^2/2$) and sign stands for the signum function. The final decision rule can be simply obtained by thresholding the ensemble combination

$$\hat{l} = \text{sign}(\Pi(x) - \alpha_0),$$

where $\hat{l}$ is the estimated label and α_0 is the global threshold value.

3.1 Defining the Base Classifiers

Observe that, in an ad hoc sense, the hyperplane located at a characterizing boundary point defines a locally optimal margin classifier—since there is no other point in a neighboring region around the CBP that can alter the decision boundary.

Thus, given the pair of points (x_i, x_j), the base linear classifier is defined as a hyperplane in the following way:

$$\pi_{x_{cp}^{i,j}}(x) = (x - x_{cp}^{i,j})\vec{n_j}, \quad x_{cp}^{i,j} = \frac{x_i + x_j}{2}, \quad \vec{n_{x_{cp}^{i,j}}} = \frac{x_i - x_j}{\|x_i - x_j\|},$$

where $x_i : \{(x_i, l_i) \in S \mid l_i = +1\}$ and $x_j : \{(x_j, l_j) \in S \mid l_j = -1\}$ correspond to the elements of class labeled as $+1$ and -1, respectively. Notice that, by convention, the normal vector that defines the hyperplane, $\vec{n_j} \in \mathbb{R}^d$, always points to class $+1$. A positive value of $\pi_{x_{cp}^{i,j}}(x)$ indicates that an example belongs to class $+1$ and to -1 otherwise.

3.2 Weights Estimation Using Tikhonov Regularization

The influence of each base classifier in the ensemble is governed by the weighting vector α in (4). The estimation of α is a classical example of an ill-posed problem. One of the most common approaches for stabilizing this problem is the use of further information about the desired solution. This is the purpose of regularization. Changing our formulation into the matrix form, we are looking for the solution of the linear system of equations

$$A\alpha = l, \quad A \in \mathbb{R}^{M \times N}, \quad \alpha \in \mathbb{R}^N, \quad l \in \mathbb{R}^M, \tag{5}$$

where $A(k,i) = \text{sign}(\pi_k(x_i)), k \in \{1 \ldots |\{x_{cp}\}|\}$, $i \in \{1 \ldots M\}$, and l is the label vector of the training set. Notice that the matrix has an ill-determined rank. Moreover, the terms $\{A, l\}$ can be perturbed by an error. One meaningful approximation to the solution of (5) is to compute the minimal-norm least squares solution. Given the limited number of base classifiers obtained by means of the CBP, we opt for a conservative approach to the solution in which we retain as much expressivity as possible (nonlinear behavior) while being able to control the robustness in front of noise with a parameter λ. Thus, we can use the Tikhonov regularization framework, based on requiring that the 2-norm, or an appropriate seminorm, of the solution is small. Thus, the lambda solution of the system is the minimizer of

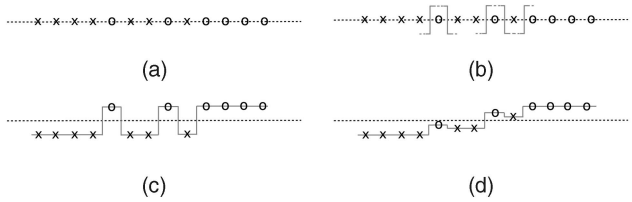

Fig. 3. One-dimensional example of the ensemble using OGE. (a) The toy problem. (b) Location of the base linear decision rules. (c) Example of an expressive solution (the height of each data sample corresponds to the value assigned by the ensemble $\Pi(x)$ without thresholding). (d) Example of a smoother solution.

$$\alpha_\lambda = \arg\min_\alpha (\|A\alpha - l\|_2^2 + \lambda^2 \|(\alpha - \alpha^*)\|_2^2), \tag{6}$$

where λ controls the weight of the residual norm and α^* is an initial estimate of the solution. Note that λ controls the sensitivity of α_λ to the perturbations in A and l. In practice, the λ-regularized ensemble solution is expected to reduce the effects of the overfitting problem inherent in the local classifiers selection based on neighboring rules.

By convention, we take one direction in the decision rule of each base classifier and we want the optimization process to respect this convention. Therefore, we must guarantee the nonnegativity of the solution. This means that we do not allow the optimization process to change the decision rule of a particular classifier. In addition, we would like to reduce the number of marginal classifiers as much as possible without hindering the global performance. For all of these reasons, we opt for the aggressive but effective solution of setting all negative weights to zero in the final solution.[2] Additionally, if α^* is set to zero, we guide the optimization process to find the solution ensuring that a high number of weights are zero while preserving its smoothness.

In this paper, we use SVD and filter factors for solving the regularized problem. In order to find the optimal value of λ, we use a model selection technique by means of a 5-fold cross-validation [8] on the data associated to the training set.

Summarizing the method, Table 1 shows the general procedure for computing the optimized geometry-based ensemble.

Fig. 3 shows an intuitive example of the optimized geometric ensemble (OGE). Fig. 3a shows the original one-dimensional toy problem. In 1D, the CBPs are found as the middle points between elements of the different classes, as we see in Fig. 3b. Each CBP defines a local decision $\pi_{x_{cp}}(x)$. These are assembled in a final ensemble $\Pi(x)$. Figs. 3c and 3d show two possible results of the ensemble. The height associated to each data sample corresponds to the value assigned by the ensemble $\Pi(x)$ without thresholding. Fig. 3c shows a very expressive solution, while Fig. 3d displays a smoother solution. Observe that the adequacy of each solution depends on the definition of the problem (i.e., assumptions about the inaccuracy of the labels, noise in the data examples, etc.).

4 Linearization Meta-Algorithm for OGE (LOGE)

The weakest spots of the method come from the computational complexity in obtaining the characteristic boundary points and the lack of control in the final model complexity. As we noted in the former section, the algorithm for finding the CBPs can be computed in $\mathcal{O}(M^3 + dM^2)$ and the number of classifiers in

2. Note that many sequential optimization procedures, such as LASSO, ensure nonnegativity in the solution by setting all negative coefficients to zero at each step of the process.

the final model is upper bounded by $M^2/2$. Thus, we propose a linearization method with an explicit model complexity control based on the subdivision and aggregation of smaller problems. The algorithm can be seen as a hierarchical process in which each node aggregates and solves a set of problems and yields a partial solution of a certain desired complexity. The aggregation of the different solutions is performed by joining the examples that create the CBPs for each of the subproblems involved. Due to the local definition of the classifiers, this method is expected to capture the same distribution of CBPs as the complete problem when the number of aggregated examples is high enough. Moreover, since the complexity of the partial solutions is fixed, the computational cost of the method becomes linear with the number of subproblems solved.

Input: Set of data points
Output: Trained classifier $\tilde{c}$
Initialize $L = \varnothing$, $\tilde{c} = \varnothing$;
while *iter* $\leq$ *Niters* **do**
  Create a problem p by selecting k data samples from each class;
  Obtain the pair solution-problem $(\tilde{p}, p)$ resulting of the optimization of problem p;
  $L_1 = L_1 \bigcup (\tilde{p}, p)$;
  foreach *level of the list* L **do**
    if $|L_i| \geq \kappa$ *(aggregation factor)* **then**
      Create a new problem p_{join} joining the data samples from the problems stored at level i;
      $L_i = \varnothing$;
      [Optional] **if** $complexity(p_{join}) > \mu$ **then**
        Select representative data of problem p_{join} (in OGE we select μ data samples according to the alpha value of the CBP they create)
      end
      Obtain the pair solution-problem $(\tilde{p_{join}}, p_{join})$ resulting of the optimization of the problem p_{join};
      $L_{i+1} = L_{i+1} \bigcup (\tilde{p_{join}}, p_{join})$;
    end
  end
end
[Flush] **foreach** *level* i *from* L **do**
  Create a temporal solution c_i by joining elements of the solutions from the problems stored at level i;
  $\tilde{c} = \tilde{c} \bigcup c_i$;
end

Algorithm 2 allows for the linearization of the computational cost of the OGE. Consider the use of groups of k input samples, a fixed aggregation factor κ that defines the maximum number of subproblems to be aggregated at each level of the system and a maximum allowed complexity of the solution model μ—in OGE, μ is measured as the number of CBPs involved in the solution. The algorithm can be seen as a tree structure in which each node aggregates (optimizes) κ problems and yields a μ-complexity solution. The algorithm starts with small subproblems of size k. The optimization of each subproblem yields a partial μ-solution model. When κ models at that level are obtained, they are optimized into a new μ-solution model of a higher level. Observe that the algorithm has an optional step for controlling the complexity of the model that selects the μ most relevant CBPs according to their α value. When there are no more data available, the system flushes the different levels aggregating the solutions into a final model. As a result, if the optimization time of a problem of size μ is equal to τ, then, given an M-size input problem, the worst-case solution is given by a binary tree $(k = 2, \kappa = 2)$. Therefore, the complexity of the system is $\mathcal{O}(\frac{M}{k}\tau)$, where $\tau \approx \mathcal{O}(\mu^3 + d\mu^2)$. Note that this linearization comes at the cost of having a predefined complexity on the output classification model.

5 Experiments and Results

In this section, we describe and discuss the experiments we have performed with data from three different environments. First, we discuss our approach on toy problems. Second, data from the UCI repository is used to compare and validate our method. Finally, we apply this approach to six different real computer vision problems.

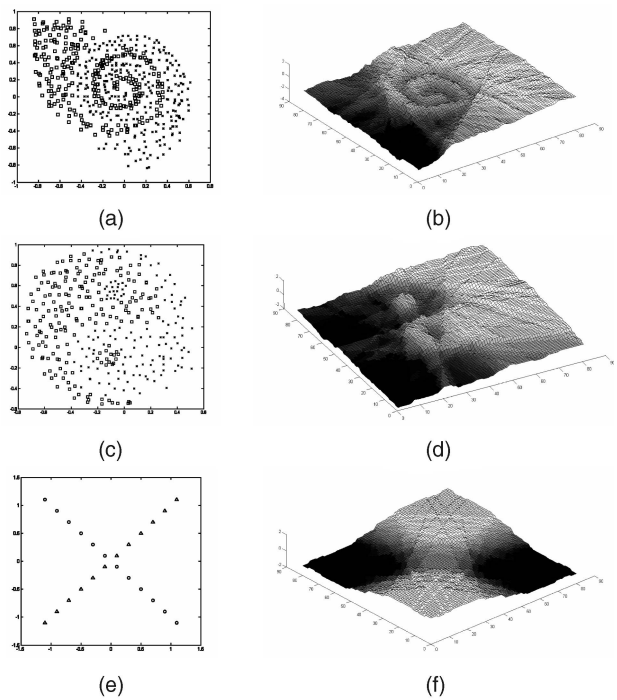

Fig. 4. Ensemble surface on three toy problems. (a), (c), and (e) Original data sets. (b), (d), and (f) Surfaces of the result of the ensemble.

5.1 Toy Problems

In this section, we pretend to illustrate qualitatively the behavior of the proposed strategy. For this reason, we have designed three toy problems with different particularities. The problems (see Figs. 4a, 4c, and 4e) have been designed to show the expressivity of the resulting classifier. They are the spiral, the yin-yang, and the extended xor problems. On the right side of each figure, we plot the ensemble surface scoring before thresholding, $\Pi(x)$. In Figs. 4b, 4d, and 4f, one can observe that the ensemble makes qualitatively good approximations of the data points. Moreover, the behavior in the areas without data is relatively smooth.

5.2 Validation on UCI Database

In order to validate quantitatively our approach, we begin with an analysis using the standard UCI database [9]. The data sets selected can be seen in Table 2.

Experimental settings:

- For this experiment, we have chosen a set of state-of-the-art classifiers in order to empirically evaluate the performance of the methods: k-nearest neighbors, support vector machines with a radial basis function kernel, linear support vector machines, discrete Adaboost ensemble with decision stumps, relevance vector machines [16], and a recently proposed Bayesian classifier [17].
- The experiments have been performed using the same 10 randomized runs of stratified 10-fold cross validation for all the classifiers.
- The parameters (C, σ, γ) of the SVM and RVM, the parameter k of the k-NN, and the parameter λ of our approach have been found using 5-fold cross-validation on the training set using the same folds for all the classifiers. As we commented, α^* is set to zero in all the experiments. The Discrete Adaboost is allowed to use a maximum of 1,000 decision stumps.

TABLE 2
UCI Data Sets Used in the Experiments with the Number of Examples and Dimensionality d

Database	code	Examples	d	Database	code	Examples	d
Breast Winsc.	(bcw)	699	10	Statlog Heart	(shd)	270	13
Bupa liver	(bld)	345	6	Tic Tac Toe	(ttt)	958	9
Heart Clev.	(hec)	920	13	Spect	(spt)	267	22
Ionosphere	(ion)	351	34	Echocardiogram	(ech)	132	13
Breast	(bre)	569	32	New thyroid	(nth)	215	5
Pima indians	(pid)	768	8	Voting records	(vot)	435	16
Sonar	(snr)	208	60	Monks complete	(mon)	432	7
Statlog Au Credit	(sac)	690	14	Credit	(cre)	690	15

- All data sets have been normalized with respect to the mean and variance to avoid penalizing the nearest neighbors approach.

Table 3 displays the resulting mean accuracy. For future comparison purposes, we also include the confidence interval at 95 percent testing for statistical significance using a corrected two-tailed t-test. The table must be read in the following way: There is a • marker on the right side of the result that achieves the absolute higher mean accuracy. The marker ∘ shows the method with the poorest performance. The table also shows the mean rank value according to the accuracy.

From Table 3, it must be noted that OGE performs generally better than most of the commonly used classifiers. Taking into account the number of times a method achieves the best mean performance, OGE is the first choice in 6 out of 16 data sets, followed by RVM with 4, k-NN with 3, SVM.rbf with 2, and Ham with 1. Regarding the worst performances, Ham is the last option in 5 data sets, RVM in 4, k-NN and Adaboost in 3, and SVM.lin in 2. Observe that OGE and SVM.rbf are never the last choice. Following the guidelines proposed by Demsar [18], we test for global statistical significance by means of the Friedman's test. We proceed as follows: First suppose that the differences in Table 3 are due to randomness. In order to accept/reject this hypothesis, we use the Friedman statistic. The Friedman's Test with Iman and Davenport correction obtains $F_F = 2.43$. With seven methods and 16 data sets, F_F is distributed according to the F distribution with 6 and 90 degrees of freedom. The critical value of $F(6, 90)$ for $\alpha = 0.05$ is 2.20. Since the obtained corrected value is higher than the critical value, we can reject the null hypothesis. This means that the rankings obtained convey significant information. Observe that the mean ranking (random ranking) is 3.5. The only methods that achieve a ranking below the average ranking value are SVM.rbf and OGE.

We continue the analysis comparing our method with the different representative samples of each of the families of classifiers according to the concepts related in OGE. In order to perform this comparison, we use the Bonferroni-Dunn test [18], which shows a much greater power as a posthoc test when classifiers are compared to a control classifier. For each test, we recompute the ranking between OGE and the members of each family.

The proposed method is a member of the ensemble family. Thus, comparing with Adaboost, the rank difference between both techniques is 0.750, which is higher than the critical difference for this test, $CD_{2,0.10} = 0.411$. Thus, OGE is statistically better than Adaboost. From the point of view of regularization frameworks without kernels, when compared with SVM.lin, the rank difference between both techniques is 0.750, which tells that OGE is statistically better than SVM.lin ($CD_{2,0.10} = 0.411$ in this test). Regarding nonlinearity, comparing with kernel techniques, SVM.rbf and RVM, the rank differences are smaller than the critical difference, $CD_{3,0.10} = 0.693$. Thus, with this test, we cannot distinguish statistically among these techniques. With respect to the remaining techniques, OGE and Ham have a difference value of 0.781, which shows that OGE statistically outperforms Ham. On the other hand, OGE and k-NN have a difference value of 0.531. Thus, at 90 percent, both techniques are barely statistically indistinguishable. However, reducing the statistical significance to 85 percent, OGE performs statistically better than k-NN (p-value $= 0.138$).

5.3 Relationship to Graph-Driven Algorithms

The notion of the Gabriel neighboring rule in supervised environments has been previously used in literature to edit the training set to drive a nearest neighbor classifier [3], [4]. For the sake of completeness, we experimentally compare the Gabriel edited nearest neighbors approach (G-NN) and the k-NN rules on the same 16 UCI data sets using the same settings as in the former experimental section of the paper (see Table 4). In brackets, the value of k is displayed. Observe that G-NN performs very similar to k-NN. When compared to OGE on the same folds, the OGE approach fares better in 11 out of the 16 data sets.

5.4 Time Comparison for OGE Basic Algorithm

In order to experimentally assess the computational performance of the basic method, in Fig. 5 we show the mean training and test time rankings after running 100 times the different algorithms on

TABLE 3
Comparison of Different Learning Techniques on Several UCI Data Sets

Database	k-nn	SVM.rbf	SVM.lin	Adaboost	OGE	RVM	Hamsici
(bcw)	97.03 ± 0.43	97.01 ± 0.38	96.65 ± 0.39	95.01 ± 0.51	97.09 ± 0.41•	94.61 ± 1.31∘	96.40 ± 0.41
(bld)	63.54 ± 1.31	66.34 ± 1.48	68.63 ± 1.23	69.00 ± 1.53	69.74 ± 1.31	71.46 ± 1.47•	62.43 ± 1.44∘
(hec)	82.87 ± 1.23	80.70 ± 1.31	82.40 ± 1.26	75.93 ± 1.60, ∘	82.93 ± 1.23•	79.93 ± 1.35	82.83 ± 1.23
(ion)	86.47 ± 0.91∘	94.86 ± 0.64	87.89 ± 0.95	90.50 ± 1.05	90.92 ± 0.90	94.95 ± 0.82•	86.47 ± 0.95∘
(bre)	96.71 ± 0.47	97.81 ± 0.35•	97.03 ± 0.48	96.24 ± 0.49	97.69 ± 0.40	95.04 ± 0.58∘	96.67 ± 0.43
(pid)	66.57 ± 1.40∘	68.47 ± 1.66	73.17 ± 1.49	70.17 ± 1.63	72.33 ± 1.41	75.90 ± 1.06	76.06 ± 0.96•
(snr)	87.09 ± 1.26•	86.82 ± 1.31	75.05 ± 1.74	83.27 ± 1.48	78.68 ± 1.63	84.73 ± 1.65	74.41 ± 1.77∘
(sac)	86.09 ± 0.80	85.20 ± 0.82	85.24 ± 0.92	84.43 ± 0.92	87.07 ± 0.90•	83.88 ± 0.84∘	85.91 ± 0.91
(shd)	82.74 ± 1.25	82.74 ± 1.15	83.63 ± 1.35	81.63 ± 1.53∘	84.41 ± 1.24•	82.15 ± 1.39	84.00 ± 1.30
(ttt)	84.64 ± 0.58	92.57 ± 0.46	64.95 ± 0.01	65.97 ± 0.97	69.08 ± 0.93	94.89 ± 0.49•	60.20 ± 1.39∘
(spt)	85.83 ± 1.05	90.97 ± 0.96	80.14 ± 1.10∘	89.50 ± 0.96	81.31 ± 1.22	93.81 ± 0.70•	89.19 ± 0.98
(ech)	88.57 ± 2.42 ∘	92.57 ± 2.05	96.57 ± 1.44	94.00 ± 1.39	98.57 ± 0.84•	98.00 ± 0.98	91.29 ± 1.82
(nth)	96.18 ± 0.71•	94.91 ± 0.82	89.23 ± 1.03	94.55 ± 0.72	95.18 ± 0.78	95.91 ± 0.89	86.32 ± 1.09∘
(vot)	93.64 ± 0.57	96.52 ± 0.65•	95.95 ± 0.64	94.95 ± 0.63	95.25 ± 0.63	83.94 ± 1.12∘	96.00 ± 0.55
(mon)	91.27 ± 1.29•	89.20 ± 0.68	67.25 ± 1.12 ∘	74.84 ± 0.92	78.55 ± 1.80	88.09 ± 0.98	86.46 ± 1.76
(cre)	86.02 ± 0.82	85.92 ± 0.83	86.75 ± 0.80	84.20 ± 0.84∘	86.97 ± 0.79•	86.70 ± 0.73	86.62 ± 0.78
Rank	4.03	3.34	4.25	5.25	2.69	4.00	4.44

TABLE 4
Results Using k-NN and NN with a Gabriel Edited Training Set

	bcw	bld	hec	ion	bre	pid	snr	sac	shd	ttt	spt	ech	nth	vot	mon	cre
G-NN	97.01(18)	63.54(18)	81.87(11)	86.36(1)	96.72(5)	66.30(19)	87.09(1)	86.09(19)	82.79(5)	84.64(6)	85.83(1)	88.57(12)	96.14(1)	93.64(5)	90.89(5)	86.02(8)
k-NN	97.03(10)	63.54(18)	82.87(13)	86.47(1)	96.71(4)	66.57(17)	87.09(1)	86.09(19)	82.74(5)	84.64(5)	85.83(1)	88.57(1)	96.18(1)	93.64(1)	91.27(5)	86.02(8)

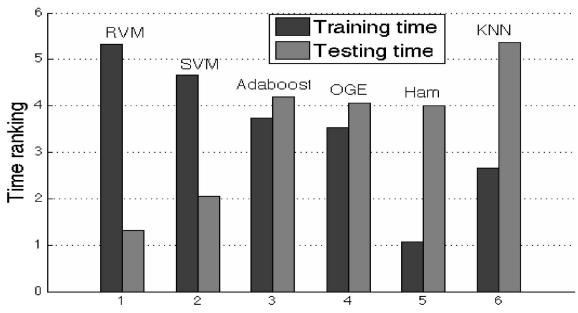

Fig. 5. Comparison of the mean training and testing times on the UCI data sets (in rank). The training time includes a 5-fold cross validation for adjusting the parameters of each algorithm when needed.

all data sets. In order for this to be a fair comparison, we have performed this evaluation using the same machine (P5 2GHz with 1 GHz RAM). In particular, we have used the codes provided by the authors of the Hamsici et al. [17], PRTools' [22], nearest neighbor, OSUSVM [21], and Tipping's code for RVM [16]. Observe that there is a delicate trade-off between the training time and the test time. Usually, a large training time resolves into a small test time and vice versa. Note that our method stands in the middle with moderate training and test times.

5.5 Online and Large-Scale Learning

Online learning refers to the property of a classifier self-adapting to the arrival of new data without revisiting previous training data. The linearization algorithm for OGE allows it to handle online learning by setting $k = 1$. In order to validate the classifier in this scenario, we feed the system one sample at a time using the same experimental settings described formerly. Table 5 shows the accuracy results using the system as online learner on the 16 UCI data sets. Additionally, the mean training time for OGE and LOGE are provided in Table 5. Observe that the training time is drastically reduced and that the accuracy performance is similar to the original OGE.

Finally, we have performed experiments with large-scale data sets using three common databases found in the large-scale literature: Covertype1 [9], Shuttle [9], and Reuters-RCV1 CCAT [20]—the class CCAT versus the rest. The description of the data sets, accuracy, time results, and relative complexity of the resulting model with respect to the training set (RC) are shown in Table 6. Observe that the method allows us to cope with this kind of problem and, due to the complexity control, the resulting model has a very low complexity. Compared to the state of the art, on the Reuters data set, the accuracy is lower than the highest one reported in literature (93 percent). However, Covertype1 and Shuttle results are as good as the ones reported, 72 percent and 99 percent, respectively.

5.6 Results on Visual Object Recognition Problems

In order to further validate the results of our technique, we have applied it to six real computer vision and signal processing problems: gender recognition based on internal features of faces, intravascular ultrasound (IVUS) image tissue characterization, speed traffic sign recognition, Chagas' disease myocardial damage severity detection, old musical scores clef classification and action recognition using 3D accelerometer data from a wearable device.

Problem description:

- **Gender recognition problem (fac).** We use the publicly available ARFace database [10]. The AR data set consists of 26 images per subject from 126 different subjects (70 male and 56 female) acquired in two separate sessions. Face images are preprocessed by manually selecting the center pixel of each eye, and each sample has been rotated and scaled. A final 36×33 centered thumbnail with only the internal face features is used in the experiments.
- **Tissue characterization in Intravascular Ultrasound images [14] (ivu).** Intravascular Ultrasound (IVUS) images show the morphological and histological properties of a cross section of the coronary arteries. The physicians are interested in characterizing the presence of a large soft (lipidic) core with a thin fibrous cap. The RF signals are acquired using a 12-bit Acquiris acquisition card with a sampling rate of 200 MHz. Each IVUS image consists of a total of 256 A-lines (ultrasound beams) and a length of 6.5 mm. The RF is acquired from patient pullback sequences in vivo. The data set comprises 500 regions of tissues from 11 patients that have been previously segmented and labeled by the physicians. For each region, the corresponding power spectrum of the radio frequency data is obtained and modeled with a 21 degree autoregressive model. As a result, we obtain 200 features per region.
- **Recognition of speed traffic signs (spd).** This is one of the most difficult tasks in road mapping and guided navigation systems. In real applications, it is usual to previously remove speed signs from the training set and distinguish among them in a posterior processing step. In this problem, we have a set of 32 different signs split in two groups (speed versus nonspeed). The data set was extracted from eight car drive records at different locations, highways, and local roads. The total number of examples is 2,217. Each extracted sign image measures 35×35 pixels. Each pixel is considered as an object feature. Thus, the feature vector dimensionality is 1,225.
- **Chagas' disease myocardial damage severity detection [13] (cha).** The application is focused on distinguishing two clusters of patients affected by the Chagas' disease. Chagas' disease is an infectious illness characterized by alterations in the cardiovascular system caused by the Tripanosoma Cruzi parasite. The aim of the application is to distinguish between early stages of the disease in which the myocardial damage is small but the patient shows abnormal ECG and medium and late stages, where the damage is severe. High-resolution electrocardiogram (HRECG) is obtained from 107 patients. Two basic phenomena are used for defining the features: the QRS signal and the ventricular late potential (VLP) presence. From these, 16 features are obtained (upward and downward slopes of the QRS, QRS interval mean and variance, root-mean-square value in a 40 ms interval, etc.
- **Old musical scores clef recognition (cle).** The data set of clefs and accidentals is obtained from a collection of

TABLE 5
Comparison of Time and Accuracy Using LOGE in an Online Scenario

	bcw	bld	hec	ion	bre	pid	snr	sac	shd	ttt	spt	ech	nth	vot	mon	cre
OGE Train	5.37	2.29	1.20	2.52	0.67	19.69	3.11	10.66	0.88	165.83	27.36	0.01	0.08	7.31	24.84	13.86
LOGE Train	0.29	1.18	0.99	0.75	0.81	0.61	2.47	4.80	0.98	12.30	2.56	0.02	0.08	2.30	3.91	1.91
OGE Accuracy	97.09	69.74	82.93	90.92	97.69	72.33	78.68	87.07	84.41	69.08	81.31	98.57	95.18	95.25	78.55	86.97
LOGE Accuracy	96.90	67.71	79.91	89.89	95.12	69.20	78.14	84.55	83.81	70.23	79.75	93.05	92.54	95.30	82.36	84.48

TABLE 6
Results on Large-Scale Data Sets

Dataset	Accuracy	Training	Features	Test set	Training time	RC
Covertype1	72.31%	11340	54	565892	6.7 s	$5.2 \cdot 10^{-2}$
Shuttle	99.86%	43500	9	14500	24.7 s	$3.4 \cdot 10^{-3}$
Reuters CCAT	87.5%	781265	47152	23149	231.0 s	$2.5 \cdot 10^{-4}$

modern and old musical scores (19th century) of the Archive of the Seminar of Barcelona. The data set contains a total of 4,098 samples among seven different types of clefs and accidentals from 24 different authors. For this application, we extract the Blurred Shape Model [19] using a grid size of 8×8. We reduce the dimensionality using PCA keeping 95 percent of the energy.

- **Action recognition using a wearable device (acc).** This application consists in action recognition using data provided by a 3D accelerometer on a wearable device. We distinguish between the climbing and moving action. The data set is recorded from 10 runs of half an hour each using a wearable device in two natural scenarios by 10 diferent people. From the raw data, we compute 363 features corresponding to the mean value, energy, FFT values, and cross correlation between acceleration axis in windows of 40 samples with 20 samples of overlapping.

We have chosen a representative classifier from each of the meaningful families according to the results obtained in the former sections with the same experimental setting described in Section 5.2.

Table 7 shows the figures corresponding to the results of each experiment. Observe that OGE is the first choice in four out of six applications, obtaining the best overall ranking value. The second choice is SVM.rbf that closely follows our approach, achieving the best performance in three out of the six applications. Note, however, that it is the last choice in the (ivu) application and the third choice in (cle) and (acc). Adaboost is close behind SVM.rbf, and SVM.lin is only comparable to the rest of strategies in the (ivu) application.

TABLE 7
Results on Real Computer Vision and Pattern Recognition Applications

Database	SVM.rbf	SVM.lin	Adaboost	OGE
(fac)	$86.44 \pm 2.02\bullet$	$83.42 \pm 2.64\circ$	85.34 ± 2.16	85.89 ± 2.72
(ivu)	$85.56 \pm 0.86\circ$	86.66 ± 0.85	86.82 ± 0.82	$87.74 \pm 0.88\bullet$
(spd)	$91.60 \pm 2.01\bullet$	$81.40 \pm 2.92\circ$	88.20 ± 2.06	90.60 ± 3.67
(cha)	$81.04 \pm 3.23\bullet$	$77.23 \pm 4.10\circ$	78.19 ± 4.06	$81.04 \pm 3.23\bullet$
(cle)	94.31 ± 8.00	$93.70 \pm 4.67\circ$	$96.20 \pm 3.02\bullet$	$96.20 \pm 3.96\bullet$
(acc)	97.90 ± 0.81	$96.51 \pm 1.17\circ$	98.22 ± 1.08	$98.47 \pm 0.64\bullet$
rank	2.25	3.83	2.41	1.5

6 Conclusions

This paper introduces a new kind of binary discriminant classifier with nonlinear behavior based on the notion of characteristic boundary points—points that belong to an optimal classification boundary under some constraints. Using this concept, we build a set of linear classifiers that are assembled by means of a Tikhonov regularized optimization procedure in an additive model to create a final λ-smooth decision rule. As a result, a very simple method with a clear geometric and structural meaning is obtained. It automatically deals with the nonlinearities in the boundary—it does not need to alter the metric of the space using kernels. Thus, it removes the necessity of selecting and tuning kernels. Its accuracy statistically outperforms some of the most well-known machine learning strategies, such as Adaboost, k-NN, or Linear SVM, and it is on par with kernel approaches such as SVM with Radial Basis Function kernels or Relevance Vector Machines. The basic definition of the classifier has a moderate computational complexity. Thus, we propose an algorithm that allows to compute the ensemble in linear time as well as to control the complexity of the final model. There is a trade-off between complexity—memory requirements and computational time—and accuracy. The less complex the solution, the worse it will handle nonlinearities. This can potentially lead to a decrease in performance.

Due to its simplicity, it is easy to devise extensions for dealing with current machine learning problems. As an example, we show the performance of the technique as online learner and with three large-scale data sets. We validate this technique on 16 UCI data sets and apply it to six real computer vision and signal processing problems: Gender recognition based on face images, Intravascular Ultrasound tissue classification, speed traffic sign recognition, Chagas' disease myocardial damage severity detection, old musical scores clef classification, and action recognition using 3D accelerometer data from a wearable device.

Since this paper introduces a new technique, there are many open lines of research. Some points that deserve further attention are: the search of different criteria for defining the robustness notion for the CBPs, the analysis of alternate optimization strategies forcing sparsity constraints to control the complexity of the final model, or the theoretical relationship of this strategy with the Bayesian framework.

References

[1] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics*, pp. 337-407, 2000.
[2] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
[3] B.K. Bhattacharya, R.S. Poulsen, and G.T. Toussaint, "Application of Proximity Graphs to Editing Nearest Neighbor Decision Rule," *Proc. Int'l Symp. Information Theory*, 1981.
[4] K. Mukherjee, "Application of the Gabriel Graph to Instance Based Learning," MSc thesis, Simon Fraser Univ., 2004.
[5] W. Zhang and I. King, "A Study of the Relationship between Support Vector Machine and Gabriel Graph," *Proc. Int'l Conf. Neural Networks*, vol. 1, pp. 239-244, 2002.
[6] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, pp. 259-270, 1969.
[7] B. Bhattacharya, K. Mukherjee, and G.T. Toussaint, "Geometric Decision Rules for Instance-Based Learning Algorithms," *Proc. First Int'l Conf. Pattern Recognition and Machine Intelligence*, pp. 60-69, 2005.
[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.
[9] P.M. Murphy and D.W. Aha, *UCI Repository of Machine Learning Databases*, Univ. of California, Dept. of Information and Computer Science, 1994.
[10] A. Martinez and R. Benavente, "The AR Face Database," Technical Report 24, Computer Vision Center, Spain, 1998.
[11] K. Bennett and E. Bredensteiner, "Duality and Geometry in SVM Classifiers," *Proc. 17th Int'l Conf. Machine Learning*, pp. 57-64, 2000.
[12] J. Bi and K.P. Bennett, "A Geometric Approach to Support Vector Regression," *Neurocomputing*, vol. 55, pp. 79-108, 2003.
[13] S. Escalera, O. Pujol, E. Laciar, J. Vitrià, E. Pueyo, and P. Radeva, "Coronary Damage Classification of Patients with the Chagas' disease with Error-Correcting Output Codes," *Proc. IEEE Int'l Conf. Intelligence Systems*, vol. 2, pp. 12-17-12-22, 2008.
[14] K.L. Caballero, J. Barajas, O. Pujol, N. Salvatella, and P. Radeva, "In-Vivo IVUS Tissue Classification: A Comparison between RF Signal Analysis and Reconstructed Images," *Lecture Notes in Computer Science*, vol. 4225, pp. 137-146, 2006.
[15] M. Bern, D. Eppstein, and F. Yao, "The Expected Extremes in a Delaunay Triangulation," *Int'l J. Computational Geometry and Applications*, vol. 1, no. 1, pp. 79-92, 1991.
[16] C. Bishop and M. Tipping, "Variational Relevance Vector Machines," *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, pp. 46-53, 2000.
[17] O. Hamsici and A. Martinez, "Spherical-Homoscedastic Distributions: The Equivalency of Spherical and Normal Distributions in Classification," *J. Machine Learning Research*, vol. 8, pp. 1583-1623, 2007.
[18] J. Demsar, "Statistical Comparison of Classifiers over Multiple Data Sets," *J. Machine Learning Research*, vol. 7, pp. 1-30, 2006.
[19] S. Escalera, A. Fornes, O. Pujol, J. Llados, and P. Radeva, "Multi-Class Binary Object Categorization Using Blurred Shape Models," *Proc. Iberoam. Congress Pattern Recognition*, pp. 142-151, 2007.
[20] D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *J. Machine Learning Research*, vol. 5. pp. 361-397, 2004.
[21] OSU SVM Toolbox, http://svm.sourceforge.net, 2009.
[22] PRTools Toolbox, Faculty of Applied Physics, Delft Univ. of Technology, http://ww.prtools.org, 2009.