Application of Learning Machine Methods to 3 D Object Modeling

Cristina Garcia, and José Ali Moreno

Laboratorio de Computación Emergente, Facultades de Ciencias e Ingeniería, Universidad Central de Venezuela. Caracas, Venezuela.
{cgarcia, jose}@neurona.ciens.ucv.ve
http://neurona.ciens.ucv.ve/Laboratorio

Abstract. Three different machine learning algorithms applied to 3D object modeling are compared. The methods considered, (Support Vector Machine, Growing Grid and Kohonen feature Map) were compared in their capacity of modeling the surface of several synthetic and experimental 3D objects. The preliminary experimental results show that with slight modifications these learning algorithms can be very well adapted to the task of object modeling. In particular the Support Vector Machine Kernel method seems to be a very promising tool.

1 Introduction

Object modeling is a very important technique of computer graphics and has been object of study for more than two decades. The technique has found a broad range of applications, from computer-aided design and computer drawing to image analysis and computer animation. In the literature several approaches to object modeling can be found. One approach is to look for a rigid model that best fits the data set, an alternate one is to deform a model to fit the data. The later ones are known as dynamically deformable models. They were first introduced by Kass, Witkin and Terzopoulos [8] and have created much interest since then because of their clay like behavior. A complete survey can be found in [7].

To achieve 3D object modeling, it is desirable that the applied method show a great flexibility in terms of shape and topology representation capacity. A deformable model with this characteristic is the simplex mesh developed by Delingette [4][5]. It is a non-parametric deformable model, that can take virtually any shape of any topology with a very low computational cost as compared to other deformable models with the same flexibility. It is well known that several machine learning algorithms posses the ability to induce efficiently, in a more or less automatic manner, arbitrary surfaces and topology preserving mappings on arbitrary dimensional noisy data. The proposed approach to 3D object modeling takes advantage of this low cost surface representation capacity inherent to these learning algorithms.

In this paper we describe the application of the Support Vector Kernel Method (SVM) [3], the Kohonen Self-Organizing Feature Maps [9] and the Growing Grid [6] machine learning algorithms to model objects from data sets that contain information from one or more 3D objects. In general the application of the algorithms start with a cloud of 3D data points and no a priori information regarding the shape or topology of the object or objects in the scene. These data points are applied to the learning algorithms, that with simple modifications on the learning rules, generate adaptively a surface adjusted to the surface points. In case of the Kohonen Feature Map and the Growing Grid a spherical network is randomly initialized in the interior of the cloud of points. Then the learning rule is applied and the network deforms and grows (Growing Grid) until it reaches stability at the surface of the cloud of points. In case of the Support Vector Kernel Method the data points are mapped to a high dimensional feature space, induced by a Gaussian kernel, where support vectors are used to define a sphere enclosing them. The boundary of the sphere forms in data space (3D) a set of closed surfaces containing the cloud of points. As the width parameter of the Gaussian kernel is increased, these surfaces fit the data more tightly and splitting of surfaces can occur allowing the modeling of several objects in the scene. At the end of each of these processes we will have a model for each object.

The organization of the paper is as follows: in the second section, an overview of the applied learning machine methods and their modifications for 3D object modeling are discussed. In the third section experimental results are presented and finally in the fourth section the conclusions and further work are described.

2 Learning Machine Methods

3D Object modeling is an ill-defined problem for which there exist numerous methods [2],[7],[10],[12]. In the present approach, learning machine methods that produce data clustering are applied to surface modeling. These clustering methods can be based on parametric models or can be non-parametric. Parametric algorithms are usually limited in their expressive power, i.e. a certain cluster structure is assumed. In this work experiments of surface modeling with two parametric (Kohonen Feature Map and Growing Grid) and a non-parametric (SVM Kernel Method) clustering algorithms are presented and compared. In what follows a brief discussion of each learning method is made.

2.1 Support Vector Kernel Method

The idea behind the application of the support vector formalism [3] to object modeling follows the SVM clustering method in [1]. Let $\{x\}$ be a data-set of N 3D points representing the object in the scene. Using a nonlinear transformation f from the input 3D space to some high dimensional feature-space, we look for the smallest enclosing sphere of radius R, described by the constraints:

$$\left\| f\left(\boldsymbol{x}_{\boldsymbol{i}}\right) - \boldsymbol{a} \right\|^{2} \leq \boldsymbol{R}^{2} \quad \forall\, \boldsymbol{i} \tag{1}$$

Where use is made of the Euclidean norm and ***a*** is the center of the sphere. Soft constraints are incorporated by adding slack variables ξ_ι

$$\left\| f\left(\boldsymbol{x}_{\boldsymbol{i}}\right) - \boldsymbol{a} \right\|^{2} \leq \boldsymbol{R}^{2} + x_{\boldsymbol{i}} \;; \quad x_{\boldsymbol{i}} \geq 0 \quad \forall\, \boldsymbol{i} \tag{2}$$

This problem is solved in the formalism of Lagrange by introducing and minimizing the Lagrangian

$$\boldsymbol{L} = \boldsymbol{R}^{2} - \sum_{\boldsymbol{i}} \left(\boldsymbol{R}^{2} + x_{\boldsymbol{i}} - \left\| f\left(\boldsymbol{x}_{\boldsymbol{i}}\right) - \boldsymbol{a} \right\|^{2} \right) b_{\boldsymbol{i}} - \sum x_{\boldsymbol{i}} m_{\boldsymbol{i}} + \boldsymbol{C} \sum x_{\boldsymbol{i}} \tag{3}$$

Where m, ***b*** are positive Lagrange multipliers, *C* is a constant and the last term is a penalty term.

The stationarity of the Lagrangean with respect to ***R*** and ξ_ι leads to the following relations:

$$\sum_{\boldsymbol{i}} b_{\boldsymbol{i}} = 1 \tag{4}$$

$$\boldsymbol{a} = \sum_{\boldsymbol{i}} b_{\boldsymbol{i}} f\left(\boldsymbol{x}_{\boldsymbol{i}}\right) \tag{5}$$

$$b_{\boldsymbol{i}} = \boldsymbol{C} - m_{\boldsymbol{i}} \tag{6}$$

The Karush, Kuhn and Tucker complementary conditions result in:

$$x_{\boldsymbol{i}} m_{\boldsymbol{i}} = 0 \tag{7}$$

$$\left(\boldsymbol{R}^{2} + x_{\boldsymbol{i}} - \left\| f\left(\boldsymbol{x}_{\boldsymbol{i}}\right) - \boldsymbol{a} \right\|^{2} \right) b_{\boldsymbol{i}} \tag{8}$$

From these relations it is easy to verify that a point $\boldsymbol{x_i}$ with $\xi_i > 0$ is outside the sphere in feature space, such points have $m_i = 0$ and $b_i = C$. A point with $\xi_i = 0$ is inside or on the surface of the sphere in feature space. To be on the surface it must have b_i not equal to zero. Points with $0 < b_i < C$ will be referred to as Support Vectors.

The above relations allow the derivation of the Wolf dual of the Lagrangian:

$$W = \sum_i b_i f(x_i).f(x_i) - \sum_{ij} b_i b_j f(x_i).f(x_j) \tag{9}$$

and the problem is solved by maximizing the dual. The dot products $f(x_i).\ f(x_j)$ can be conveniently replaced by a suitable Mercer kernel $\boldsymbol{K(x_i, x_j)}$ in this way the Wolf dual can be rewritten as

$$W = \sum_i b_i K(x_i, x_i) - \sum_{ij} b_i b_j K(x_i, x_j) \tag{10}$$

The Lagrange multipliers b_i are obtained by maximizing this expression. This is computationally done by the application of the SMO algorithm [11].

In the approach to object modeling with SVM the Gaussian kernel is employed

$$K(x_i, x_j) = exp\left(-q\left\|x_i - x_j\right\|^2\right) \tag{11}$$

In feature space the square of the distance of each point to the center of the sphere is

$$R^2(x) = \left\|f(x) - a\right\|^2 \tag{12}$$

The radius of the sphere is

$$R = \left\{R(x_i) \mid x_i \text{ is a support vector}\right\} \tag{13}$$

In practice the average over all support vectors is taken. The surface of the clouds of points in 3D data space is given by the set:

$$\left\{x \mid R(x) = R\right\} \tag{14}$$

2.2 Kohonen Feature Map

The Kohonen Feature Map [9] is an unsupervised learning machine that typically consists of one layer of neurons in a network of constrained topology. In its learning phase the weight vectors are randomly initialized. During learning, for every input vector the Best Matching Unit (BMU) is determined. The BMU and a number of units in a neighborhood of the BMU, in the constrained topological network, are adjusted in such a way that the weight vectors of the units resemble the input vector more closely. The units surrounding the BMU are adjusted less strongly, according to the distance they have to the BMU.

The weight vectors $w_j(t)$ are adjusted by applying the Kohonen Learning Rule:

$$\boldsymbol{w_j}(\boldsymbol{t}+1)=\boldsymbol{w_j}(\boldsymbol{t})+e(\boldsymbol{t}).F_{\boldsymbol{ji}}(\boldsymbol{t}).(\boldsymbol{x}(\boldsymbol{t})-\boldsymbol{w_j}(\boldsymbol{t})) \quad \textbf{(15)}$$

where the learning rate $e(t)$ is a linear time decreasing function, $\boldsymbol{x(t)}$ is the input vector at time t, and $F_{ij}(t)$ is the neighborhood function with the form:

$$F_{\boldsymbol{ji}}(\boldsymbol{t})=exp\left(-\left|\boldsymbol{j}-\boldsymbol{i}\right|^2 / s^2(\boldsymbol{t})\right) \quad \textbf{(16)}$$

Here j is the position of the BMU in the topological network and i the position of a unit in its neighborhood. The width parameter $s(t)$ is also a linear time decreasing function. It can be noted that since the learning rate and the width parameter both decrease in time the adjustments made on the weight vectors become smaller as the training progresses. On a more abstract level, this means that the map will become more stable in the later stages of the training.

It can be seen that the result of learning is that the weight vectors of the units resemble the training vectors. In this way the Kohonen Feature Map produces a clustering of the n-dimensional input vectors onto the topological network.

In order to model the surface of the input data points, two modifications to the Kohonen Feature Map are introduced: (a) The constrained topological network chosen is a spherical grid. (b) the weight vectors of the BMU and its neighbors are actualized only if the input vector is external to the actual spherical grid.

The implementation used three different rates to decrease parameters ε and σ: a first rapid decreasing stage a middle slower but longer stage and a final short fine tuning stage.

2.3 Growing Grid Method

This model [6] is an enhancement of the feature map. The main difference is that the initially constrained network topology grows during the learning process. The initial architecture of the units is a constrained topological network with a small number of units. A series of adaptation steps, similar to the Kohonen learning rule, are executed in order to update the weight vectors of the units and to gather local error information at each unit. This error information is used to decide where to insert new units. A new unit is always inserted by splitting the longest edge connection emanating from the unit with maximum accumulated error. In doing this, additional units and edges are inserted such that the resulting topological structure of the network is conserved.

The implemented process involves three different phases: an initial phase where the number of units is held constant allowing the grid to stretch, a phase where new units are inserted and the grid grows, and finally a fine tuning phase.

3. Experimental Results

The learning algorithms for modeling are applied on several synthetic objects represented in the form of clouds of 3000 points each obtained from the application of a 3D lemniscate with 4, 5 and 6 foci. The initial and final parameters used for the Kohonen Feature Map were: $\varepsilon_0 = 0.5$, $\varepsilon_f = 0.0001$, $\sigma_0 = 4$, $\sigma_f = 0.01$, with a total 10^6 iterations. For the Growing Grid: $\varepsilon_0 = 0.08$, $\varepsilon_f = 0.05$, a constant $\sigma = 0.7$, the number of iterations is distributed as: 500 * grid size in the stretching phase; 500 * grid size in the growing phase; and 200 * grid size iterations for fine tuning. The parameters in the SVM algorithm are C = 1.0 and q = 2 (Fig. 1 and Fig. 3), q = 4 (Fig. 2) and q = 0.00083 (Fig. 4).

In figure 1 the original surface (5 foci lemniscate) and the surface models resulting from the application of the three learning methods (a) Kohonen Feature Map (b) Growing Grid and (c) SVM algorithms are shown. The Kohonen Map consisted on a spherical topological network with 182 units randomly initialized in the interior of the cloud of points. The Growing Grid was also a spherical network with 6 initial units randomly initialized in the interior of the cloud, the network was grown up 162 units. The SVM model was constructed with 49 support vectors.

It can be appreciated that the three algorithms achieve a reasonable modeling of the original object. The best results are produced by the Growing Grid and SVM methods.

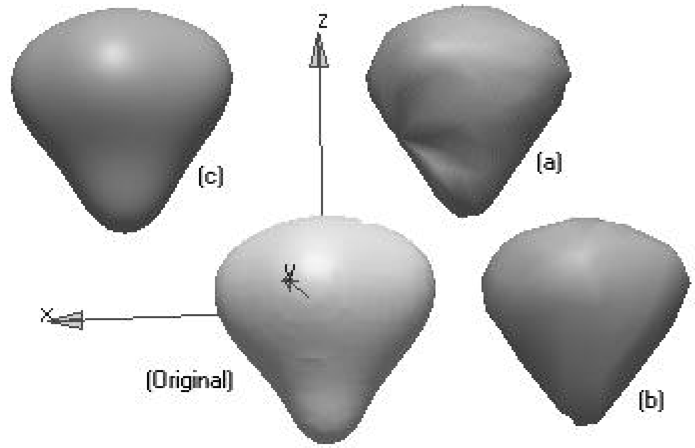

Fig. 1. Results of the three machine learning methods in the modeling of a surface from a solid generated by a 5 foci lemniscate. (a) Kohonen Feature Map (b) Growing Grid and (c) SVM Kernel method.

In figure 2 the original surface (6 foci lemniscate) and the surface models resulting from de application of the three learning methods (a) Kohonen Feature Map (b) Growing Grid and (c) SVM algorithms are shown. The Kohonen Map consisted on a spherical topological network with 266 units randomly initialized in the interior of the cloud of points. The Growing Grid was also a spherical network with 6 initial units randomly initialized in the interior of the cloud the network and grown up to 338 units. The SVM model was constructed with 77 support vectors.

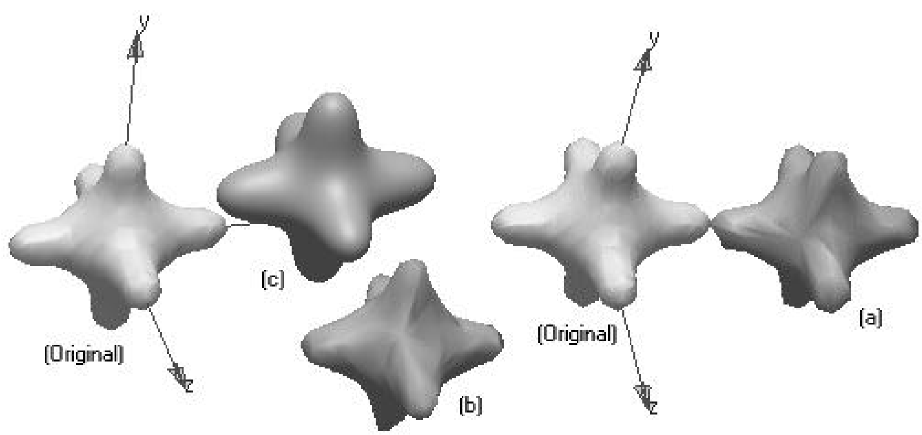

Fig. 2. Results of the three machine learning methods in the modeling of a surface from a solid generated by a 6 foci lemniscate. (a) Kohonen Feature Map (b) Growing Grid and (c) SVM Kernel method.

It can be appreciated that again in this experiment the three algorithms achieve a reasonable modeling of the original object. The best results are produced by the Growing Grid and SVM methods.

In figure 3 the original surface (4 foci lemniscate) and the surface models resulting from the application of the two learning methods (a) Growing Grid and (b) SVM algorithms are shown. For this case the Growing Grid was a spherical network with 6 initial units randomly initialized in the interior of the cloud of points. The network was grown up 134 units. The SVM model was constructed with 38 support vectors. In this experiment the object is of particular interest since it consists of two parts joined by a point, an approximation to a scene of two separate objects. In this case both the Kohonen feature Map and the Growing Grid do not produce a good model of the surface. However, it can be appreciated that the SVM method achieves a very good model and is clearly superior.

In figure 4 the original surface (experimental data from the left ventricle of a human heart echocardiogram) and the surface models resulting from the application of the two learning methods (a) Growing Grid and (b) SVM algorithms are shown. For this case the Growing Grid was a spherical network with 6 initial units randomly initialized in the interior of the cloud of points. The network was grown up 282 units. The SVM model was constructed with 46 support vectors.

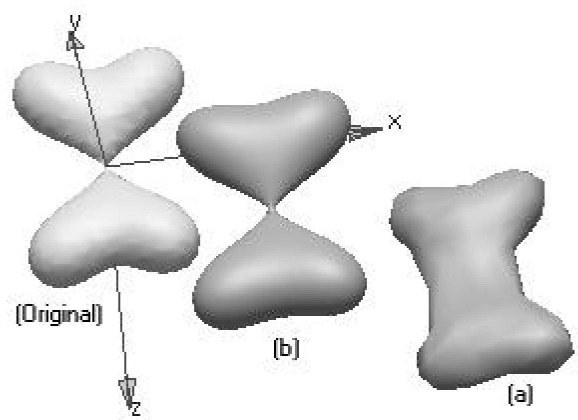

Fig. 3. Results of two machine learning methods in the modeling of a surface from a solid generated by a 4 foci lemniscate. (a) Growing Grid and (b) SVM Kernel method.

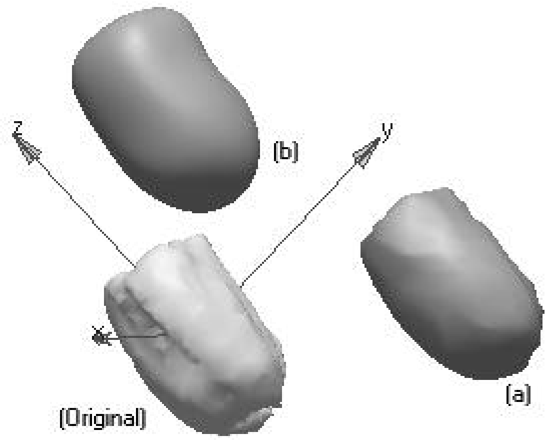

Fig. 4. Results of two machine learning methods in the modeling of a surface from experimental data (left ventricle of a human heart echocardiogram) (a) Growing Grid and (b) SVM Kernel method.

4. Conclusions and Future Work

This work compared the application of three machine learning algorithms in the task of modeling 3D objects from a cloud of points that represents either one or two objects.

The experiments show that the Kohonen Feature Map and the Growing Grid methods generate reasonable models for single objects with smooth spheroidal surfaces. If the object posses pronounced curvature changes in its surface the modeling produced by these methods is not very good. An alternative to this result is to allow the number of units in the network to increase together with a systematic prune mechanism of the edges in order to take account of the abrupt changes on the surface. This modifications are theme of further work in case of the Growing Grid algorithm.

On the other hand, the experimental results with the Support Vector Kernel Method are very good. In the case of single smooth objects the algorithm produces a sparse (small number of support vectors) model for the objects. A very convenient result for computer graphics manipulations of the object. This extends to the case with two objects in which the method is able to produce models with split surfaces. A convenient modification of the SVM algorithm would be to include a better control on the number of support vectors needed in the model. This possibility could hinder the rounding tendency observed in the SVM models and allow the modeling of abrupt changes of the surface as seen on the data of the echocardiogram.

To model multiple objects it is necessary, in the Kohonen and Growing Grid methods, the application of a splitting algorithm to the topological network. This splitting algorithm is related to the systematic prune of the edges and is also theme of further work.

The data sets for multiple objects are not a representation of a real scene. In a real scene the clusters of data points will be connected by background information (walls and floor). The future work will also include the extension of the actual algorithms to make them applicable to real scenes.

Finally it must be noted that in all cases the computational costs of the algorithms are not very high a fact that can lead to real time implementations.

References

1. A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, A Support Vector Method for Clustering. International Conference on Pattern Recognition 2000.
2. M. Bro-Nielsen: Active Nets and Cubes. Technical Report, Institute of Mathematical Modeling Technical University of Denmark, November 1994.
3. N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines. Cambridge University Press, 2000.
4. H. Delingette, General Object Reconstruction based on Simplex Meshes. Technical Report 3111, INRIA, France. (1997)
5. H. Delingette, Simplex Meshes: A General Representation for 3D Shape Reconstruction. Technical Report 2214, INRIA, France. (1994)
6. B. Fritzke, Some Competitive Learning Methods. Institute for Neural Computation, Ruhr-Universitat Bochum, Draft Report, 1997.
7. S.F.F. Gibson, B. Mirtich, A Survey of Deformable Modeling in Computer Graphics. Mitsubishi Electric Research Laboratory Technical Report, November (1997)
8. M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models. Int. J. Of Computer Vision. 2 (1998) 321-331
9. T. Kohonen, Self-Organization Associative Memory. (3rd Edition). Springer-Verlag, Berlin 1989.
10. D. Metaxas, Physics-based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging. Kluwer Academic, Boston, 1997.
11. J. Platt, Fast Training of Support Vector Machines Using Sequential Minimal Optimization. http://www.research.microsoft.com/~jplatt
12. K. Yoshino, T. Kawashima and Y. Aoki: Dynamic Reconfiguration of Active Net Structure, Proceedings of Asian Conference in Computer Vision, pp. 159-162, 1993.