

# Primal Optimization for Implicit Surface Modeling

Michael Rawson  
mr4209@nyu.edu

**Abstract.** We optimize the computation of hypersurfaces from finite samples. This is done by mapping sample points into a reproducing kernel Hilbert space and determining regions in terms of hyperplanes. We use different methods to analyze run times and scaling with epsilon insensitivities. The hypersurface can then be used for many applications, including computer models.

**Key words.** Kernel Methods, Gram Matrices, Support Vector Machines, Unsupervised Learning

**AMS subject classifications.** 65D15

## 1. Introduction.

Computerized models are very useful with today’s technology. These models can be used to simulate stress on objects, assemble complex objects, and 3D print precise objects repeatedly. 3D scanners are typically used to create computer models of physical objects using interpolation of some form. However, creating a model by interpolating linear planes between points can only create an approximation of the real model because a 3D scanner can only supply a finite number of samples. For example, when attempting to model aerodynamical forces, having a smooth object, and not linear approximations, is very important. An object can be smoothly represented by a hypersurface in a reproducing kernel Hilbert space. We employ support vector machines to transform finite samples, or 3D points, into a hyperplane to represent the 3D object.

Our goal is to find a function that represents a surface at it’s level set, or zero value, and is a “reasonable” model.

## 2. Previous Work.

### 2.1. Single-Class SVM.

In support vector machines (SVMs), often the input data is transformed into some feature space,  $\Phi(x)$ , specifically a reproducing kernel Hilbert space. We can have a kernel such that  $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$ . A Gaussian kernel takes the form  $k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$ .

Single-class SVMs were published in [1] and [2], and estimate a surface given independently and identically distributed data from some probability distribution. By minimizing the norm of the representative function and optimizing to pass through the data, a unique solution is had.

$$\min_{w \in H, \xi \in R^m, \rho \in R} \frac{1}{2} \|w\|^2 + \frac{1}{vm} \sum_i \xi_i - \rho$$

$$\text{subject to } \langle w, \Phi(x_i) \rangle \geq \rho - \xi_i, \xi_i \geq 0$$

Where  $H$  in a reproducing kernel Hilbert space and  $\xi_i$  are slack variables.

Usually a quadratic program solver is used to solve the dual problem. Let  $k(x_i, x_j)$  be the distance between two  $x$  points according to some feature map or kernel.

$$\min_{\alpha \in \mathbb{R}^m} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j)$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{vm} \text{ and } \sum_i \alpha_i = 1.$$

Then, let  $f$  be the function to represent the surface. Then our evaluation of the function  $f$  is

$$f(x) = \sum_i \alpha_i k(x_i, x) - \rho.$$

## 2.2. Slab SVM.

Scholkopf, Glenssen, and Spallinger created the slab SVM in [3]. The slab SVM allows for a richer set of solutions by allowing weights, or slack, to be negative compared to a slab around the hypersurface of some width  $\delta$ . Consider an optimization problem where a  $*$  makes the weight positive and no  $*$  means negative (let  $(*)$  mean either).

$$\min_{w \in H, \xi_i^{(*)} \in \mathbb{R}^m, \rho \in \mathbb{R}} \frac{1}{2} \|w\|^2 + \frac{1}{vm} \sum_i (\xi_i + \xi_i^*) - \rho$$

$$\text{subject to } \delta - \xi_i \leq \langle w, \Phi(x_i) \rangle - \rho \leq \delta^* + \xi_i^* \text{ and } \xi_i^{(*)} \geq 0$$

Where  $H$  in a reproducing kernel Hilbert space,  $\xi_i^{(*)}$  are slack variables, and  $\delta$  is the slab width.

## 3. Experimental Results.

In all our experiments we used a Gaussian kernel while tuning the  $\sigma$  parameter. We optimize upon the primal problem using Newton's method with Armijo backtracking line search, where the iterations are

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n), n \geq 0.$$

Where  $f$  is the objective function to minimize and  $H$  is the Hessian.

We notice that if the slab width,  $\delta$ , is 0, then in practice we can pre-compute the Hessian and it's decomposition with the Cholesky method,  $n^3/3$  FLOPs. We plot volumetric data using marching cubes algorithm.

We notice that as sigma grows, some eigenvalues of the Gram matrix move towards 0. These eigenvalues often can become negative due to machine precision, and then we cannot proceed with a non-positive semi-definite matrix.

We also run implement incomplete Cholesky decomposition, but as this drops low eigenvalues, we again run into the same problem with non-positive semi-definite Kernel matrices due to machine precision.

We notice that as the data size grows, inverting the Hessian grows at  $n^3$ . To combat this, we can optimize with steepest descent or a Jacobian approximation.

#### 4. Conclusion.

Scaling with large datasets is becoming more important as computing capacity grows. 3D printers are also becoming more powerful and need accurate models to be useful. Manipulating all of this data together, at once, is too taxing for hardware. Thus using more compact models is highly desirable for many of these applications.

#### REFERENCES

- [1] B. SCHOLKOPF, J. PLATT, J. SHAWE-TAYLOR, A. J. SMOLA, AND R. C. WILLIAMSON. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443-1471, 2001.
- [2] D. M. J. TAX AND R. P. W. DUIN. Support vector data description. *Machine Learning*, 54:45-66, 2004.
- [3] BERNHARD SCHOLKOPF AND JOACHIM GIESEN AND SIMON SPALINGER. Kernel methods for implicit surface modeling. *Advances in Neural Information Processing Systems 17*, 1193-1200, 2005.
- [4] BACH AND JORDAN. Kernel Independent Component Analysis. *Journal of Machine Learning Research* 3, 1-48, 2002.
- [5] FINE AND SCHEINBERG. Efficient SVM Training Using Low-Rank Kernel Representations. *Journal of Machine Learning Research* 2, 243-264, 2001.
- [6] BACH AND JORDAN. Predictive low-rank decomposition for kernel methods. *ICML '05*, 33 - 40, 2005.